Федеральное государственное бюджетное учреждение науки
Санкт-Петербургский институт информатики и автоматизации
Российской академии наук
(СПИИРАН)

На правах рукописи

Тушканова Ольга Николаевна

Семантические структуры и причинные модели больших данных для принятия решений с приложением к рекомендательным системам

Специальность 05.13.01 – Системный анализ, управление
и обработка информации (технические системы)

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель
д.т.н, профессор
Городецкий В.И.

Санкт-Петербург – 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность темы исследования. В настоящее время проблема обработки больших данных относится к числу наиболее актуальных проблем в области информационных технологий, и она же порождает наиболее трудные проблемы алгоритмического характера, связанные с обеспечением точности, устойчивости и вычислительной эффективности процессов их обработки. Эти проблемы обусловлены тем, что большинство традиционных методов интеллектуального анализа данных напрямую не могут быть применены для анализа больших данных либо вследствие вычислительной неустойчивости, либо вследствие вычислительной сложности. Не менее трудные проблемы обусловлены гетерогенным характером больших данных: они могут содержать атрибуты разных типов и неструктурированные данные, например, тексты на естественном языке.

Одно из важных требований к методам обработки больших данных – это семантически ясная интерпретация результатов. В современных моделях знаний это обеспечивается средствами онтологии, однако ее построение для больших данных также является проблемой: ввиду огромного количества потенциальных понятий ручная разработка онтологии становится непомерно трудоемкой, а потому требует максимальной автоматизации, а в ряде классов приложений – полной автоматизации.

Анализ состояния исследований и разработок по сформулированным проблемам показывает, что существующие методы и алгоритмы обработки больших данных не отвечают ожиданиям и потребностям специалистов в этой области.

Обычно целью обработки больших данных является построение эмпирической модели целевых переменных, учитывающей некоторые атрибуты данных. При этом ключевым требованием является минимизация числа используемых атрибутов при условии обеспечения заданной точности модели. В данной работе задачи такого типа являются основным предметом исследований и разработок. Среди наиболее перспективных подходов к решению этой задачи в настоящее время выделяется подход, который базируется на обнаружении ассоциативных связей в данных и последующим использовании их в моделях принятия решений,

в частности, в моделях ассоциативной классификации. Однако задачи разработки вычислительно эффективных алгоритмов поиска ассоциативных правил классификации и построения механизмов принятия решений на основе этих правил пока не имеют удовлетворительных решений и остаются актуальными.

В исследованиях К.Ф. Алайфериса (C.F. Aliferis) строго показано, для принятия решений, в частности, для ассоциативной классификации, среди ассоциативных связей наиболее информативными являются связи причинного характера и их структуры. Поиск причинных структур традиционно основан на построении и анализе байесовских сетей доверия. Этот подход требует решения задач обучения экспоненциальной сложности относительно размерности пространства атрибутов, и практически он может использоваться только тогда, когда размерность не превышает 20. Поэтому этот подход бесперспективен для причинного анализа больших данных. Важно также отметить, что Я. Виттен (Y. Witten) подчеркивает, что в байесовской сети, построенной на основе машинного обучения, далеко не все выявленные связи между переменными в реальности оказываются причинными. Эти обстоятельства требуют разработки альтернативных подходов к поиску причинных связей в больших данных. Такой альтернативой в настоящее время является ассоциативно-причинный анализ данных. Он базируется на использовании специальных мер оценки ассоциативной связи, которые позволяют из всего множества таких связей выделить те, которые носят причинный характер.

Целью работы является разработка алгоритмов обучения и принятия решений в задачах классификации на основе семантических и причинных моделей больших данных, их реализация в форме программного прототипа, а также экспериментальная оценка по таким характеристикам как масштабируемость, вычислительная эффективность и точность в задачах принятия решений, в частности, в рекомендательных системах.

В соответствии с поставленной целью в работе сформулированы и решены следующие частные **задачи исследования**:

1. Теоретически и экспериментально обоснованный выбор семантически корректной и вычислительно эффективной формальной меры, оценивающей «силу» причинной связи атрибутов данных.

2. Разработка алгоритма автоматической генерации семантической модели больших данных, понятия которой используются для представления знаний о данных и результатов обучения в форме причинных моделей для принятия решений на основе ассоциативно-причинной классификации.

3. Разработка единой структуры для представления синтаксиса и семантики больших данных, а также метаинформации о них. Структура должна обеспечивать доступ к данным и упрощать вычисления различных статистик.

4. Разработка математически корректного, масштабируемого и вычислительно эффективного алгоритма поиска причинных связей в больших данных.

5. Разработка масштабируемого алгоритма минимизации размерности причинной модели принятия решений и механизма ассоциативно-причинной классификации.

6. Экспериментальная оценка разработанных алгоритмов на стандартных наборах данных из области рекомендательных систем третьего поколения.

Результаты решения перечисленных задач – это компоненты многошагового алгоритма генерации семантических моделей больших данных для решения задач ассоциативно-причинной классификации. Их разработка, интеграция в рамках единого процесса, программное прототипирование и экспериментальное исследование с использованием стандартных наборов данных из области рекомендательных систем составляют содержание работы.

Научной новизной обладают следующие результаты работы:

1. Теоретически и экспериментально обоснованная, семантически корректная и вычислительно эффективная мера оценки «силы» причинной связи между атрибутами данных. Рекомендации по ее выбору основаны на обширном экспериментальном исследовании и построены на численных оценках.

2. Алгоритм автоматической генерации семантической модели больших данных, ориентированной на построение и минимизацию множества причинных правил модели ассоциативно-причинной классификации. Алгоритм отличается тем, что построен как комбинация методов и средств автоматизированной генерации иерархии понятий онтологии данных и генерации двойственных формальных понятий, определяющих условия останова процесса генерации понятий.

3. Семантическая модель больших данных и структура для ее представления. Эта модель включает в себя метаинформацию о данных, их синтаксис и семантику, которые представлены в единой структуре. Эта структура состоит из иерархии понятий онтологии данных и иерархии соответствующих им двойственных формальных понятий, при этом каждое понятие обеих иерархий ссылается на множество примеров данных, составляющих его объем. Такая структура для представления семантической модели данных обеспечивает эффективный поиск причинных зависимостей в данных для алгоритмов ассоциативно-причинной классификации. Эта структура строится за один проход по данным, а все последующие вычисления не требуют дополнительного обращения к данным. В то ж время в ней представлены и данные сами по себе, что обеспечивает возможность инкрементного обогащения сематической модели данных при поступлении новых данных.

4. Масштабируемый и вычислительно эффективный алгоритм поиска множества причинных зависимостей между атрибутами данных, использующий семантическую модель данных.

5. Алгоритм минимизации мощности множества причинных правил модели ассоциативно-причинной классификации путем устранения избыточности правил. Алгоритм отличается тем, что основан на кластеризации множества правил: каждый кластер содержит в себе сильно коррелированные правила, а правила из разных кластеров слабо коррелированы.

Теоретическая и практическая ценность работы заключается в разработке множества теоретически корректных и экспериментально проверенных алгорит-

мов и программной библиотеки, реализующей базовые процедуры обучения моделей ассоциативно-причинной классификации и принятия решений с ориентацией на задачи рекомендательных систем третьего поколения. Программная библиотека реализована в виде множества классов на языке Java и других вспомогательных программ которые построены с учетом возможности повторного использования в широком круге задач обработки больших данных и принятия решений. Разработанные алгоритмы и программный комплекс протестированы на больших данных, ориентированных на разработку приложений в области персонифицированных и кросс-доменных рекомендательных систем. На основании результатов тестирования даны практические рекомендации по выбору алгоритмов и технологий разработки современных рекомендательных систем третьего поколения.

Методы и средства исследования. В работе использовались методы корреляционного, ассоциативного и причинного анализа, методы машинного обучения и объединения решений распределенных классификаторов, методы теории графов, теории вероятностей и математической статистики, методы и средства онтологического моделирования, методы теории частично упорядоченных множеств и решеток, методы анализа формального понятий, кластерного анализа. При разработке архитектуры программного комплекса использованы функционально- и объектно- ориентированные подходы.

Положения, выносимые на защиту:

1. Теоретически и экспериментально обоснованная мера оценки силы причинной связи обеспечивает вычислительную эффективность и масштабируемость алгоритмов поиска причинных связей в больших данных.

2. Вычислительно эффективный алгоритм автоматического построения онтологии позволяет построить семантическую модель больших данных, пригодную для обучения и минимизации причинных моделей в задачах ассоциативно-причинной классификации.

3. Семантическая модель больших данных, которая представляет мета-свойства данных, их синтаксис и семантику в рамках единой структуры, удобной с

точки зрения доступа к большим данным в задачах обучения и принятия решений на основе ассоциативно-причинной классификации.

4. Алгоритм поиска причинных связей в данных является вычислительно эффективным и масштабируемым, что обеспечивается выбранной мерой оценки силы причинной связи и использованием семантической модели данных.

5. Алгоритм минимизации размерности пространства причинных правил модели ассоциативно-причинной классификации реализует снижение избыточности модели принятия решений и минимизацию размерности модели практически без потери точности.

Публикации. Основные положения диссертации опубликованы в 9 печатных работах, включая 3 публикации в рецензируемых научных изданиях из перечня ВАК: «Труды СПИИРАН», «Информационные технологии и вычислительные системы», также из перечня изданий, индексируемых в международных базах данных Web of Science и Scopus (4 работы).

Структура и объем работы. Диссертация изложена на 226 страницах машинописного текста, содержит 50 иллюстраций и 19 таблиц, состоит из введения, четырех глав, заключения, списка литературы (140 наименований) и четырех приложений.

В первой главе выявлены особенности задачи построения моделей принятия решений на основе больших данных, выделены основные проблемы и проведен анализ состояния исследований и разработок в этой области. Даны краткие сведения о рекомендательных системах как о типовом приложении обработки больших данных, описана суть методов ассоциативного и причинного анализа больших данных в задачах принятия решений, а также обоснованы и детализированы цели и задачи исследования.

Во второй главе выполнен детальный критический обзор основных подходов, методов и алгоритмов, предложенных в области ассоциативной и ассоциативно-причинной классификации, в том числе для обработки больших данных. В ходе сравнительного анализа дана оценка вклада различных методов и алгоритмов

ассоциативной классификации, в развитие этого направления, а также их возможностей в задачах обработки больших данных. Представлен краткий обзор численных мер оценки ассоциативных связей, разработанных в статистике, социологии, машинном обучении и интеллектуальном анализе данных, и проведен сравнительный анализ нескольких их ключевых свойств, важных с позиций причинного анализа данных.

В третьей главе предложен новый подход к автоматическому построению онтологии данных, названный семантическим анализом понятий, а также новая модель представления данных, объединяющая их мета-свойства, синтаксис и семантику в рамках единой структуры.

В четвертой главе предложен новый подход к построению причинного семантического профиля интересов пользователя рекомендательной системы третьего поколения, описаны разработанные алгоритмы выработки рекомендаций различными способами, а также приведены результаты экспериментального исследования всех разработанных алгоритмов на наборе данных Amazon.

1 АНАЛИЗ ЗАДАЧИ ПОСТРОЕНИЯ МОДЕЛЕЙ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ БОЛЬШИХ ДАННЫХ

1.1 Особенности задачи построения моделей принятия решений на основе больших данных

В настоящее время важнейшие и наиболее сложные современные проблемы интеллектуального анализа данных относятся к разделу «Большие данные». Понятие больших данных введено в научном сообществе специалистов по анализу данных относительно недавно (2008 год) [1]. К большим данным относят обычно не просто данные большого объема или высокой размерности. Полагают, что большие данные являются гетерогенными, обладают сложной структурой, а также, возможно, хранятся в распределенных базах данных. Структуры, используемые для представления больших данных, описываются разнородными атрибутами (числовыми, булевыми, ординальными, номинальными), они могут представлять собой изображения, содержать тексты на естественном языке, данные специальных форматов, (веб–ссылки, адреса электронной почты, номера телефонов и т.п.). Часто большие данные имеют характер потоков во времени и представляются множеством транзакций, записанных в транзакционных базах данных, которые могут иметь объемы, измеряемые петабайтами и более. Характерным признаком больших данных является также огромное количество (сотни и тысячи) атрибутов (высокая размерность пространства представления данных), каждый их которых, в свою очередь, может иметь структуру или быть неструктурированным. В англоязычной литературе про такие данные говорят, что они характеризуются «четырьмя V»: Volume, Velocity, Variety, Value, т.е. объемом, скоростью прироста, разнообразием шкал и структур представления компонент данных и ценностью [2].

Примерами больших данных являются потоки текстовых сообщений в социальных сетях, метеорологические, экологические и другие пространственно–временные сенсорные данные об окружающей среде, потоки данных о соединениях абонентов сотовой связи и их местонахождениях, серверные данные интернет–торговли, содержащие данные о покупателях, товарах и данные о динамике и

структуре покупок, данные о финансовых потоках банков с распределенными офисами, данные о геноме человека и другая информация медицинского назначения и т.п.

Согласно [3], существуют две основные цели анализа больших данных, именно разработка эффективных методов прогнозирования данных и выявление существующих зависимостей между атрибутами данных с целью, например, уменьшения размерности, построения эмпирических моделей целевых переменных, повышение эффективности поиска.

Аналитиков больших данных интересуют, прежде всего, те аспекты, методы и результаты анализа больших данных, которые связаны с решением различных задач принятия решений, ради которых и строятся прогностические модели данных. Хорошо известно, что разнообразие задач, которые необходимо решать с использованием больших данных, очень велико.

Возможно, наиболее распространенным классом таких задач является класс задач принятия решений по множеству распределенных источников (потоков) гетерогенных данных, которые могут обладать разной степенью достоверности, разнообразной частотой обновления, содержать пропущенные значения и т.п. Типичным примером задач этого класса является задача классификации ситуаций в сложных организационных и технических системах. Аналогичные задачи возникают при оценке и прогнозировании террористической опасности по результатам мониторинга террористических сайтов. Еще одним примером является задача выявления мошеннических банковских операций. Сюда же можно отнести круг задач принятия решений, связанный с анализом социальных сетей, например, задачи отнесения пользователей социальных сетей к той или иной группе, определение неформальных лидеров, обнаружение опасных сообществ и тому подобные задачи. Такие задачи чаще всего требуют построения подробной онтологии предметной области. Учитывая размерности данных и разнообразие задач, построение онтологии данных должно выполняться в максимально автоматизированном режиме, а в ряде важных и актуальных задач построение онтологии данных должно выполняться в полностью автоматическом режиме.

Отдельный, специфический и весьма важный класс задач обработки больших данных – это анализ многомерных временных рядов в реальном времени [4]. Одним из примеров здесь является конкурсная задача компании NineSigma [5]. Исходными данными в этой задаче являются данные мониторинга непрерывно работающего автомобильного производства, получаемые примерно четырьмя миллионами сенсоров, которые возвращают результаты измерений состояния оборудования и технологического процесса каждые 10 секунд. Такие данные накоплены более чем за 8 лет работы производства. Они содержат более 15 миллионов записей примерно по 4 миллиона чисел каждая. Задача обработки этих данных состоит в том, чтобы на основании предварительного анализа всех накопленных данных сначала выявить, классифицировать и структурировать типовые аномалии в поведении технологических процессов (процесс построения моделей аномалий) и их примеры. Полученные результаты (модели типовых аномалий и их примеры) необходимо использовать для принятия решений в ходе дальнейшей эксплуатации производства в интересах выявления или предсказания аномальных процессов в режиме реального времени, их диагностики и выдачи рекомендаций (на основе построенных моделей аномалий), в основном, о том, каким способом можно восстановить нормальный режима работы тех или иных компонент производства.

Широкий класс задач анализа больших данных связан с сетевым, или «вирусным» маркетингом [6]. Практика показала, что скорость распространения информации, например, в социальных сетях сильно зависит от того, в какие именно узлы социальной сети эта информация вводится. Задача поиска множества узлов сети, обеспечивающих оптимальность решения задачи вирусного маркетинга, явилась прототипом целого ряда других прикладных сетевых задач, которые требуют анализа скорости распространения данных, введенных в тот или иной узел крупномасштабной сети. Такими примерами являются оптимальная стратегия заражения сети компьютерными вирусами, формирования общественного мнения и другие.

Построение современных рекомендательных систем также относится к классу задач обработки больших данных, причем эта задача является весьма вос-

требованной в области интеллектуальной обработки данных и машинного обучения. Данные о пользователях, на основе которых строятся рекомендательные системы, относятся к большим данным. Отметим, что данные непосредственно о пользователях сами по себе могут и не быть большими, однако, для поиска интересов пользователя и эффективной работы рекомендательной системы всегда требуется привлечение данных из дополнительных источников, таких, например, как база данных IMDb (при работе с фильмами), базы данных об организациях, в которых могут работать и с которыми могут сотрудничать пользователи [7]. Если рекомендательная система использует алгоритмы коллаборативной фильтрации, то необходима информация о связи пользователя с другими пользователям, которая может быть представлена графом очень большой размерности. Если речь идет об использовании локального пространственного контекста пользователя в некоторой ситуации, то придется подключать большие базы данных о сервисах, доступных пользователю в конкретном локальном контексте и их провайдерах. В связи с тем, что интересы пользователя меняются со временем, данные о контенте, который интересует пользователя, могут иметь потоковый характер. Так как развитые рекомендательные системы учитывают контекст потребления того или иного контента, услуги или товара, то это потребует обрабатывать данные, описывающие время, место, социальное окружение, а иногда и эмоциональное и психическое состояние пользователя в момент потребления контента. В настоящее время для извлечения интересов пользователя активно используется такая дополнительная информация как тэги, которыми пользователь помечает потребляемый контент, товар, услугу в некоторой информационной системе. Анализ тэгов, в свою очередь, также требует привлечения дополнительных источников информации для анализа семантики, таких как семантические базы данных WordNet, дампы категорий Википедии, онтологические и поисковые компоненты DBpedia и т.п. Все вышесказанное относительно рекомендательных систем показывает, что данные, которые следует использовать для построения и динамического мониторинга профиля пользователя, обладают всеми свойствами больших данных.

Задача построения рекомендательных систем во многом объединяет в себе черты всех вышеперечисленных задач. Как было сказано выше, в таких системах обучающая информация о пользователе извлекается из всех доступных источников: его персональных страниц в социальных сетях, истории поиска по Интернет-сайтам и история покупок в Интернет—магазинах и т.п. Для анализа зачастую привлекается дополнительная информация, объёмы которой в сотни и тысячи раз превосходят изначальный объём данных о пользователе. Такие данные являются гетерогенными, могут обладать разной степенью достоверности, быть неполными. Отметим, что построение персонифицированной рекомендательной системы требует построения онтологии интересов пользователя рекомендательной системы. При этом построение этой онтологии должно выполнятся автоматически и принимать во внимание все разнообразие интересов пользователя. Так как интересы пользователя могут меняться во времени, а также в зависимости от контекста, в котором пользователь находится, задача принятия решений в этом случае может рассматриваться как задача анализа многомерных временных рядов.

При построении рекомендательных систем, использующих алгоритм коллаборативной фильтрации, зачастую необходим глубокий анализ графа связей пользователей. Таким образом, данные, с которыми приходится иметь дело при построении рекомендательных систем, по размерности, объему, разнообразию типов и динамике во времени обладают всеми свойствами данных, которые принято называть большими данными. По этой причине алгоритмические проблемы интеллектуальной обработки данных, которые необходимо решать при создании персонифицированных кросс–доменных контекстно–зависимых рекомендательных систем во многом сходны с алгоритмическими проблемами, которые возникают при создании других классов приложений, использующих большие данные как источник знаний. Поэтому модели и алгоритмы, которые используются при построении рекомендательных систем могут быть обобщены или адаптированы для использования в других классах приложений в области больших данных.

В настоящее время во всем мире ведутся активные работы в области интеллектуальной методов, моделей, алгоритмов и программных средств поддержки

процессов обработки больших данных. И хотя имеются уже десятки успешных приложений в данной области, разработаны инструментальные средства декомпозиции данных и т.п., достижения в области интеллектуального анализа больших данных пока далеки о желаемого уровня. Более того, соотношение ожиданий и достижений в этой области свидетельствует о наличии большого пробела между ними не в пользу достижений.

1.2 Современные средства обработки больших данных

Практические возможности современных методов и средств анализа больших данных в большой степени определяются возможностями современных компьютерных инфраструктур по параллельной обработке больших объемов данных и функционалов программных средств, реализующих эти процессы. Такие средства включают в себя программную платформу, которая управляет декомпозицией задачи, параллельной обработкой подзадач и сборкой результатов, а также программные средства интеллектуального анализа данных, в частности, анализа связей в данных и визуализации результатов.

1.2.1 Открытые платформы обработки больших данных

На сегодняшний день наиболее популярной платформой обработки больших данных является платформа Apache Hadoop. Она обладает передовыми возможностями и наибольшим числом удачных приложений, разработанных с ее помощью.

Apache Hadoop – это программная платформа для распределенных вычислений с открытым исходным кодом, реализованная на языке Java, которая находится под управлением компании Apache Software Foundation [8]. Платформа предназначена для создания и исполнения любых распределенных программ, однако, чаще всего ее название упоминается именно в контексте обработки больших данных. Hadoop реализует также возможности хранилища файлов, способное хранить огромные массивы данных.

Технически Hadoop состоит из двух ключевых компонент:

- распределенная файловая система Hadoop (англ. *Hadoop Distributed File System, HDFS*), которая отвечает за хранение данных на кластере Hadoop;

- система MapReduce, предназначенная для выполнения задач по обработке больших объемов данных на кластере.

На основе этих ключевых компонент создано несколько подпроектов, таких как Pig, Hive, HBase и т.д. Apache Pig – это платформа для анализа больших данных, которая включает в себя язык высокого уровня для написания программ анализа больших данных, хранящихся на кластере Hadoop, и инфраструктуру для их выполнения. Программное обеспечение Apache Hive предоставляет возможность создавать запросы к большим данным распределенного хранилища на языке HiveQL, сходном с языком SQL. Apache HBase – это открытая, распределённая, версионная, не реляционная СУБД. HBase позволяет работать с отдельными записями в реальном времени. Подробное описание перечисленных проектов может быть найдено в [9,10, 11].

Hadoop Distributed File System, или HDFS – это основная система хранения данных, используемая приложениями Hadoop. Особенность ее заключается в том, что она многократно копирует блоки данных (по 64 или по 128 Мб) и распределяет их по вычислительным узлам, тем самым обеспечивая высокую надежность и скорость вычислений. Благодаря избыточности, возникающей при этом, платформа продолжает работать, даже если какой-то из серверов выходит из строя. HDFS предназначена для потоковых считываний файлов, и файлы записываются в системе лишь однократно, так что внесение произвольных записей в файлы невозможно. При этом приложения Hadoop могут работать с файлами распределенной файловой системы через программный интерфейс Java.

MapReduce – это модель программирования и шаблон для написания приложений, предназначенных для высокоскоростной обработки больших объемов данных на параллельных кластерах вычислительных узлов. MapReduce, по сути, реализует подход «акщщяй и властвуй» (*англ*. divide and conquer), т.е. выполняет разделение данных на части и их обработку по частям параллельно на разных узлах (шаг «map») с последующим объединением результатов (шаг «reduce»). MapReduce обеспечивает автоматические распределение задач по узлам кластера,

а также обладает встроенными механизмами сохранения устойчивости и работоспособности в случае сбоя отдельных элементов системы. Одно из ключевых достоинств MapReduce состоит в том, что данные обрабатываются на том же узле, на котором они хранятся, что позволяет экономить время, затрачиваемое обычными алгоритмами на передачу данных по сети от узла к узлу.

В настоящее время многие компании используют Hadoop и MapReduce в исследовательских и производственных целях. Рассмотрим один из характерных примеров.

Социальная сеть Facebook на сегодня имеет десятки миллионов пользователей и более миллиарда просмотров в день. Практически с первых дней работы сеть Facebook столкнулась с необходимостью разработки масштабируемого средства хранения и обработки накопленной информации, так как с ее помощью можно существенно улучшить удобство использования Facebook и эффективнее решать множество других прикладных задач. Несколько лет назад в сети Facebook впервые начали использовать Hadoop и MapReduce для обработки и агрегирования накопленных данных. Сегодня Facebook имеет несколько развернутых кластеров Hadoop (самый большой имеет около 2500 процессорных ядер и 1 петабайт дискового пространства), ежедневно загружает в файловую систему Hadoop более 2 терабайт данных, которые обрабатываются сотнями процессов. Список задач, для решения которых используется созданная инфраструктура, включает формирование статистики об использовании сайта, борьбу со спамом, таргетирование рекламы и многое другое. Руководство компании Facebook считает решение о начале использования Hadoop одним из ключевых решений, так как созданная на этой основе инфраструктура позволила им весьма эффективно использовать накопленные данные о пользователях.

Отметим, однако, что у MapReduce есть существенный недостаток, который состоит в том, что результаты работы каждого вычислительного шага *map* и *reduce* записываются на диск. Так как большинство реальных задач для своего ре-

шения требуют несколько последовательных этапов MapReduce, то в итоге суммарное время считывания и записи на диск, затраченное при решении задачи, зачастую в много раз превышает время самих вычислений.

Описанный недостаток отсутствует у системы Apache Spark [12]. Apache Spark – это высокопроизводительное средство обработки данных, хранящихся в кластере Hadoop, спроектированное в университете Беркли. Spark тоже использует идею локальности данных, однако выносит большинство вычислений в память вместо диска. Ключевым понятием в Spark является RDD (*англ*. resilient distributed dataset) – указатель на «ленивую» распределённую коллекцию данных. Большинство операций над RDD не влечет какие-либо вычисления, а только создаёт обёртку над ними, «обещая» выполнить операции только тогда, когда понадобятся её результаты.

По сравнению с механизмом MapReduce Spark по заявлениям разработчиков обеспечивает в 100 раз большую производительность при обработке данных в памяти и в 10 раз - при размещении данных на дисках.

Spark может использоваться как в типовых сценариях обработки данных, похожих на MapReduce, так и для реализации специфических методов, таких как потоковая обработка, SQL, интерактивные и аналитические запросы, решения задач машинного обучения и работа с графами (рисунок 1.1). Программы для обработки данных могут создаваться на языках Scala, Java и Python. Spark после пребывания в инкубаторе с февраля 2014 года стал ведущим проектом компании Apache Software Foundation. Из компаний, которые используют Spark, можно выделить Alibaba, Cloudera, Databricks, IBM, Intel и Yahoo.

Компания Yahoo успешно использует Apache Spark в двух своих проектах: персонализация новостей для посетителей сайта и аналитика контекстной рекламы. В проекте по персонализации новостей компания использует алгоритмы машинного обучения, реализованные в библиотеках Spark, для того, чтобы выявить интересы отдельных пользователей, и для классификации новых новостей по мере их поступления с целью выяснить, какие типы пользователей будут в них заинтересованы.

Отметим, что перечисленные выше инструменты платформы Apache Hadoop служат для считывания больших данных, хранения и их подготовки для анализа, но не для самого анализа.

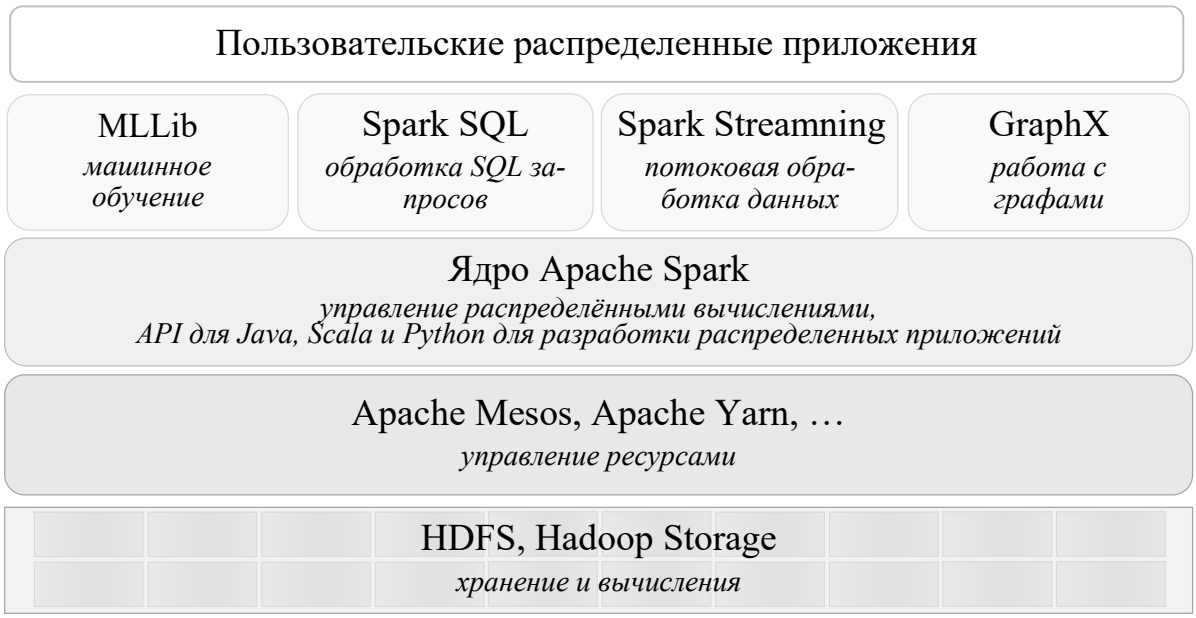


Рисунок 1.1 – Архитектура экосистемы Apache Spark

Наиболее популярными специализированными инструментами аналитики и машинного обучения, которые используются совместно с Hadoop, являются два крупных проекта:

– Mahout – это первая большая библиотека, реализовавшая алгоритмы кластеризации, коллаборативной фильтрации, случайных деревьев и примитивы для факторизации матриц средствами MapReduce;

– MLlib – библиотека, использующая для вычислений Apache Spark и реализующая алгоритмы базовой статистики, линейной и логистической регрессии, метод опорных векторов, кластеризацию *k*-средних и другие алгоритмы. Данный проект активно развивается и с каждым релизом значительно прибавляет в своем функционале.

1.2.2 Коммерческие программные средства обработки и анализа данных

Среди разработчиков коммерческих программных средств обработки и анализа больших данных следует выделить компанию IBM. Например, такой продукт, как InfoSphere BigInsights от IBM предоставляет собой набор аналитических

возможностей по работе с массивами неструктурированных и структурированных данных в их исходном формате. Этот программный продукт построен на основе платформы Apache Hadoop с использованием аналитических средств разработки IBM, включая, по заявлениям компании, сложную текстовую аналитику. Основные подсистемы платформы представлены на рисунке 1.2.

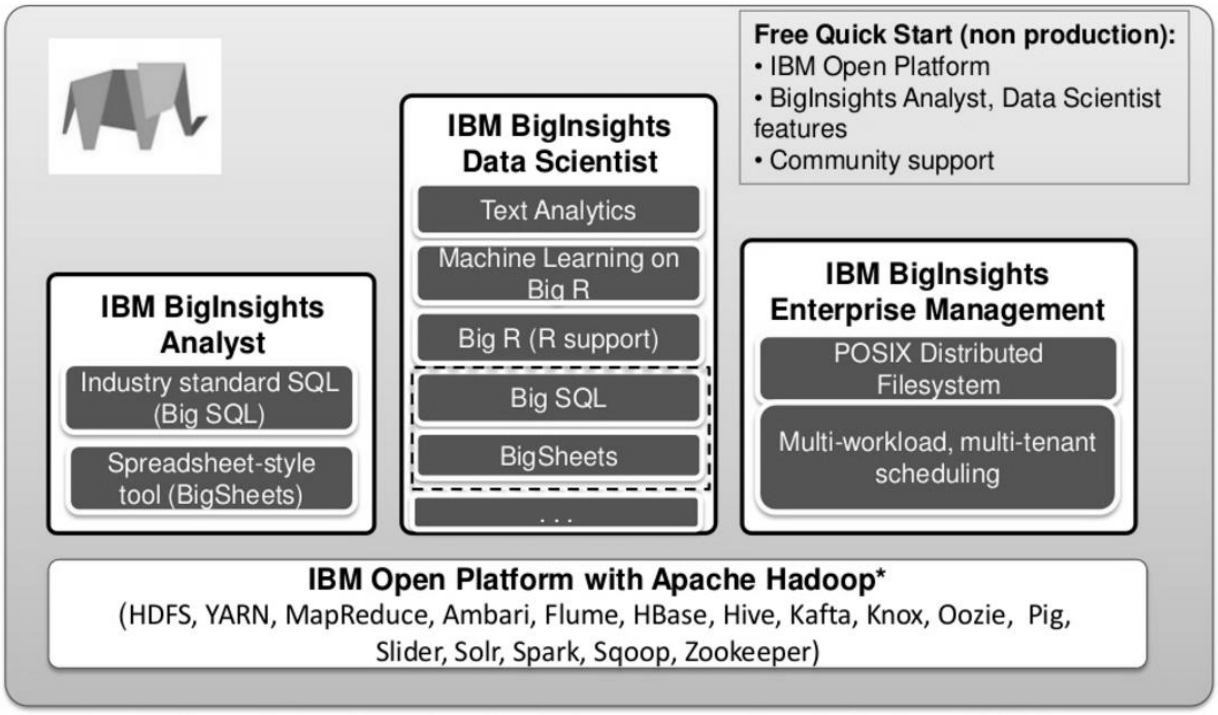


Рисунок 1.2 – Компоненты платформы IBM InfoSphere BigInsights (рисунок позаимствован из [13]

InfoSphere BigInsights позволяет использовать множество технологий компании IBM, которые расширяют возможности Hadoop и являются аналогами открытых проектов Apache Foundation. Как показано на рисунке 1.3, эти технологии включают средства повышения производительности приложений, средства аналитики, инструменты для разработки систем, средства интеграции с программным обеспечением предприятия, инструменты для администрирования и т.д.

В целом, аналоги всех средств платформы InfoSphere BigInsights, большая часть из которых является платной, могут быть найдены в экосистеме Hadoop Apache Foundation.

Программное средство InfoSphere Streams компании IBM предназначено для извлечения паттернов из потоковых данных в реальном времени [15]. Приложе-

ния, разработанные с помощью средства InfoSphere Streams, состоят из так называемых отдельных операторов, связанных между собой и работающих одновременно с несколькими потоками данных. Потоки данных могут поступать как извне, так и генерироваться внутри приложения. На рисунке 1.4 приведены различные типы операторов, которые предоставляет разработчику средство InfoSphere Streams и которые могут быть применены к потокам данных: фильтрация, классификация, преобразование, подсчет корреляции между потоками, слияние потоков и т.д. Программное средство InfoSphere Streams может объединять потоки и получать новые данные из нескольких потоков.

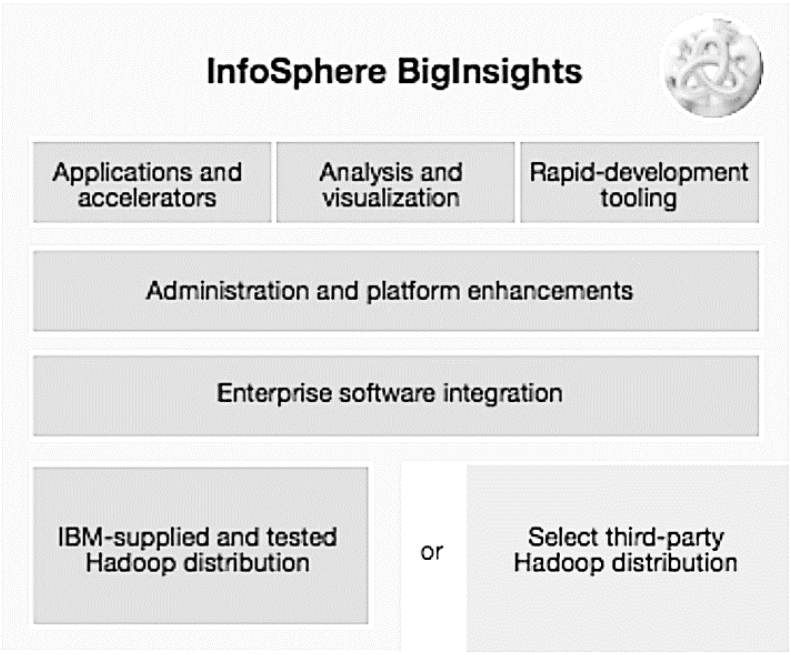


Рисунок 1.3 – Технологии IBM, входящие в состав InfoSphere BigInsights (рисунок позаимствован из [14])

Более подробную информацию о перечисленных и других средствах, рекламируемых в настоящее время в качестве средств поддержки аналитики больших данных, можно найти в [16, 17, 18].

Рассмотрим интересный и успешный пример использования средств IBM для работы с большими данными для решения крупномасштабных практических задач. Это приложение позволяет выполнять анализ потоков данных для оценки дорожного трафика в реальном времени на основе данных, собираемых с автомобилей, радаров, установленных вдоль дорог, данных о скоплениях машин у станций

оплаты, о погоде, о дорожных работах и инцидентах и т.п. Проект выполнен компанией IBM по заказу компаний KTN Institute и Royal Institute of Technology, Швеция. Система анализа дорожного трафика построена на базе компьютерных инфраструктур IBM, платформы Apache Hadoop и программного инструмента IBM®InfoSphere®Streams. Информация, получаемая в результате обработки этих данных, используется для оценки времени, которое потребуется тому или иному водителю для перемещения из текущего положения в заданное место, и для того, чтобы предлагать водителям разные маршруты и улучшить тем самым состояние дорожного движения в центре города.

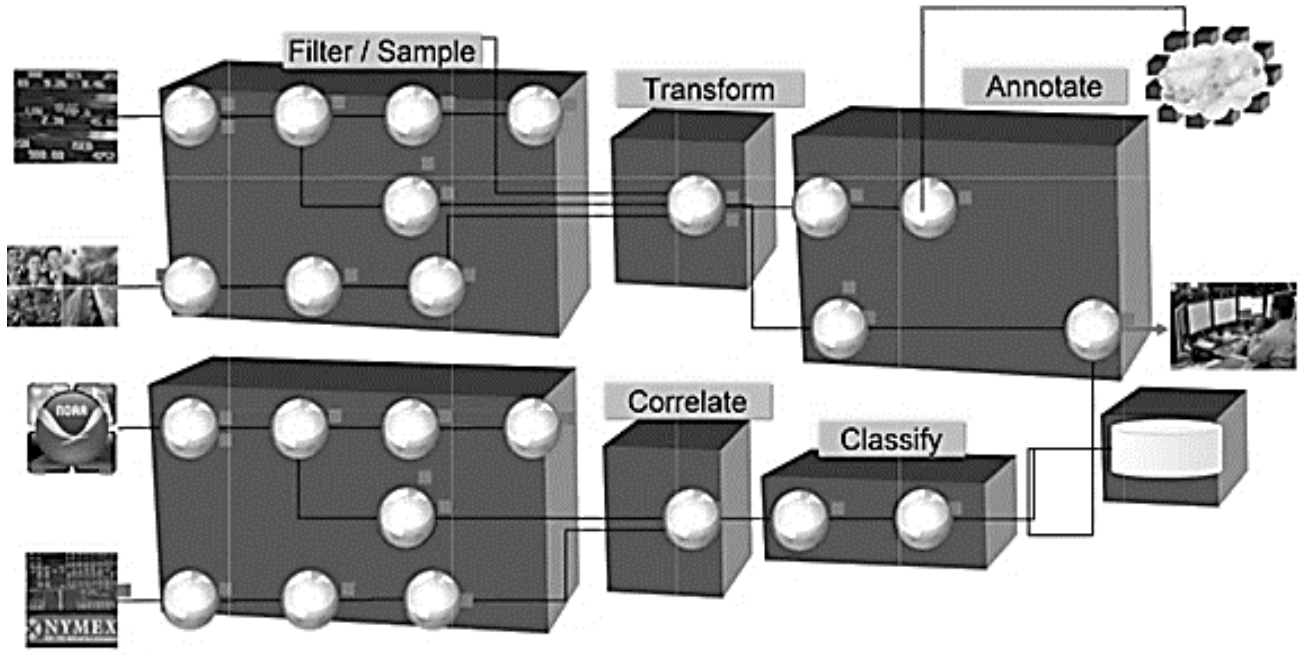


Рисунок 1.4 – Основные операторы InfoSphere Streams (рисунок заимствован из [17])

Описание нескольких других примеров успешного практического применения аналитических средств IBM, используемых совместно в платформой Apache Hadoop, приведено в [18].

Следует отметить, что все описанные выше и другие программные средства анализа больших данных, существующие в настоящее время, все ещё имеют несколько ограниченные возможности и, по сути, реализуют функции технологии OLAP–анализа данных. Для решения современных задач интеллектуального анализа больших данных этих возможностей недостаточно. Подчеркнем, что функ-

циональность существующих средств интеллектуального анализа данных обычной размерности и объема, не относящихся к типу больших данных, намного более развита и разнообразна.

1.3 Основные проблемы в области построения моделей принятия решений на основе больших данных

Как было отмечено выше, одной из ключевых задач интеллектуального анализа больших данных является поиск различных типов связей и зависимостей между атрибутами данных и их последующее использование в различных задачах оценивания, прогнозирования, моделирования и принятия решений. Среди атрибутов данных особую роль играют так называемые целевые переменные, которые либо представляют собой значения показателей качества, которые необходимо оптимизировать, либо являются именами классов в задачах классификации, либо имеют другой смысл, существенный для решаемой задачи анализа данных. В общем случае, основная задача анализа больших данных состоит в построении модели одной или нескольких целевых переменных в виде функций от других переменных – атрибутов данных или функций от атрибутов данных.

В литературе по обработке больших данных выделяют несколько критических проблем статистического анализа больших данных, порожденных высокой их размерностью и огромным объемом [19]. Опишем их кратко.

1. *Накопление ошибок* (англ. *noise accumulation*). Как известно, результаты измерений, которые представляются в виде атрибутов данных, практически всегда содержат ошибки. Например, в прикладных системах данные получаются от сенсоров, которые всегда содержат ошибки. В процессе обработки больших данных ошибки промежуточных и финальных вычислений нарастают и постепенно начинают доминировать над полезным сигналом. Особенно ярко это проявляется в случаях, когда алгоритмы обработки включают в себя, например, обращение матриц, поиск собственных значений и другие процедуры подобного типа. Естественно, что накопление ошибок всегда оказывает отрицательное влияние на результаты обработки больших данных. Проблема накопления ошибок и методы ее преодоления подробно рассмотрены в [20, 21, 22, 23].

Поясним негативную роль накопления ошибок на примере задачи классификации при большом исходном числе атрибутов (признаков), с использованием которых может строиться модель классификации [24]. Пусть имеются данные об экземплярах двух классов, представленные выборками с нормальными распределениями $X_1,...X_n \sim N_d(\mu_X, I_d)$ и $Y_1,...Y_n \sim N_d(\mu_Y, I_d)$, где I_d – единичная ковариационная матрица размером *d*, *n*=100 (объем выборки) и *d*=1000 (размерность пространства признаков) [4]. При этом все компоненты вектора математических ожиданий признаков μ_X для первого класса равны нулю. Для второго класса первые 10 компонент вектора математических ожиданий μ_Y равны 3, а остальные - 0. Пусть в задаче классификации во внимание принимаются только *m* наилучших признаков из 1000, а при классификации используется критерий максимума значения меры близости. При этом значение меры близости для некоторого входного вектора признаков вычисляется как сумма его проекций на первую и вторую главные компоненты соответствующего класса. Эти компоненты вычисляются по данным выборок обоих классов. Авторы работы [6] исследуют влияние накопления ошибок в этой задаче в зависимости от количества учитываемых признаков *m*. Эксперименты выполнены для *m=2, 40, 200* и *1000*. Соответствующие иллюстрации представлены на рисунке 1.5. Из этого рисунка видно, что при *m=2* представители разных классов хорошо разделяются линейной границей. Несколько хуже, они разделяются при использовании 40 наилучших атрибутов. При *m=200* и *m=1000* накопленная ошибка (ошибка измерений признаков и ошибка вычислений) уже значительно превосходит полезный сигнал, и классы оказываются неразделимыми в пространстве первых двух главных компонент.

Подобное явление является характерным и для других задач обработки больших данных. Таким образом, накопление ошибок может оказать негативное влияние на результаты обработки больших данных. Наиболее распространенным способом снижения отрицательной роли накопления ошибок является декомпозиция задачи обработки больших данных на подзадачи с последующим объединением решений подзадач. Для реализации этой идеи разработаны специальные инструментальные программные средства [8]. Однако для них острым является вопрос о

том, как именно найти наилучший способ декомпозиции пространства атрибутов данных и как затем корректно объединить решения, полученные для отдельных подпространств пространства атрибутов в единое решение. Эта задача в настоящее время имеет решения только для частных случаев, и она тоже формирует одну из тяжелых проблем обработки больших данных. В конце концов, обычно, требуется находить модели целевых переменных с заданной точностью, используя минимальное число атрибутов данных для того, чтобы в итоговой модели избежать вредного эффекта от накопления ошибок. Это еще одна важная и «тяжелая» проблема, которую приходится решать в области алгоритмизации процессов обработки больших данных.

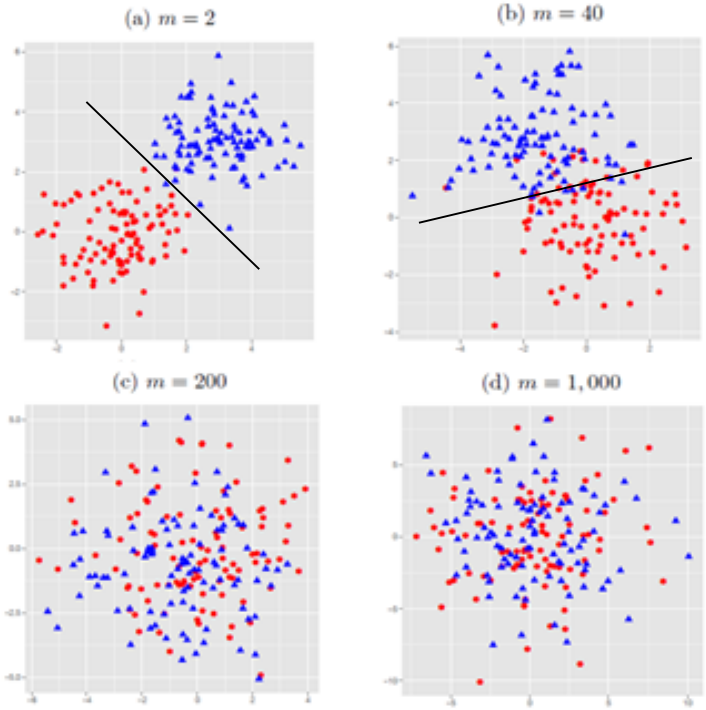


Рисунок 1.5 - Графики разброса проекций наблюдаемых данных на две главные компоненты подпространства размерности m (рисунок заимствован из работы [19])

2. *Появление ложных выборочных корреляций* (англ. *spurious correlations*) между переменными, которые реально являются независимыми.

В работе [24] приводятся результаты экспериментов по оценке выборочного коэффициента корреляции на множестве теоретически независимых переменных $X = \{x_1,...,x_d\}$, распределенных по нормальному закону $N(0, I_d)$, I_d – единичная

матрица. При больших значениях размерности вектора X выборочные корреляции этих переменных могут оказаться значительными, несмотря на то что теоретически они независимы. На рисунке 1.6 показано распределение максимального значения выборочного коэффициента корреляции переменной x_1 с другими теоретически независимыми переменными. Это распределение было построено для n= 60 при d = 800 и 6400 по 1000 реализаций для каждого случая размерности. Анализ приведенных результатов показывает, что ложные корреляции действительно возникают, причем их число, а также среднее значение коэффициента корреляции возрастают при возрастании размерности пространства независимых случайных переменных.

В результате появления таких ложных корреляций может быть сделан неправильный выбор переменных, задающих модель некоторой целевой переменной на основе измерений атрибутов данных. Помимо этого, ложные корреляции могут приводить к неверным статистическим выводам. В [24] показано, что при наличии большого количества переменных, ошибочно включенных в модель из-за ложных корреляций, оценки дисперсий коэффициентов модели оказываются слишком заниженными, что приводит к неверным выводам о статистической значимости выбранных переменных модели.

3. *Статистическая зависимость между помехой и переменными модели* (англ. *incidental endogeneity*).

Если рассматривать регрессионную модель

$$Y = A \cdot X + \varepsilon \tag{1.1}$$

где $X = \{x_1, ..., x_d\}$ – вектор переменных модели, то речь идет о корреляции между помехой ε и компонентами вектора атрибутов X. Большинство статистических моделей использует предположение о независимости помехи ε и переменных. В отличие от ложных корреляций, речь идет о реальном существовании зависимости между помехой и переменными модели. Высокая размерность больших данных способствует значительному возрастанию вероятности появления таких зависимостей. Возникновению зависимостей между переменными и ошибкой также

способствует то, что объединяемые данные могут быть получены из разных источников, в разное время и измерены с разной точностью. В настоящее время рассматриваемая проблема пока изучена слабо, однако имеются работы, например, [25, 26, 27], предлагающие альтернативные методы статистической обработки больших данных, которые работают при более слабых допущениях. В частности, такие методы предложены для решения задач линейной регрессии. Данная работа также предлагает вариант решения этой проблемы.

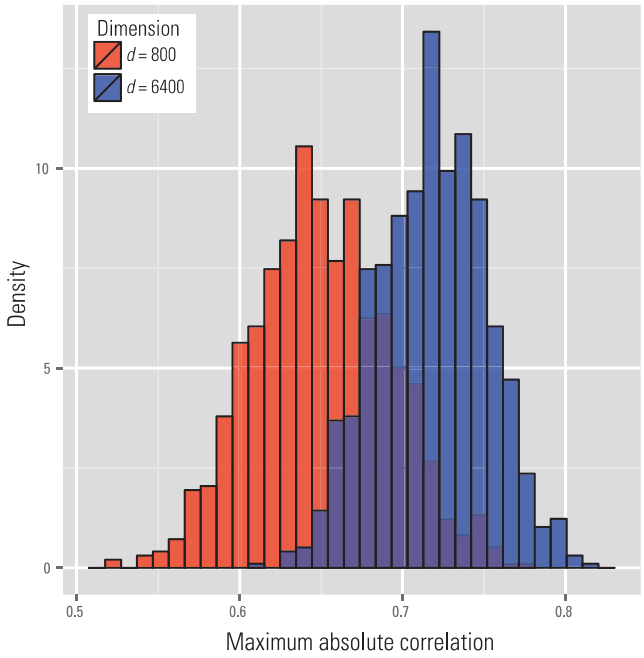


Рисунок 1.6 - Иллюстрация ложных корреляций: распределение максимума абсолютного значения выборочного коэффициента корреляции между x_1 и остальными переменными (график заимствован из работы [24])

4. *Значительные вычислительные затраты и вычислительная неустойчивость существующих алгоритмов анализа*. Большие объемы данных предъявляют повышенные требования к аппаратному обеспечению и вычислительным мощностям. Одним из решений этой проблемы в каком-то виде являются не слишком дорогостоящие кластеры с развернутой инфраструктурой Apache Hadoop. Кроме того, при поиске статистических зависимостей между переменными важно избегать использования вычислительно-неустойчивых алгоритмов для того, чтобы не допустить накопления ошибок.

Существуют и другие алгоритмические проблемы, которые существенно усложняют задачу анализа больших данных. Для преодоления этих трудностей

необходим пересмотр подходов и методов решения задач статистического анализа. Анализ больших данных требует использования более гибких и устойчивых процедур обработки. Эти процедуры должны быть, прежде всего, направлены на то, чтобы сбалансировать вычислительную эффективность, устойчивость и точность вычислений. Как уже было сказано выше, для этого в первую очередь необходимо снижать размерность больших данных, сохраняя при этом наиболее информативные переменные, которые далее будут использоваться в качестве атрибутов моделей целевых переменных решаемой проблемы.

Отметим, что авторы многих работ, например, [28], подчеркивают важность причинного анализа как основного метода минимизации числа переменных. В [2] также подчеркивается, что именно причинность событий в процессов представляет наибольший интерес при решении различных прикладных задач. Необходимы дальнейшие исследования и разработка таких моделей принятия решений на базе больших данных, которые позволят принимать более обоснованные решения в короткие сроки.

Авторы [2] отмечают, что исследователи в области алгоритмов анализа больших данных, в первую очередь, должны сосредоточиться на решении следующих задач:

– поиск и моделирование причинно-следственных связей в структурированных данных. Эта задача понятна и активно исследуется в области интеллектуального анализа данных;

– обнаружение и моделирование причинно-следственных связей в неструктурированных данных. Количество работа на эту тему в области искусственного интеллекта и машинного обучения постоянно растет. Однако, эта проблема пока слабо разработана и требует дальнейших серьезных исследований;

– интеграция неструктурированных и структурированных моделей причинности.

1.4 Большие данные и современные рекомендательные системы

Рекомендательными системами называют класс систем принятия решений, которые, используя разнородную информацию о предпочтениях человека в раз-

личных контекстах, пытаются спрогнозировать его предпочтения, реакцию на предложение некоторой внешней системы, рекомендующей ему некоторый контент, товар или некоторую услугу [29].

Исследования в области рекомендательных систем ведутся со второй половины 90-х годов прошлого столетия. Появление и развитие систем такого типа в то время было обусловлено задачами маркетинга. В настоящее время рекомендательные системы получили значительное теоретическое развитие и очень широко применяются на практике. Главной причиной этого интереса является возможность значительного повышения продаж товаров и услуг с помощью Интернет-торговли, а также широкое распространение мобильных устройств и приложений, которые поддерживают соответствующие функции.

Принято выделять три основных этапа развития рекомендательных систем в период с 1990-х годов и по настоящее время (рисунок 1.7).

Рекомендательные системы первого поколения зародились в середине 1990-х годов и, их теоретическое развитие фактически прекратилось к 2005 году. Однако, они до сих пор имеют широкое практическое применение. Модель типичной рекомендательной системы первого поколения основана на трех матрицах [30] (рисунок 1.8). Первая матрица $\mathbf{U}=\{u_{i,j}\}$ задает по строкам множество имен пользователей $\{c_1,c_2,...,c_N\}$, а по столбцам – множество имен атрибутов пользователей $\{x_1,x_2,...,x_L\}$. Таким образом, элемент $u_{i,j}$ матрицы **U** задает значение атрибута *j* для пользователя *i*. Вторая матрица $\mathbf{P}=\{p_{i,j}\}$ задает по строкам множество имен товаров/услуг $\{s_1,s_2,...,s_M\}$, а по столбцам – множество атрибутов (свойств) этих товаров/услуг $\{y_1,y_2,...,y_Q\}$. Наконец, третья матрица $\mathbf{R}=\{r_{i,j}\}$ имеет в качестве строк имена пользователей $\{c_1,c_2,...,c_N\}$, а в качестве столбцов – имена товаров/услуг $\{s_1,s_2,...,s_M\}$. Ее элемент $r_{i,j}$ задает значение рейтинга товара/услуги *j*, присвоенное ему пользователем *i*. Обычно матрица рейтингов **R** заполнена слабо, так как пользователь чаще всего в прошлом оценивал только небольшое количество товаров/услуг из их общего количества. Задача рекомендательной системы

состоит в том, чтобы спрогнозировать значение рейтинга $\ddot{r}_{i,j}$, который будет присвоен пользователем новому для него товару/услуге в матрице **R**. Еще раз обратим внимание на то, что интересы пользователя здесь представлены в матричной форме, и в модели рекомендательной системы надобности в использовании модели знаний, например, онтологии - нет.

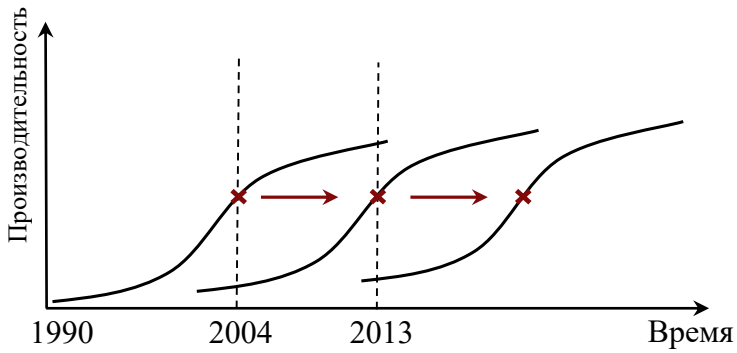


Рисунок 1.7 – Этапы развития рекомендательных систем (рисунок позаимствован из работы [30])

Выделяют три базовых метода принятия решений (рисунок 1.9), которые используются в рекомендательных системах первого поколения [30]. В методах, основанных на фильтрации контента, прогноз рейтинга пользователя для нового товара/услуги формируется по рейтингам других товаров/услуг, которые пользователь ранее оценивал и которые сходны с новым товаром/услугой. При этом для вычисления сходства товаров/услуг могут использоваться различные метрики [31]. Методы коллаборативной фильтрации основаны на предположении о том, что рейтинг пользователя для нового товара/услуги будет сходен с рейтингами других пользователей с похожими интересами по отношению к этому товару/услуге. В этом случае решающую роль играет сходство пользователей по рейтингам относительно одних и тех же товаров/услуг.

Существуют также гибридные методы, которые комбинируют оба названных выше подхода и, возможно, используют еще какие-то новшества. Отметим, что перечисленные методы являются статистическими и слабо учитывают персональные предпочтения конкретного пользователя. Пользователь в них характеризу

ется выборкой примеров товаров/услуг, которым он присвоил рейтинг в той или иной шкале. По этой причине предсказательная сила рекомендательных систем первого поколения оказалась достаточно слабой. Однако простота их разработки и программной реализации привели к их широкому практическом использованию, а полученный при этом коммерческий успех привел к тому, что и в настоящее время они, главным образом, распространены на практике.

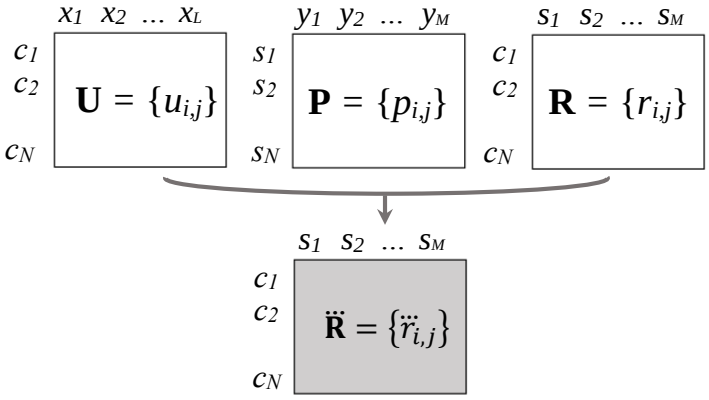


Рисунок 1.8 - Модель рекомендательной системы первого поколения

Рекомендательные системы второго поколения начинали развивать в первой половине 2000-х годов. Их важным отличием является учет контекста, в котором пользователю предлагается тот или иной товар/услуга. Иначе говоря, рейтинг пользователя зависит не только от свойств товара/услуги, но и от дополнительной переменной *C*, которая представляет контекст [32]:

$$\boldsymbol{R: U \times P \times C \rightarrow R,} \qquad (1.2)$$

В роли атрибутов контекста обычно выступают время, место, социальный контекст и даже эмоциональное состояние пользователя. Например, если пользователю рекомендуется фильм, то следует учитывать время (рабочий день или выходной день, время года и т.п.) и место (дом или кинотеатр) просмотра, а также коллектив, в котором он планирует просмотр (с детьми, с друзьями и т.п.).

Один из вариантов формального представления контекста использует аналогию с OLAP [32] (рисунок 1.10). В нем рассматривается многомерная модель данных, представляющих контекст. Каждое измерение D_1, …, D_n рассматривается как размерность, причем в множество измерений включаются также измерения

Пользователь и Продукт/услуга. Пространство представления контекстов вместе с измерениями Пользователь и Продукт рассматривается как прямое произведение всех измерений. В итоге контекстно-зависимое пространство решений (рекомендаций) определяется как $S = D_1 \times D_2 \times \ldots \times D_n$, а рейтинг определяется как отображение $R: D_1 \times D_2 \times \ldots \times D_n \rightarrow R$. Пространство контекстов в этом случае представляется гиперкубом S (множеством таблиц) (рисунок 1.10), а функция рейтинга – частичной функцией, в которой известно только некоторое начальное множество рейтингов, а большинство позиций не заполнено.

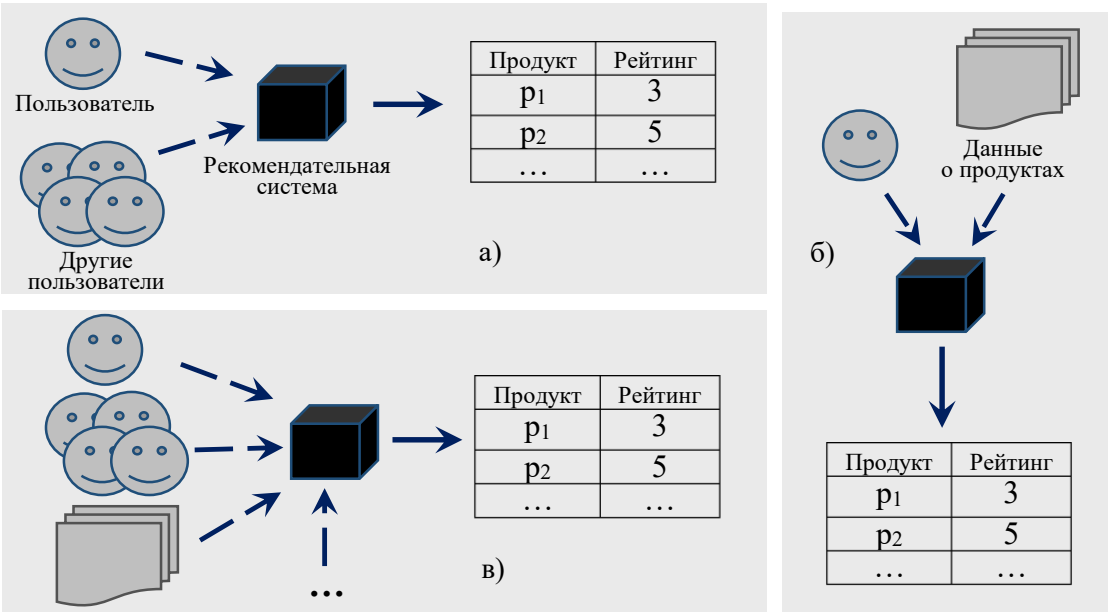


Рисунок 1.9 - Базовые методы принятия решений используемые в рекомендательных системах первого поколения: а) метод коллаборативной фильтрации; б) метод фильтрации контента; в) гибридный метод [Tuzhilin12]

Следует отметить, что ввиду слабого заполнения гиперкуба данными могут возникнуть серьезные проблемы при обучении рекомендательной системы, поскольку для реально появляющегося контекста в нем может не оказаться близких примеров, которые могли бы использоваться в процессах обучения. Аналогичные проблемы свойственны и онтологическому представлению контекста, когда он представлен совместно с интересами пользователя в рамках единой онтологии. В этом случае большинство узлов не будет иметь ни одного примера в обучающих данных, а потому поиск «близких» прецедентов будет сложной задачей.

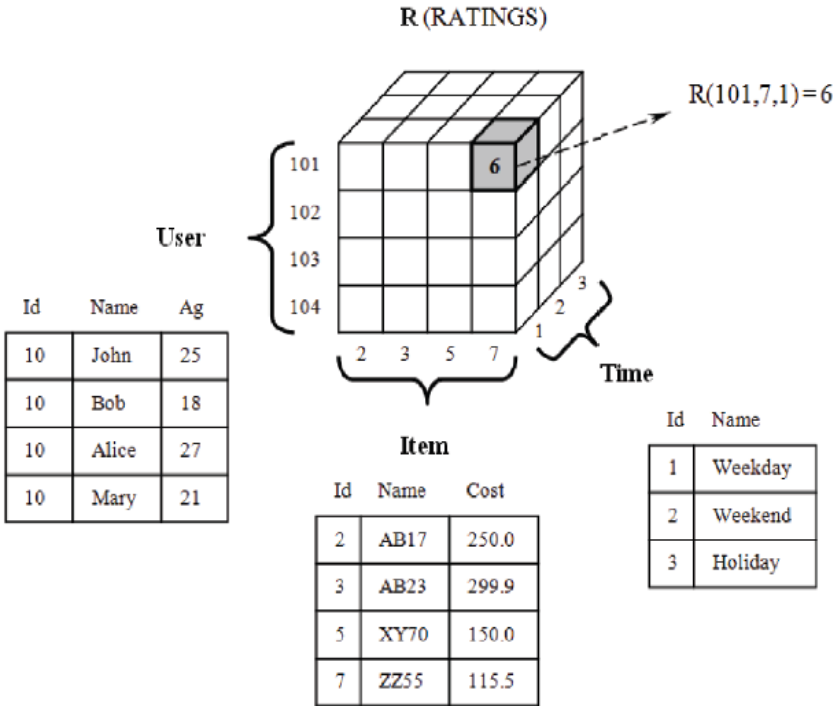


Рисунок 1.10 - Представление контекста в структуре гиперкуба (рисунок заимствован из [34])

В работе [33] рассматриваются три метода учета контекста, которые используются в настоящее время. Они различаются этапом, на котором контекст «встраивается» в процесс выработки рекомендаций.

1. Метод предварительной фильтрации контекста (*англ*. context pre-filtering). Заданный контекст используется при выборе данных, которые в дальнейшем будут использованы для выработки рекомендаций обычными методами (см. рисунок 1.9). Иными словами, информация о контексте используется для выбора релевантных записей в данных.

2. Контекстуальная «пост–фильтрация» (*англ*. contextual post-filtering). В этом варианте рекомендации (подсчет рейтингов продуктов) вырабатываются без учета контекста, однако после выработки множества рекомендаций они уточняются с помощью контекста каждого пользователя.

3. Моделирование контекста (*англ*. contextual modeling) или встраивание контекста в функцию выбора (*англ*. contextualization of recommendation function). В этом случае учет контекста выполняется уже в процессе подсчета рейтингов продуктов/услуг.

Рисунок 1.11 демонстрирует эти варианты учета контекста графически.

Детальный анализ сущности контекста, моделей его формального представления и учета в процессах выработки рекомендаций можно найти в [29, 32, 33, 34].

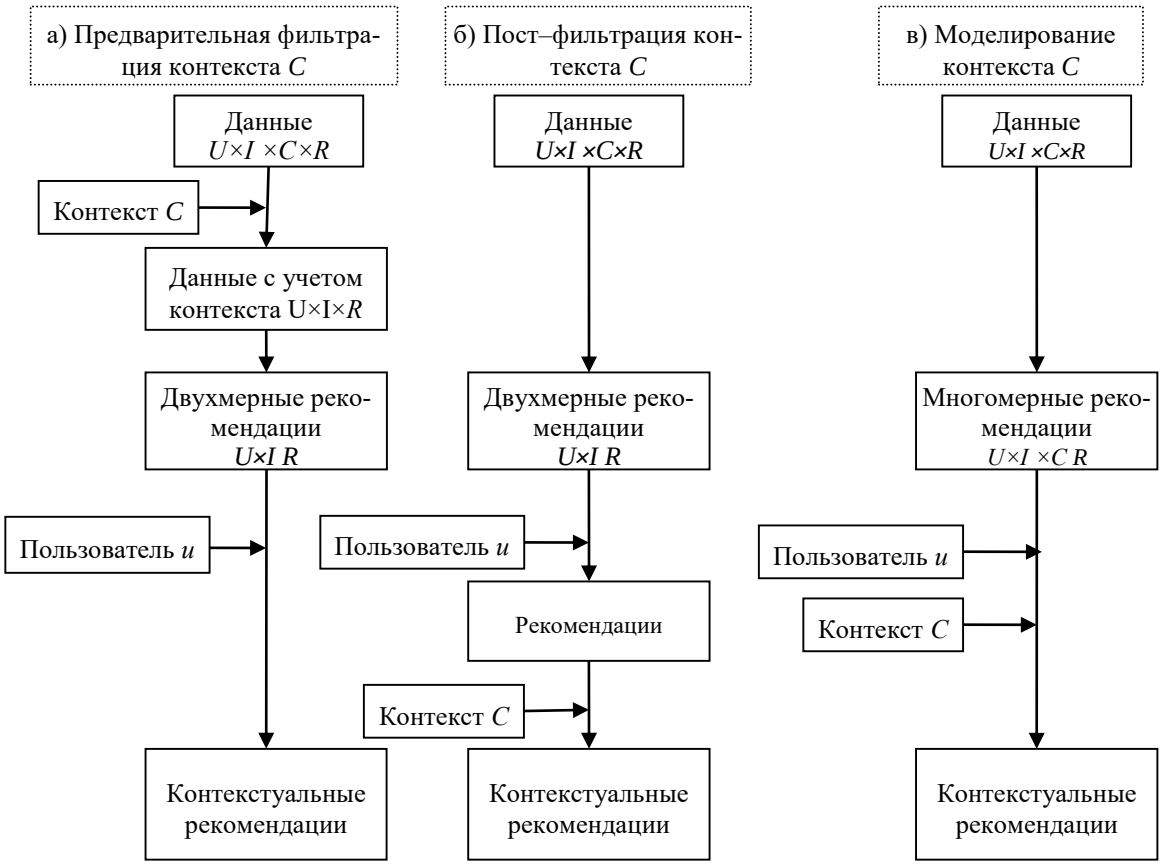


Рисунок 1.11 - Варианты учета контекста в контекстно–зависимых рекомендующих системах (в основу положен рисунок из работы [33])

Среди ключевых проблем контекстно–зависимых рекомендательных систем второго поколения авторы работы [34] упоминают необходимость тщательного исследования различных моделей встраивания контекста в процесс поиска рекомендаций. Сравнительный анализ достоинств, недостатков и возможностей предложенных методов учета контекста при поиске рекомендаций, фактически отсутствует, исследование этих вариантов проведено весьма поверхностно. Не менее важным является вопрос о разумной комбинации этих методов учета контекста.

Требует дополнительного исследования задача категоризации контекстов и возможности автоматизации построении онтологии контекстов. Необходимо более глубоко изучить корректность использования процедур обобщения контекста,

а также построить различные топологические метрики оценки близости контекстов, что может помочь более эффективно преодолевать дефицит данных, доступных для обучения контекстно–зависимых систем.

Следует обратить внимание на то, что все проблемы вычислительной эффективности и качества решений, которые возникают в области интеллектуальной обработки больших данных, полностью переносятся и на область контекстно–зависимых рекомендательных систем, поскольку процессы поиска рекомендаций зачастую связаны с интеллектуальной обработкой больших данных.

Рекомендательные системы второго поколения могут обладать и другими новыми возможностями, например, использовать многокритериальные рейтинги и рейтинги, формируемые для групп пользователей. Такие системы могут вырабатывать рекомендации в контексте социальных сетей, использовать тэги в качестве источника информации о профиле пользователя. В рекомендательных системах второго поколения знания об интересах пользователя и контексте выработки рекомендаций играют уже большую роль. Онтологии становятся в таких системах неотъемлемой части модели знаний, используемой в рекомендательных системах второго поколения.

Наиболее эффективные практические приложения в области рекомендательных систем в настоящее время соответствуют именно второму их поколению. Активные научные исследования в интересах развития таких систем активно продолжаются, однако это развитие в большей мере ведется «вширь» в том смысле, что постоянно расширяется множество классов приложений и типов рекомендательных систем, о которых появляются публикации. Однако, для исследователей особый научный интерес представляют сегодня рекомендательные системы третьего поколения [30]. Такие системы, ориентируются на семантические аспекты персонального профиля пользователя и его интересов. В них большое значение придается мотивации пользователя, т.е. причинам, которые определяют тот или иной выбор пользователя. Рекомендательные системы третьего поколения должны быть в состоянии давать объяснения выбора пользователя, его мотива-

цию. При этом они должны учитывать экономические, психологические и другие факторы, определяющие выбор пользователя [30].

Таким образом, на первый план выходит представление модели интересов и предпочтений пользователя в терминах семантических понятий естественного языка. При этом онтологии стали рассматриваться в качестве семантической структуры для представления всех компонент знаний рекомендательной системы третьего поколения. В соответствии с этим взглядом онтология должна объединять в себе знания о предметной области рекомендаций и о персональном профиле интересов пользователя. Контекстная зависимость профиля интересов пользователя в онтологической концепции рекомендательных систем также должна быть представлена в онтологической модели знаний. Онтология должна описывать кросс-доменные связи, необходимые для выработки кросс–-доменных рекомендаций (например, имея информацию об интересах пользователя в области музыки, формировать рекомендации в области кинофильмов, художественной литературы или новостей). Подчеркнем, что главное здесь то, что интересы пользователя (и кросс-доменные, и контекстно-зависимые) рассматриваются в рекомендательных системах третьего поколения как ее подструктуры в иерархии понятий онтологии, а поиск интересов пользователя сводится к поиску таких подструктур онтологии. При этом онтология, в которой нужно найти подструктуры, описывающие интересы пользователя, в свою очередь, строится «от данных конкретного пользователя», которые вовлекаются в процесс построения рекомендательной системы.

Следует отметить, что в рекомендательных системах третьего поколения при построении профиля интересов и предпочтений пользователя используется, как правило, вся доступная информация о нем, полученная из разнородных источников, в которых, так или иначе, отражаются «следы» присутствия или деятельности пользователя, таких как социальные сети (Facebook, LinkedIn, Twitter и т.д.), история браузера, история покупок и т.п. При этом, дополнительно для обогащения информации обычно используются облачные ресурсы и глобальные онтологии, такие как WordNet, Wikipedia, DBpedia и другие ресурсы в, в частности, Linked

Data Web [35]. С учетом этого данные, используемые для построения рекомендательной системы третьего поколения, очевидно относятся к классу больших данных.

Вышесказанное позволяет сделать вывод о том, что проблемы вычислительной эффективности и качества решений, которые возникают в области интеллектуальной обработки больших данных, во многом характерны и для приложений в области рекомендательных систем третьего поколения. В обоих случаях необходимо решать задачу обработки распределенных гетерогенных данных большого объема и размерности, в том числе данных, представленных на естественном языке. Более того, в большинстве случаев именно данные на естественном языке представлены в источниках данных о пользователе. В задачах построения рекомендательных систем, как и в других задачах в области больших данных, важной является проблема автоматизации построения онтологий.

В данной работе предложенные методы, модели и алгоритмы обработки больших данных, а также разработанные программные компоненты и системы, демонстрируются на тестовых примерах (англ. *benchmarks*), представленных наборами данных для тестирования разработок в области рекомендательных систем. Однако это не означает, что полученные результаты пригодны только для обработки больших данных, используемых для разработки рекомендательных систем. Естественно, что рекомендательные системы являются только одним из примеров приложений, построенных с использованием интеллектуальной обработки больших данных, а полученные результаты применимы и ко многим другим типам систем, которые строятся на основе извлечения знаний из больших данных.

1.5 Методы ассоциативного и причинного анализа в задачах принятия решений на больших данных

Как было отмечено в предыдущем разделе, семантическая модель больших данных есть модель знаний о различных предпочтениях пользователя, представленная иерархией понятий онтологии, в которой представлены не только знания о том, что он «предпочитает», но и другие знания о нем, о разных контекстах его интересов, о том, что ему совершенно неинтересно и прочие знания. На этой

структуре знаний необходимо решать различные задачи, которые могут формулироваться для построения тех или иных конкретных рекомендаций, для оценки информативности различных понятий онтологии в тех или иных ситуациях, для поиска интересов пользователя, которые являются общими или, наоборот, различными для различных пользователей или их групп и т.д. Постановок задач может быть множество. Но все эти задачи обычно формулируются как задачи классификации. Теоретический базис большинства задач классификации, которые строятся на основе данных, интерпретированных, не интерпретированных или частично интерпретированных, являются различные методы установления связей между переменными данных и метками классов, рассматриваемых в конкретной постановке задачи. Начиная с 1990-х годов большое внимание уделяется поиску ассоциативных связей между различными переменными, числовыми, дискретными, категориальными, между семантическими понятиями естественного языка в конкретном контексте, представленном большими данными. Смысл ассоциативной связи может быть различным, и это обычно становится ясным из постановки задачи. Например, в традиционном виде [36] ассоциативные правила имеют смысл условной вероятности заключения при истинности посылки ассоциативного правила. Иногда ассоциации формулируются как импликации, как это иногда бывает в вероятностных логиках, и тогда алгоритмы поиска таких правил будут иметь иной вид.

В данной работе рассматриваются ассоциации переменных в смысле классических работ по ассоциативным правилам [36]. При этом эта проблема поиска рассматривается для конкретного частного случая, а именно для построения ассоциативных правил классификации, которые необходимо строить, например, на множестве иерархии понятий онтологии. Частным случаем таких правил являются причинные правила ассоциативной классификации, которые, как отмечалось ранее, являются наиболее ценными для рекомендательных систем третьего поколения, поскольку именно такие правила объясняют мотивацию того или иного выбора, что является принципиально важным для объяснения рекомендаций таких систем. Поэтому далее рассматривается сначала само понятие ассоциативных пра-

вил классификации по сравнению с общим видом ассоциативных правил, затем анализируются их достоинства, а также проблемы, которые возникают при попытках их использования для построения систем классификации в общем случае больших данных. Следующим шагом поиска интересов пользователя на основе иерархии понятий онтологии, построенной для конкретного набора данных, является выделение из них множества причинных ассоциаций (правил), и в данном разделе еще раз акцентируется внимание на важности поиска именно причинных ассоциативных правил.

Изначально исследования в области анализа ассоциаций были, в основном, ориентированы на удовлетворение практических потребностей в маркетинге и направлены на решение таких задач, как анализ покупательской корзины, прогнозирование спроса на товары, анализ сезонности покупательского спроса и т.п.

Однако в настоящее время все больше возрастает интерес к ассоциативному и причинному анализу именно в области больших данных. Такие методы очень хорошо подходят для использования в задачах принятия решений на больших данных описанных выше, например, для прогнозирования общественного мнения в социологии, политологии, идентификации мошеннических операций с кредитными картами, для рекомендации сервисов, при обучении привычкам пользователя в «интеллектуальном доме», в задачах медицинской диагностики, анализе и предсказании сбоев в работе различных систем и во многих других прикладных областях.

Одним из активно развиваемых направлений практического использования ассоциаций является ассоциативная классификация, которая имеет целью поиск в данных ассоциаций между атрибутами данных и некоторой дискретной или непрерывной целевой переменной, рассматриваемой в качестве метки класса. Однако в отличие от традиционной интерпретации ассоциации как меры статистической связи между переменными, в задачах ассоциативной классификации рассматриваются, прежде всего, причинные связи. Иначе говоря, обнаруженные ассоциативные правила или сформированные на их основе классификаторы должны отражать не просто статистические зависимости некоторых атрибутов, но именно

причинно-следственные связи, существующие в данных. Причинная обусловленность моделей принятия решений позволяет сократить размерность данных, сохраняя при этом наиболее информативные переменные, что в свою очередь позволяет значительно увеличить точность прогноза и избежать типичных проблем обработки больших данных, о которых говорилось выше.

Методы ассоциативного и причинного анализа оказываются удобными для того, чтобы справиться с особенностями, характерными для больших данных вообще. Наиболее известными в этой области являются работы [28, 36, 37, 38, 39]. В работах [28, 39] подводится, фактически, промежуточный итог исследований в этом направлении на 2010 г. В этой работе показывается, что причинный анализ данных фактически предлагает новый подход к решению классической задачи анализа данных, в частности, задачи поиска и отбора признаков. В свою очередь, хорошо известно, что эта проблема является базовой во многих прикладных задачах принятия решений.

Анализ причинных правил способствует лучшему пониманию устройства естественных и искусственных систем, помогает определить, с помощью каких воздействий можно добиться желаемого поведения системы, или изменения ее свойств в нужную сторону.

Можно утверждать, что использование ассоциативного анализа данных для построения моделей принятия решений при работе с большими данными, в частности, ассоциативной классификации на основе причинных связей представляется достаточно перспективной идеей. Эта интеграция, сможет дать новый толчок эффективному решению проблемы обучения в задачах классификации и синтеза классификаторов применительно к большим данным.

1.6 Выводы: формулировка цели и задач исследования

В предыдущих разделах было кратко описано состояние исследований и разработок в области больших данных, а также проблемы и вызовы, которые исследователям в этой области необходимо решать уже сегодня. Перечислим основные из них:

1. Несоответствие задач, которые могут быть решены с помощью существующих средств, ожиданиям, которые выдвигаются потребностями приложений в области обработки больших данных.

Большинство традиционных интеллектуальных и статистических методов анализа данных, которые, в частности, базируются на выявлении и анализе связей между атрибутами данных, напрямую не могут быть использованы для работы с большими данными, в связи с вычислительной сложностью.

2. Отсутствие эффективных методов автоматического построения онтологии.

Важной задачей является семантическое обогащение больших данных. Семантически богатое представление больших данных, а, значит, и результатов их обработки должно позволить получить семантически ясные интерпретации получаемых закономерностей. В роли такого представления, в соответствии с существующими в настоящее время представлениями, должна выступать онтология, которую необходимо использовать в качестве метамодели данных. Обычно онтология представляется иерархией классов понятий (категорий) предметной области, каждому из которых поставлено в соответствие некоторое множество атрибутов. Кроме того, на множестве понятий онтологии задаются и другие типы отношений. Построение онтологии для больших данных является достаточно трудоемкой задачей, если построение выполняется вручную. Поэтому важным является привлечение методов автоматизации построения онтологий, которые позволили бы снизить нагрузку на экспертов.

3. Необходимость разработки новых эффективных и вычислительно устойчивых методов и алгоритмов анализа больших данных и моделей принятия решений с их использованием, а также гибких и устойчивых средств их компьютерной поддержки, обеспечивающих баланс между эффективностью, устойчивостью и точностью вычислений, а также высокое качество процессов обучения и принятия решений на основе таких данных.

Для выполнения этих требований, как уже было отмечено выше, в первую очередь необходимо снижать размерность больших данных, сохраняя при этом

наиболее информативные переменные, которые будут затем использованы для моделирования задачи принятия решений.

К числу перспективных методов в этой области относят большую группу методов, которые объединяются общим названием ассоциативный и причинный анализ данных. В этих методах акцент делается именно на выделение и анализ связей между атрибутами, задающими данные.

4. Необходимость разработки масштабируемых методов и алгоритмов поиска причинных зависимостей в данных.

Следует подчеркнуть, что имеются теоретические и экспериментальные работы, в которых строго показано, что среди ассоциативных связей наиболее полезными с точки зрения информативности в задачах принятия решений являются связи причинного характера [28, 39].

Однако поиск причинных структур, традиционно используемый в причинном анализе, и основанный на построении и анализе байесовских сетей доверия [40], требует решения задач обучения экспоненциальной сложности относительно числа атрибутов [28]. Поэтому использование такой модели в задачах причинного анализа больших данных бесперспективно. Стоит также отметить, что авторы [41], считают, что в байесовской сети, построенной автоматически на некоторых данных, далеко не все выявленные связи между переменными являются причинными. Эти обстоятельства делают актуальной задачу разработки альтернативных методов поиска метрик оценки и методов поиска причинных связей в данных.

Использование методов ассоциативного и причинного анализа в области больших данных может дать новый толчок в направлении эффективного решения проблем принятия решений применительно к большим данным.

Целью работы является разработка алгоритмов обучения и принятия решений в задачах классификации на основе семантических и причинных моделей больших данных, их реализация в форме программного прототипа, а также экспериментальная оценка по таким характеристикам как масштабируемость, вычислительная эффективность и точность в задачах принятия решений, в частности, в рекомендательных системах.

В соответствии с поставленной целью в работе сформулированы и решены следующие частные задачи исследования:

– теоретически и экспериментально обоснованный выбор семантически корректной и вычислительно эффективной формальной меры, оценивающей «силу» причинной связи атрибутов данных.

– разработка алгоритма автоматической генерации семантической модели больших данных, понятия которой используются для представления знаний о данных и результатов обучения в форме причинных моделей для принятия решений на основе ассоциативно-причинной классификации.

– разработка единой структуры для представления синтаксиса и семантики больших данных, а также метаинформации о них. Структура должна обеспечивать доступ к данным и упрощать вычисления различных статистик.

– разработка математически корректного, масштабируемого и вычислительно эффективного алгоритма поиска причинных связей в больших данных.

– разработка масштабируемого алгоритма минимизации размерности причинной модели принятия решений и механизма ассоциативно-причинной классификации.

– экспериментальная оценка разработанных алгоритмов на стандартных наборах данных из области рекомендательных систем третьего поколения.

Результаты решения перечисленных задач – это компоненты многошагового алгоритма генерации семантических моделей больших данных для решения задач ассоциативно-причинной классификации. Их разработка, интеграция в рамках единого процесса, программное прототипирование и экспериментальное исследование с использованием стандартных наборов данных из области рекомендательных систем составляют содержание работы.

2 АССОЦИАТИВНЫЕ И ПРИЧИННЫЕ МОДЕЛИ КЛАССИФИКАЦИИ

В разделе 1.6 данной работы показано, что одним из наиболее перспективных направлений в области анализа больших данных, является группа методов под общим названием *ассоциативный и причинный анализ данных*. Эти методы специализируются прежде всего на выделении и анализе связей между атрибутами, задающими данные, и, помимо всего прочего, могут быть использованы для сокращения размерности больших данных и выявления наиболее информативных переменных применительно к конкретной прикладной задаче.

В данной главе более подробно рассмотрена задача ассоциативной классификации, которая является базовой в области ассоциативного и причинного анализа. В первую очередь необходимо оценить текущее состояние разработок в области ассоциативной классификации и проанализировать существующие алгоритмы в точки зрения эффективности их применения в области анализа больших данных.

В данной главе будут рассмотрены модели, методы и алгоритмы ассоциативной классификации, ориентированные на обработку данных большого объема, и выполнен их сравнительный анализ, а также выделены основные преимущества ассоциативно-причинного анализа и особенности его использования при обработке больших данных. В главе решена одна из задач исследований, сформулированная в разделе 1.6, именно задача выбора наилучшей формальной меры, оценивающей «силу» причинной связи между переменными, которая является семантически корректной и вычислительно эффективной;

2.1 Общая постановка задачи ассоциативной классификации

Задачу поиска ассоциативных правил, в общем случае, относят к задачам поиска связей в данных (англ. *data mining*), в которых не конкретизируется приложение, где эти правил будут использованы. В отличие от этого, в ассоциативной классификации прикладная задача определена, поэтому она относится к области машинного обучения (англ. *machine learning*). Кроме того, в правой части правил ассоциативной классификации присутствует целевая переменная, а именно *метка класса*, что существенно снижает сложность поиска ассоциативных правил классификации. Такие правила принято называть *ассоциативными правилами класса*

(англ. *class associative rules,* CARs). Рассмотрим формальную постановку задачи ассоциативной классификации.

Пусть $\boldsymbol{D}$ – транзакционная база данных (множество данных), $\boldsymbol{D_i} \in \boldsymbol{D}$ – произвольная транзакция, $\boldsymbol{X}$ – множество всех идентификаторов (символов, символьных строк), которые используются для обозначения объектов (признаков, атрибутов) в транзакциях множества $\boldsymbol{D}$, $\boldsymbol{A}$ – подмножество идентификаторов из множества $\boldsymbol{X}$ и $\boldsymbol{D(A)}$ – подмножество множества транзакций из множества $\boldsymbol{D}$, каждая из которых содержит подмножество идентификаторов $\boldsymbol{A} \in \boldsymbol{X}$. Пусть даны два набора идентификаторов (объектов) $\boldsymbol{A} \in \boldsymbol{X}$ и $\boldsymbol{B} \in \boldsymbol{X}$, причем $\boldsymbol{A}$ и $\boldsymbol{B}$ не имеют общих элементов, и пусть σ и γ – вещественные числа из интервала [*0, 1*]. Говорят [42, 43], что выражение вида $\boldsymbol{A} \rightarrow \boldsymbol{B}$ есть *ассоциативное правило* с *порогом уверенности* $\boldsymbol{conf(A \rightarrow B) = \gamma}$ и *порогом поддержки* $\boldsymbol{supp(A \rightarrow B) = \sigma}$ ($\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$ – ассоциативное правило), если справедливы следующие неравенства

$$supp(\boldsymbol{A} \rightarrow \boldsymbol{B}) = n_{AB} / n \geq \sigma, \tag{2.1}$$

$$conf(\boldsymbol{A} \rightarrow \boldsymbol{B}) = n_{AB} / n_A \geq \gamma, \tag{2.2}$$

где n – общее количество транзакций в множестве $\boldsymbol{D}$; n_A - количество транзакций в подмножестве $\boldsymbol{D(A)}$; n_{AB} – количество транзакций в множестве $\boldsymbol{D}$, которые содержат объединение множества объектов подмножеств $\boldsymbol{A}$ и $\boldsymbol{B}$. Модель ассоциативного правила, заданную условиями (2.1), (2.2), принято называть моделью типа *поддержка–уверенность*.

Подмножество (последовательность) элементов $\boldsymbol{A}$ принято называть посылкой ассоциативного правила $\boldsymbol{A} \rightarrow \boldsymbol{B}$, а подмножество (последовательность) $\boldsymbol{B}$ – его следствием. Иногда эти последовательности называют паттернами (англ. *patterns*). В задачах ассоциативной классификации заключение правила может содержать только один идентификатор, который является именем (меткой) одного из классов. Поэтому в общем случае основная подзадача задачи ассоциативной классификации сводится к поиску множества ($\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$)–ассоциативных правил для каждого класса. Эта подзадача называется обычно задачей *обучения* классификатора.

Другая подзадача – это синтез классификатора на множестве найденных ассоциативных правил.

Отметим, что ассоциативное правило задает *статистическую зависимость* между посылкой правила $\boldsymbol{A}$ и его следствием $\boldsymbol{B}$, если последовательность идентификаторов $\boldsymbol{A}$ рассматривать как сложное случайное событие, а идентификатор $\boldsymbol{B}$ рассматривать как случайное событие, когда оба они заданы выборкой, представленной в транзакционной базе данных. В этом случае числа $\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$ являются статистическими (эмпирическими) оценками двух вероятностей. Величина $\boldsymbol{\sigma}$ является оценкой вероятности $p(\boldsymbol{AB})$ появления подпоследовательности (подмножнства) идентификаторов $\boldsymbol{AB}$ в транзакциях базы данных $\boldsymbol{D}$, а величина γ является оценкой вероятности появления идентификатора $\boldsymbol{B}$ в транзакциях этой же базы, в которых появилась и подпоследовательность $\boldsymbol{A}$, т.е. $\boldsymbol{\gamma}$ является оценкой условной вероятности $p(\boldsymbol{B/A})$. Подчеркнем, что семантика отношения, задаваемого ассоциативным правилом, является иной, чем семантика отношения, задаваемого импликацией в вероятностной пропозициональной логике или отношением выводимости, задаваемым в аналогичном пропозициональном исчислении.

Классические методы поиска ассоциативных правил используют модель *Apriori* [38], которая является переборной с механизмом отсечения, основанном на свойстве антимонотонности вероятности появления паттерна (его поддержки) по мере увеличения его длины. Более эффективными являются алгоритмы, известные под названием *FP-growth* [44]. Заметим, что если обучающие данные представлены не в транзакционном виде, то для использования названных алгоритмов потребуется некоторое дополнительное преобразование исходных данных в транзакционную форму.

Понятие ассоциативного правила, введенное условиями (2.1) и (2.2), обладает большим недостатком, так как оно не учитывает возможную вероятностную независимость паттернов $\boldsymbol{A}$ и $\boldsymbol{B}$. В этом случае, говорить о существовании ассоциации не имеет смысла. Действительно, можно столкнуться с ситуацией, когда меры поддержки и уверенности будут достаточно большими за счет больших значений

вероятностей компонент паттернов. В результате может быть сделано ошибочное заключение о существовании ассоциативной связи между этими паттернами.

Действие отмеченного недостатка модели *поддержка–уверенность* (2.1), (2.2) ослаблено в модели *поддержка–уверенность–зависимость*, предложенной в [45]. Дополнительно к мерам поддержки и уверенности, в ней вводится еще один параметр для выбора правила, использующий точечную статистическую меру зависимости между случайными величинами, называемую мерой интереса, которая выражается следующей формулой:

$$I = \frac{p(\boldsymbol{AB})}{p(\boldsymbol{A})\,p(\boldsymbol{B})}. \qquad (2.3)$$

В формуле (2.3) величина $p(\boldsymbol{AB})$ есть вероятность совместного появления паттернов $\boldsymbol{A}$ и $\boldsymbol{B}$, а величины $p(\boldsymbol{A})$ и $p(\boldsymbol{B})$ – это вероятности появления паттернов $\boldsymbol{A}$ и $\boldsymbol{B}$ в этой же выборке. Близость этой величины к единице свидетельствует о слабой статистической зависимости между паттернами $\boldsymbol{A}$ и $\boldsymbol{B}$. Эта величина в модели Г. Пятецкого–Шапиро используется для задания пороговой характеристики

$$\left| \frac{p(\boldsymbol{A}\,\boldsymbol{B})}{p(\boldsymbol{A})\,p(\boldsymbol{B})} - 1 \right| \geq \delta_{\min}, \qquad (2.4)$$

при этом величина δ_{min} названа автором *минимальным интересом*.

Параметрами алгоритмов для поиска ассоциативных правил в этом случае являются значения минимальной поддержки σ_{min}, минимальной уверенности γ_{min} и минимального интереса δ_{min}.

Более строго эта же модель оценки зависимости между компонентами паттерна введена в работе [43]. В ней для проверки зависимости паттернов $\boldsymbol{A}$ и $\boldsymbol{B}$ используется классический χ^2–тест математической статистики, проверяющий значимость гипотезы о равенстве случайных величин (точечных оценок вероятностей), присутствующих в числителе и знаменателе метрики Г. Пятецкого–Шапиро (2.3):

$$H_0 : p(\boldsymbol{AB}) - p(\boldsymbol{A})\,\boldsymbol{P}(\boldsymbol{B}) = 0. \qquad (2.5)$$

Алгоритм проверки этой гипотезы состоит в том, чтобы сосчитать оценки вероятностей отдельных паттернов по выборке, сосчитать оценку вероятности их совместного появления в выборке и оценить по критерию χ^2 значимость различия между этими величинами для заданного объема выборки и заданного порога отсечки. Значение порога отсечки, как обычно, задает уровень значимости этого различия.

Обратим внимание на следующее свойство описанной здесь модели ассоциативного правила. Ассоциативная связь, отвечающая модели *поддержка–уверенность–зависимость*, является симметричной относительно посылки и заключения. Другими словами, она не дает информации о направлении этой связи, утверждая только, что или $\boldsymbol{A} \rightarrow \boldsymbol{B}$, или $\boldsymbol{A} \leftarrow \boldsymbol{B}$, или эта связь двухсторонняя, т.е. $\boldsymbol{A} \leftrightarrow \boldsymbol{B}$. Естественно, что отсутствие информации о направлении ассоциативной связи в рассматриваемой здесь модели ассоциативного правила является ее большим недостатком. Этот недостаток преодолевается в моделях ассоциаций причинного типа, в которых рассматривается направленная статистическая связь вида $\boldsymbol{A} \rightarrow \boldsymbol{B}$. Этой модели посвящен раздел 2.3.

В последующем разделе описываются основные результаты, модели, методы, и алгоритмы, разработанные в области ассоциативной классификации, а также приводится их сравнительный анализ применительно к работе с данными большого объема.

2.2 Основные результаты в области ассоциативной классификации

Авторы [46], по всей видимости, первыми сформулировали задачу ассоциативной классификации. В постановке задачи рассматриваются обучающие данные, заданные в форме таблицы *объект–признак* с n примерами (строками), l атрибутами и q классами. При этом атрибуты данных являются категориальными, целочисленными или вещественными. В ходе предобработки значения категориальных атрибутов заменяются целочисленными значениями, однако, их порядок при этом во внимание не принимается, непрерывные атрибуты заменяются набором дискретных значений и также заменяются нумерованными атрибутами. Та-

ким образом, каждый пример выборки трансформируются во множество пар <*атрибут, целочисленное значение*>, которому ставится в соответствие метка класса. В работе рассматривается модель ассоциативного правила в стандартной форме *поддержка–уверенность* вида $A_i \to B_k$, где посылка $\boldsymbol{A_i} \in \boldsymbol{X}$ есть последовательность пар <*атрибут, целочисленное значение*>, а B_k есть метка класса, $k \in \{1,...,q\}$. Заметим, что авторы ошибочно называют такое правило импликацией.

Алгоритм поиска ассоциативных правил, предложенный в работе [46], назван CBA (англ. *Classification Based on Associations*). На первом шаге этого алгоритма выполняется описанная выше предобработка данных, в ходе которой выполняется их приведение к квази–целочисленной форме (на самом деле числа в ней играют просто роль символов). На втором шаге с помощью стандартного алгоритма Apriori [38] для каждого класса генерируется множество ассоциативных правил, удовлетворяющих заданным ограничениям на минимальные значения мер поддержки и уверенности. На третьем шаге осуществляется поиск правил ассоциативной классификации для каждого класса с помощью эвристической процедуры с использованием модифицированной идеи бустинга [47].

Опишем общую идею третьего шага предложенного алгоритма. Сначала на множестве всех правил некоторого класса B_k определяется линейный порядок следующим образом:

Правило $r_i \succ r_j$ (первое правило «предшествует» второму), если

1. Мера уверенности *conf* правила r_i больше, чем правила r_j, или

2. Меры уверенности обоих правил одинаковы, но правило r_i имеет большее значение меры поддержки, или

3. Обе меры имеют одинаковые значения для обоих правил, но первое правило просто сгенерировано раньше второго.

Затем, из множества упорядоченных правил класса B_k выбирается правило с наименьшим порядковым номером, и для этого правила находятся все примеры, которые этим правилом покрываются, то есть те примеры, для которых посылка

правила и его заключение одновременно истинны. Эти примеры далее не участвуют в дальнейшем процессе выбора правил для класса B_k. Если очередное правило покрывает хотя бы один новый пример, то оно рассматривается как потенциальное правило классификации и добавляется в итоговое множество. После этого для итогового множества правил классификации вычисляется коэффициент ошибок классификации, получаемый при использовании этого множества. Если эта величина превышает некоторый заданный порог, то описанные выше шаги повторяются для следующего по порядку правила. Процесс формирования итогового множества правил классификации продолжается до тех пор, пока на текущем шаге остаются примеры, которые еще не покрыты ни одним из выбранных правил. Заметим, что описанный алгоритм имеет много общего с процедурой обучения на основе бустинга [47]. После останова процедуры отбора правил классификации из итогового множества удаляются те правила, которые не улучшают точность классификации.

Описанный алгоритм выбора правил удовлетворяет следующим двум условиям:

1. Каждый пример в данных, используемых для обучения, будет покрыт хотя бы одним правилом, и это правило имеет наименьший номер в построенной ранее последовательности правил.

2. Каждое правило в выбранном множестве покрывает хотя бы один пример данных, который не покрывается другим правилом.

Такой алгоритм не является эффективным, поэтому авторы предлагают его эвристическую модификацию, в которой наилучшее правило отыскивается поочередно для каждого примера. Такой эвристический вариант алгоритма является многопроходным по множеству данных. Алгоритм состоит из трех этапов.

На первом этапе для каждого примера $d \in \boldsymbol{D}$ класса B_k находятся два правила, одно из которых правильно классифицирует этот пример и имеет наименьший номер в последовательности правил для этого класса (оно обозначается *cRule*). Второе, аналогичное первому, имеет наименьший номер, но классифицирует этот пример неверно (*wRule*). Если $cRule \succ wRule$, то *cRule* включается во множество

потенциальных правил классификации для класса B_k. Если же верно обратное, то на текущем этапе пока нельзя сказать, как правило *cRule* ведет себя по отношению к примерам других классов, и следует ли его включать в потенциальное множество правил для класса B_k. Для каждого *cRule* определяется то множество примеров, которое этим правилом покрывается.

На втором этапе для тех примеров, для которых выбор правила на этапе 1 сделан не был, выполняется повторный проход по данным. На этом проходе отыскиваются все те *wRule*–правила, которые предшествуют *cRule*-правилу, построенному для соответствующего примера. Далее каждое *wRule*–правило, найденное для примера, анализируется, и если оно помечено как *cRule*–правило для некоторого другого примера данного класса, то оно оставляется в найденном множестве, в противном случае – удаляется.

На третьем этапе выполняется финальный выбор правил для класса B_k, который реализуется в два шага. На первом шаге найденное множество правил упорядочивается и вычисляется ошибка классификации по мере увеличения числа правил в соответствии с установленным их порядком. Если при этом какие-то правила не покрывают новых примеров и не увеличивают точность классификации, то они удаляются из итогового множества. На втором шаге удаляются правила, которые вносят наибольшие ошибки. В работе [46] приведен подробный псевдокод этого алгоритма.

Авторы [46] сравнивают свой алгоритм CBA с алгоритмом C4.5 на основе экспериментов с 26 наборами данных из UCI ML Repository [48] и делают вывод о том, что алгоритм CBA демонстрирует более высокую точность. Этот вывод неубедителен с точки зрения оценки самого метода, поскольку алгоритм C4.5 не является наилучшим для всех случаев.

Ценность этой работы, однако, состоит в том, что в ней впервые сформулирована проблема ассоциативной классификации, описаны ее особенности и предложен эвристический алгоритм классификации CBA. Данный алгоритм аналогичен классическому алгоритму поиска ассоциативных правил с добавлением идей

бустинга, которые были предложены ранее для машинного обучения. Более поздние работы показывают, что задача поиска ассоциативных правил для решения задач классификации значительно своеобразнее и намного сложнее, чем это может показаться на основании работы [46], которая в определенном смысле является слишком «прямолинейным» переносом алгоритмов поиска ассоциативных правил на поиск классификационных ассоциативных правил.

Развитием алгоритма CBA является алгоритм CMAR (англ. ***C**lassification based on **M**ultiple **A**ssociation **R**ules*), предложенный в работе [49]. В этой работе задача классификации формулируется аналогично тому, как это сделано в [46]. Следует отметить, что обеспечение вычислительной эффективности ассоциативной классификации – это ключевая проблема, которая плохо решается большинством предложенных методов. Эта проблема особенно явно проявляется при работе с большими данными.

Для повышения эффективности поиска ассоциативных правил классификации авторы [49] вносят в алгоритм CBA некоторые важные изменения. Во-первых, авторы [49] отказываются от использования переборных алгоритмов типа *Apriori* для поиска ассоциативных правил в пользу разработанного ими ранее метода, хорошо известного в литературе под названием *метод возрастающих паттернов* (англ. *Frequent Pattern growth, FP-growth*) [44]. Основная же причина эффективности алгоритма CMAR, состоит в том, что все последовательности–кандидаты на включение в искомое множество часто встречающихся паттернов, формирующих посылки правил, представляются в виде *префиксного дерева последовательностей* (*FP-tree*), в котором каждый узел соответствует некоторому символу множества $\boldsymbol{X}$, а последовательности символов представляются последовательностью узлов дерева с началом в его корне. Дерево типа *FP-tree* используется как для представления множества часто встречающихся паттернов, так и для реализации процедур отсечения «плохих» правил, а также для просмотра и поиска правил в процессе классификации новых примеров, когда требуется выполнять сравнение (англ. *matching*) тестируемого примера с большим числом ассоциативных правил.

Получаемая в итоге структура для представления ассоциативных правил называется авторами *CR-tree*.

Приведем более детальное описание алгоритма CMAR и дерева *CR*-tree. Алгоритм поиска часто встречающихся паттернов в CMAR имеет два основных отличия от алгоритма *FP-growth*. Во-первых, алгоритм *FP-growth* выполняется в два шага. Сначала строится дерево часто встречающихся последовательностей, которые имеют значение меры поддержки больше заданного порогового значения, а затем генерируются ассоциативные правила с требуемым значением меры уверенности. В отличие от этого, в алгоритме CMAR часто встречающиеся паттерны и правила генерируются за один шаг. Второе отличие состоит в том, что алгоритм CMAR для каждого правила запоминает распределение значений меры поддержки на множестве всех классов, для которых данное правило имеет ненулевое ее значение. Сгенерированные в ходе алгоритма ассоциативные правила, которые хранятся в структуре дерева *CR-tree,* имеют три атрибута – метка класса и значения мер поддержки и уверенности. Пример такого дерева представлен на рисунке 2.1. *CR-tree* является компактной структурой, которая хранит индексы для доступа к правилам, что делает просмотр правил эффективной процедурой.

Отметим, что не все правила такого дерева участвуют в классификации. Худшие с точки зрения эффективности правила удаляются с помощью процедуры отсечения, которая также реализована с помощью дерева *CR-tree*. На множестве правил задается глобальный порядок, который строится следующим образом.

Правило r_2 предшествует правилу r_1, иначе, $r_1 \succ r_2$, если и только если

(1) $conf(r_1) > conf(r_2)$; или

(2) если $conf(r_1) = conf(r_2)$ но $supp(r_1) > supp(r_2)$; или

(3) если $conf(r_1) = conf(r_2)$ и $supp(r_1) > supp(r_2)$, но правило r_1 в левой части имеет меньшее число символов, чем правило r_2.

Говорят также, что правило r_1: $P \rightarrow C$ является более общим по отношению к правилу r_2: $P' \rightarrow C'$ тогда и только тогда, когда посылка P первого правила является подмножеством посылки P' второго правила. Введенный порядок используется алгоритмом CMAR для отсечения правил из дерева *CR-tree*.

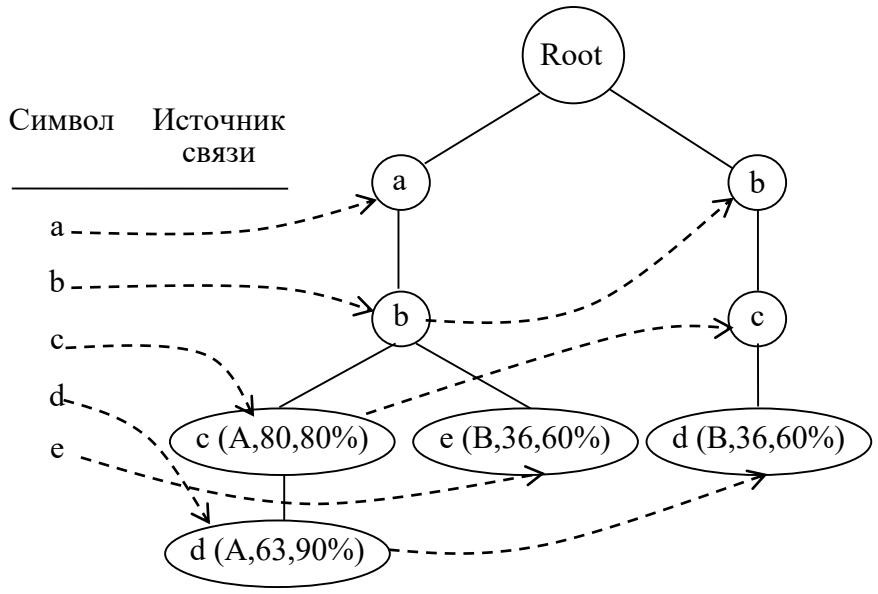


Рисунок 2.1 - Пример CR-tree. A, B–метки класса, первое число в скобках (80, 36 и т.д.) – значение меры поддержки, второе число в скобках (80%, 60% и т.д.) – значение меры уверенности (рисунок позаимствован [49])

Процедура отсечения выполняется в три шага:

Шаг 1. Отсекаются менее общие правила с низким значением меры уверенности. Это отсечение выполняется в процессе генерации правил.

Шаг 2. После построения дерева *CR-tree* из него удаляются правила, в которых посылка и заключение имеют отрицательную корреляцию, которая выявляется с применением χ^2–теста. Заметим, что с необходимостью удаления таких правил нельзя согласиться безоговорочно, поскольку правила с отрицательной корреляцией в некоторых случаях несут информацию, полезную для классификации, в особенности, если связь имеет причинный характер. Например, если некоторый паттерн имеет значение коэффициента корреляции с заключением близкое к величине –1, то такое правило является *запретом* для данного класса. А запрет представляет собой очень сильную закономерность. Его удаление из множества классификационных правил вряд ли оправдано без дополнительных обоснований.

Шаг 3. Отсекаются правила, имеющие покрытие примеров класса в обучающей выборке меньшее, чем заданный порог.

Алгоритм классификации аналогичен подобному в алгоритме CBA с некоторым отличием в модели объединения решений. В алгоритме CMAR все правила в

итоговом дереве *CR-tree* разбиваются на группы согласно метке класса в заключении правила. В ходе тестирования для каждого примера и для каждой такой группы правил вычисляется значение веса, который представляет собой значение эвристически выбранной функции – *взвешенной χ^2–статистики* [49], имеющей достаточно сложное выражение. Авторы признают, что она не имеет никакого теоретического обоснования или содержательной интерпретации. Мотивацией для ее использования в алгоритме CMAR являются только результаты экспериментальных исследований. Решение принимается в пользу того класса, для которого взвешенная χ^2–статистика принимает наибольшее значение.

Свойства алгоритма CMAR исследованы экспериментально на 26 наборах тестовых данных из UCI-репозитория [48]. На основании экспериментальных результатов, приведенных в работе, авторы заключают, что алгоритм CMAR обладает существенно лучшей вычислительной эффективностью, чем методы CBA и C4.5 [49], а также лучшими показателями точности решения задач классификации, однако в этом отношении его преимущества не столь существенны.

Для генерации правил ассоциативной классификации в работе [50] предлагается использовать идеи метода ID3 [51]. Несмотря на то, что предложенный в [50] метод CPAR (англ. ***C****lassification based on* ***P****redictive* ***A****ssociation* ***R****ules*), эксплуатирует довольно старую идею, он имеет некоторые новые свойства, заслуживающие упоминания.

В основу метода положен алгоритм FOIL [52]. Этот алгоритм рекурсивно отыскивает атрибут, добавляемый к потенциальной посылке формируемого правила, который максимизирует метрику, называемую *информационным выигрышем* (*information gain*). Максимизация этой метрики ведет к наилучшему покрытию текущего множества обучающих данных найденными правилами. Данные выборки, покрытые найденным правилом, удаляются из обучающего множества, и далее поиск атрибутов, обеспечивающих максимизацию информационного выигрыша, ведется по отношению к новому, сокращенному множеству обучающих данных. Этот метод работает с парой классов. Если классов больше двух, то метод *FOIL* использует схему «выбранный класс» – «все другие классы» для сведения

задачи с множеством классов к последовательности задач бинарной классификации. Алгоритм *CWAC* [53], по сути, аналогичен алгоритму *FOIL*. Отличие состоит только в том, что авторы [53], кроме *информационного выигрыша,* используют для оценки правил *взвешенную поддержку* и *взвешенную уверенность*.

Модификация метода CPAR применительно к задаче ассоциативной классификации состоит в следующем. В отличие от FOIL, который на каждом шаге рекурсивного формирования посылки правила допускает только одно его продолжение за счет добавления нового атрибута, алгоритм CPAR рассматривает несколько таких продолжений. В качестве вариантов продолжений он рассматривает все те атрибуты, которые имеют одинаковые или близкие значения *информационного выигрыша*. Эту часть алгоритма CPAR авторы называют PRM–алгоритмом (*от англ*. Predictive Rule Mining). Далее пример обучающих данных, покрытый вновь сгенерированным правилом, не удаляется из процесса обучения. Он используется и на последующих шагах поиска, но с меньшим весом. Каждое сгенерированное правило оценивается по некоторой метрике, значение которой используется в дальнейшем для принятия решения о том, использовать ли то или иное правило в алгоритме классификации или нет. Для каждого класса оставляется k наилучших правил по выбранной метрике, на базе которых и строится алгоритм классификации новых примеров. Заметим, что механизм классификации на основе множества построенных правил не отличается оригинальностью. В нем для классифицируемого примера оценивается среднее значение вероятности его принадлежности к каждому классу по множеству всех правил. Предпочтение отдается тому из классов, для которого эта вероятность наибольшая. Детали алгоритма, как и доказательства корректности различных его шагов применительно к задачам ассоциативной классификации, могут быть найдены в работе [50].

Метод CPAR был исследован экспериментально на 26 наборах данных из UCI–репозитория [48]. По утверждению авторов, он превзошел по точности предсказания класса другие методы, которые на момент публикации работы [50] рассматривались как наилучшие. К ним относятся, в частности, такие методы, как C4.5 [51], RIPPER [54], CBA [46], CMAR [49] и ACAC [55, 56].

Заметим, что в число алгоритмов, с которыми авторы сравнивали метод CPAR, не вошли алгоритмы, основанные на использовании понятия эмерджентных паттернов, которые рассматриваются далее, хотя эти работы были опубликованы за несколько лет до алгоритма CPAR.

Методы ассоциативной классификации, основанные на *эмерджентных паттернах,* описанные, например, в работах [57, 58, 59, 60] сформировали новое активно развиваемое направление в этой области.

Впервые понятие *эмерджентного паттерна* (англ. *Emergent Pattern,* EP) было, по-видимому, введено в работе [57]. При этом, задача ассоциативной классификации была поставлена как задача поиска правил, позволяющих отделить примеры одного класса от примеров другого класса, т.е. в качестве базового критерия отбора правил ассоциативной классификации рассматривалась их способность отличать примеры одного класса от примеров другого класса. Этот критерий остается неизменным в ходе дальнейшей эволюции эмерджентных моделей, а совершенствование методов и алгоритмов направлено на повышение вычислительной эффективности при генерации ассоциативных правил и их использовании в алгоритме классификации.

Сформулируем понятие эмерджентного паттерна (ЭП). Пусть дана упорядоченная пара $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ транзакционных данных, которые относятся к разным классам либо к разным временным интервалам лога работы некоторой системы. Каждая транзакция из множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ может содержать элементы (атрибуты, переменные, предметы, объекты, *англ.* items) из линейно упорядоченного множества или последовательности $\boldsymbol{X}$. Рассмотрим произвольный паттерн $A \subseteq \boldsymbol{X}$, который характеризуется поддержкой σ_1 во множестве $\boldsymbol{D_1}$ и поддержкой σ_2 во множестве $\boldsymbol{D_2}$. Отношение σ_1/σ_2 авторы работы [57] называют *коэффициентом возрастания поддержки* (*Growth rate*) паттерна *A* от множества данных $\boldsymbol{D_1}$ ко множеству $\boldsymbol{D_2}$. Формально значение показателя *GrowthRate(A)* определяется следующей формулой

$$GrowthRate(A) = \begin{cases} 0, \text{ если } \sigma_1(A) = 0 \text{ и } \sigma_2(A) = 0, \\ \infty, \text{ если } \sigma_1(A) = 0 \text{ и } \sigma_2(A) > 0, \\ \sigma_2(A) / \sigma_1(A), \text{ в других случаях.} \end{cases} \quad (2.6)$$

Паттерн *A* называется *эмерджентным паттерном* от множества $\boldsymbol{D_1}$ к множеству $\boldsymbol{D_2}$, если $GrowthRate(A) \geq \rho$. Таким образом, от выбора значения порога ρ зависит разделяющая способность ЭП. Отметим, что понятие меры уверенности для ЭП на заданных множествах данных $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ становится ненужным, т.к. ее значение для обоих множеств равно 1, поскольку всем транзакциям каждого из множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ ставится в соответствие постоянное заключение, например, метка класса или имя временного интервала.

Задача поиска ассоциативных правил в работе [57] сводится к поиску эмерджентных паттернов $\boldsymbol{A_i}$ со значением меры $GrowthRate(A_i) \geq \rho$. Так как задача поиска ЭП опирается на понятие коэффициента роста поддержки, «хорошие» паттерны могут иметь низкое значение поддержки в обоих множествах, что приводит к значительному возрастанию вычислительной сложности задачи поиска ЭП. Это обусловлено тем, что, во-первых, паттернов с низким уровнем поддержки существует очень много. Во-вторых, при поиске паттернов по условию $GrowthRate(A_i) \geq \rho$ не представляется возможным использовать алгоритм *Apriori,* поскольку для ЭП нельзя использовать свойство антимонотонности.

Работа [57] показала, что поиск ассоциативных правил для задач классификации является задачей, которая, во-первых, серьезно отличается от поиска обычных ассоциативных правил, и, во-вторых, имеет не так много общего с задачей поиска правил в машинном обучении. Эта работа подробно описывает алгоритм нахождения ЭП с заданными областями значений поддержки во множествах $\boldsymbol{D_2}$ и $\boldsymbol{D_1}$, используя двухмерное представление области локализации различных типов ЭП в координатах $<\sigma_2, \sigma_1>$, т.е. в координатах мер поддержки паттернов во множествах данных $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$. В этой области (рисунок 2.2, который повторяет рисунок, приведенный в работе [57]) все ЭП располагаются правее прямой l_1, тангенс угла $\boldsymbol{\alpha}$ наклона которой к оси $O\sigma_2$ равен величине $1/\rho$. Все ЭП располагаются в треугольнике *ACE*. Однако посылки правил ассоциативной классификации, должны иметь значение поддержки, превышающее минимально допустимое значение σ_{2min} во множестве $\boldsymbol{D_2}$. Кроме того, во множестве $\boldsymbol{D_1}$ они должны иметь значение поддержки меньше, чем $\boldsymbol{\sigma_{2min}}$. Паттерны с такими значениями мер

поддержки σ_2 и σ_1 располагаются в прямоугольнике *BCDG*, и именно они являются целью поиска в работе [57]. При этом авторы подчеркивают, что поиск ЭП, отвечающих треугольнику *ABG*, является вычислительно сложной задачей ввиду того, что в этой области паттерны, получаемые на основе данных множества $\boldsymbol{D_1}$, обладают низким значение поддержки, а потому их может быть катастрофически много. Наоборот, паттерны, которые получаются для множества данных $\boldsymbol{D_1}$, будут отвечать треугольнику *GDE*, и их, обычно, бывает немного, а потому их можно проверить простым перебором.

Однако, работа [57] не рассматривает, каким образом полученные ЭП могут быть использованы в алгоритме ассоциативной классификации. Этому вопросу посвящена работа [58].

Одной из особенностей задачи классификации с использованием ЭП является большое количество классифицирующих правил, каждое из которых может покрывать только небольшое число примеров обучающей выборки. Правила классификации, генерируемые большинством других методов с большим значением покрытия, можно рассматривать как самостоятельные классификаторы. Если каждый из них имеет вероятность правильной классификации больше, чем 0.5, то, в соответствии с теоремой Кондорсе результат голосования сходится с вероятностью 1 к правильному решению при увеличении числа правил [61, 62]. В отличие от этого, каждый ЭП работает правильно на очень небольшой доле обучающих данных, а вероятность правильной классификации с помощью ЭП на всем множестве данных может быть менее 0.01. Поэтому с помощью таких правил нельзя строить алгоритмы классификации на основе голосования, и было бы правильнее интерпретировать ЭП как некоторые более удобные новые признаки, каждый из которых все еще не может рассматриваться как «хороший» классификатор. Именно поэтому авторы работы [58] рассматривают задачу построения классификаторов на основе ЭП как самостоятельную задачу.

Приведем краткое описание технологии построения ассоциативного классификатора CAEP (англ. ***C**lassification by **A**ggregating **E**merging **P**atterns*), которая

позволяет преобразовать подмножества ЭП каждого класса в более выразительные структуры для увеличения их возможностей по дискриминации классов.

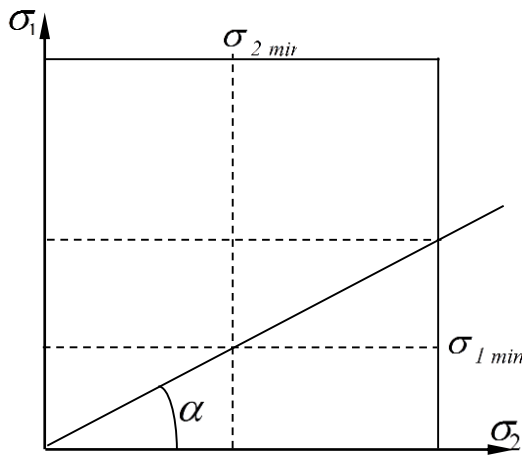


Рисунок 2.2 - Пояснение к алгоритму поиска эмерджентных паттернов

Сначала авторы рассматривают индивидуальные разделяющие возможности ЭП, которые зачастую очень ограничены, и для их оценки вводят следующую формулу

$$Score(E_i, C, s) = \frac{Growth\_Rate(E_i)}{Growth\_Rate(E_i)+1} \times support_C(E_i), \qquad (2.7)$$

где E_i – ЭП, разделительные возможности которого оцениваются функцией (2.7) для примера s по отношению к классу C, а величина $support_C(E_i)$ есть поддержка паттерна E_i в классе C.

Авторы обращают внимание на то, что в выражении $(Growth\_Rate(E_i)/(Growth\_Rate(E_i)+1)) \times support_C(E_i)$ первый сомножитель примерно равен условной вероятности события *«пример s, в котором имеется ЭП E_i, принадлежит классу C»*, а второй сомножитель – это доля примеров класса C, которые содержат данный ЭП. Сумму всех таких величин по множеству всех ЭП класса C авторы предлагают рассматривать в качестве величины, характеризующей разделительную силу построенного множества ЭП для этого класса:

$$Score(C, s) = \sum_{E_i \subseteq s, E_i \in E(C)} \frac{Growth\_Rate(E_i)}{Growth\_Rate(E_i)+1} \times support_C(E_i) \qquad (2.8)$$

Данная величина была бы равна полной вероятности класса *C* в том случае, если бы ЭП класса на выборке класса *C* были независимыми случайными величинами. А это может иметь место в том и только в том случае, когда каждый паттерн покрывает некоторое множество данных, которые не покрываются ни одним другим паттерном. Такой случай нереален, поэтому мотивация авторов в этой части является неубедительной. Авторы понимают, что оценка разделительной способности множества паттернов в виде (2.8) не является вполне корректной. В реальных ситуациях сумма в (2.8) будет напрямую зависеть от числа ЭП, сгенерированных для класса, а также от меры зависимости паттернов класса между собой. Поэтому строить выбор классификатора по максимуму этой величины нельзя. Для ослабления названных недостатков меры (2.8) при ее использовании в качестве атрибута классификации авторы предлагают нормировать эту величину по множеству всех классов. В качестве нормирующего коэффициента используется величина, которую авторы называют *base_score*(*C*), которая вычисляется как *медиана* множества значений величин *Score(C,s)* для всех обучающих данных класса *C*. Медиана отвечает значению величины *Score(C,s)* для того примера данных класса *C*, для которого 50% всех тренировочных данных этого класса имеют значения больше него. Это значение *Score(C,s)* для конкретного примера берется в качестве значения нормирующего коэффициента *base_score*(*C*) класса *C* при вычислении нормированного значения функции типа (2.8), используемой в алгоритме классификации:

$$Norm\_Score(C,s) = score(C,s) / base\_score(C) \qquad (2.9)$$

Данный алгоритм проверен авторами на большом числе наборов данных. По их утверждению он показал хорошие свойства как по эффективности поиска ассоциативных правил классификации, так и по качеству решенных задач классификации данных. В частности, экспериментальные результаты авторов показали, что алгоритм CAEP обладает лучшими свойствами по точности классификации по сравнению с классическим методом C4.5, а также по сравнению с методом CBA, который был рассмотрен ранее.

Авторы работ [57, 58], понимают, что достигнутого уровня вычислительной эффективности алгоритмов синтеза таких классификаторов недостаточно для работы с *большими данными*. Их исследования в течение последующего десятилетия позволили им существенно улучшить как эффективность, так и точность модели обучения ассоциативной классификации, в основе которой лежит понятие ЭП. Это достигнуто как за счет расширения понятия эмерджентного паттерна, так и за счет повышения эффективности алгоритмов их поиска.

В работе [60] авторы алгоритма CAEP вводят понятие *скачкообразного эмерджентного паттерна* (*англ*. **J**umping **E**mergent **P**attern, JEP). Скачкообразный эмерджентный паттерн (СЭП) – это ЭП, который в одном множестве данных имеет нулевое значение поддержки, а в другом – строго положительное. Следует отметить, что такие правила классификации, рассматривались ранее в работах [63, 64] и использовались в алгоритме обучения GK2 [65] и других алгоритмах, которые брали за основу модусы сходства и различия Д.С.Милля [66, 67]. Новизна использования понятия СЭП состоит именно в том, что он рассматривается в задаче ассоциативной классификации.

Для поиска СЭП и построения классификатора на множестве найденных СЭП (этот алгоритм назван авторами алгоритмом JEP), аналогично алгоритму CAEP, используется понятие правой и левой границ множества ЭП. Поиск СЭП сводится при этом к поиску паттернов, задающих левую границу множества всех ЭП.

Классификатор в алгоритме JEP работает следующим образом. Пусть s - это новая транзакция, для которой необходимо предсказать класс принадлежности. Для этого в [60] для каждого класса $C_p \in \{C_1, C_2, \ldots, C_q\}$ вычисляется значение меры, названной *коллективным влиянием* СЭП (*англ*. collective impact, CI), по следующей формуле:

$$CI(C_p) = \sum_{i:\ СЭП_i \in ME-JEP(C_p, \overline{C}_p)\ \&\ СЭП \subseteq s} supp_{D_p}(СЭП_i) \qquad (2.10)$$

Транзакция s считается принадлежащей к классу C_p, для которого значение коллективного влияния $CI(C_p)$ максимально. Очевидно, что такая модель принятия решения аналогична схеме взвешенного голосования.

Выделим основные различия алгоритмов JEP и CAEP. Во-первых, в них поразному строятся множества ЭП, а также есть различия и в построении алгоритма классификации. В алгоритме JEP не используется понятие *GrowthRate*. Оба алгоритма имеют примерно одинаковую точность принятия решений. Оба они, по заключению авторов, превосходят по точности и по вычислительной эффективности алгоритм CBA и классический алгоритм C4.5.

Алгоритмы CAEP и JEP имеют как достоинства, так и недостатки. Например, алгоритм CAEP использует паттерны с некоторым пороговым значением поддержки (например, 1%), а СЭП с такой поддержкой могут не существовать. Хотя алгоритм JEP создан как некое развитие алгоритма CAEP, он не дает принципиального улучшения как вычислительной эффективности, так и точности ассоциативной классификации по сравнению с аналогичными характеристиками алгоритма CAEP.

В работе [68] используется модификация СЭП, а именно СЭП с подсчетом встречаемости (англ. *Jumping Emerging Patterns with Occurrence Counts*). При этом для нахождения СЭП в работе также используется алгоритм с построением границ. Авторы экспериментально показывают, что такое расширение СЭП хорошо работает в области классификации изображений.

В своих более поздних работах (например, в [59] и др.) авторы алгоритмов CAEP и JEP признают, что несмотря на то, что число наиболее выразительных СЭП, полученных алгоритмом JEP, значительно меньше, чем общее их количество, их оказывается слишком много даже для данных небольшого объема и размерности. Например, в одной из задач для данных, содержащих 1000 примеров, описанных 20 атрибутами, общее количество СЭП оказалось равным 32244. Из них 2754 СЭП оказались наиболее выразительными. Однако, это число также слишком велико для эффективной и устойчивой классификации. Для дальней-

шего снижения числа используемых СЭП требуется более длительная фаза обучения на этапе построения классификатора. Для преодоления этих недостатков авторы предлагают искать ЭП, которые обладают большей дискриминационной силой, но присутствуют в данных в меньшем числе.

Авторы [69] также отказываются от использования СЭП в пользу обычных ЭП и экспериментально показывают, что ЭП дают более точную классификацию.

Первый шаг, позволяющий найти ЭП с большей дискриминационной силой, – это введение ограничений на минимальное значение поддержки СЭП, что обеспечивает некоторый минимальный уровень покрытия им тренировочных данных. Такой паттерн авторы [59] называют строгим СЭП и формально определяют его следующим образом. Паттерн $A \in \boldsymbol{X}$ называется с*трогим скачкообразным ЭП* (англ. *Strong Jumping Emergent Pattern,* SJEP) из множества $\boldsymbol{D_1}$ во множество $\boldsymbol{D_2}$ (ССЭП), если для него выполнены следующие два условия:

$$supp_{D_1}(A) = 0 \text{ и } supp_{D_2}(A) \geq \delta, \qquad (2.11)$$

и любое собственное подмножество элементов паттерна *A* не удовлетворяет первому условию. Этот шаг позволяет повысить выразительность паттернов, на базе которых строится ассоциативный классификатор, и уменьшить их количество.

Вторым шагом в направлении повышения вычислительной эффективности является введение специальной структуры представления множества ЭП, которая обеспечивает их эффективное хранение, просмотр и поиск. Эта структура названа *деревом контрастных паттернов* (*англ*. Contrast Pattern Tree Structure, CPT structure) [59]. Отметим, что такое дерево аналогично дереву для *представления возрастающих паттернов* (*англ*. FP-growth Tree), которое впервые было представлено в работе [44]. Авторы [59] при этом не ссылаются на работы, в которых метод *FP–growth* и структура дерева *FP–Tree* были ранее предложены. В структуре *FP-growth Tree* множество паттернов задается в префиксной форме. В ней все паттерны, имеющие общий префикс, представляются общим путем из корня дерева до узла, которому соответствует последний общий символ паттернов (последний символ общего префикса). В работе [59] паттерны генерируются и структуриру-

ются точно так же, однако содержание каждого узла структуры CPT намного богаче. Еще одно новшество структуры CPT состоит в том, что представляемые в нем данные предварительно упорядочиваются по возрастанию разделяющих свойств элементов. Дерево контрастных паттернов CPT строится так, чтобы все ветви, исходящие из корня, представляли паттерны, упорядоченные *слева направо* (чем левее паттерн, тем он важнее) и *от корня дерева вниз* – в каждой ветви (более важные одноэлементные паттерны находятся в ветви дерева ближе к корню). Каждый узел дерева представляет собой упорядоченное слева направо подмножество одноэлементных паттернов, каждому из которых поставлено в соответствие значение функции поддержки во множествах $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$. Структура CPT для представления данных используется далее алгоритмом генерации ССЭП. Описание этого алгоритма в псевдокоде дано в работе [59].

Достоинство алгоритма, представленного в работе [59], состоит в его лучшей эффективности. Он выполняет поиск ССЭП одновременно для двух множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ за один проход всей базы данных. В этом его важное преимущество по сравнению с алгоритмами поиска ЭП и СЭП, в которых этот поиск делается поочередно для каждого множества с использованием представления множества эмерджентных паттернов с помощью верхней и нижней границ.

Дальнейшие усилия авторов данного направления были направлены на поиск других типов эмерджентных паттернов, которые могут быть эффективно использованы для построения ассоциативных классификаторов. Один из предложенных вариантов, обобщая понятие ССЭП, допускает, что поддержка таких паттернов не обязательно должна быть равна нулю в одном из классов. Такой подход объясняется возможностью зашумления ССЭП, которое приведет к незначительному отклонению значения поддержки от нуля в одном из классов. Это предположение подтверждается экспериментами: на тестовых данных ССЭП почти всегда имеет место ненулевое значение поддержки в соответствующем классе. Заметим, что это подтверждается и при тестировании правил, полученных алгоритмами индуктивного обучения, например, алгоритмами AQ [63, 64] или GK2 [65]. Такие паттерны авторы называют *эмерджентными паттернами, устойчивыми к шуму* (англ.

Noise-tolerant EPs, NEPs). В формальном определении NEP допускается, что порог его поддержки в одном из множеств не превышает заданной малой величины, а порог поддержки в другом, наоборот, не меньше, чем заданное пороговое значение. Авторы вводят также понятие *обобщенного эмерджентного паттерна, устойчивого к шуму* (англ. *Generalized Noise–tolerant EP*, GNEP). Здесь вместо традиционной меры, задаваемой функцией *GrowthRate*, допускается использование других функций от значений продержки в двух множествах. Однако конструктивность и полезность такого понятия авторами не обосновывается.

В отношении использования полученных ССЭП в алгоритме ассоциативной классификации авторы не предлагают чего-либо нового и рассматривают варианты, аналогичные тем, что были предложены ими и другими исследователями в данной области.

Эффективность и качество алгоритмов генерации ССЭП, а также их использование в задачах классификации были тщательно исследованы авторами на большом количестве наборов данных. Полученные экспериментальные результаты сравнивались с результатами работы алгоритмов CBA, C4.5 и его версиями, улучшенными за счет бустинга, и другими алгоритмами. Во всех случаях алгоритм SJEP оказывался существенно лучше по эффективности и показывал, в среднем, лучшие результаты по точности классификации. Эти оценки были получены на десятках различных наборов данных из UCI–репозитория [48]. Алгоритм SJEP сравнивался также с алгоритмом JEP и показал в среднем десятикратное ускорение решения задач и сравнимую точность при меньшем числе используемых паттернов. Несколько более осторожно авторы делают заключение о перспективности использования паттернов, которые являются обобщением ССЭП, а именно эмерджентных паттернов, устойчивых к шуму и их обобщений.

Модели ассоциативной классификации на основе различных видов эмерджентных паттернов являются важным шагом в области построения эффективных моделей классификации при работе с большими данными. Однако, по всей видимости, на сегодня они исчерпали все свои возможности по дальнейшему повышению эффективности. Одной из причин для такого заключения является необходи-

мость сведения любых данных в модели ЭП к булевым. Такое преобразование, выполняемое традиционными методами, всегда приводит к заметному увеличению размерности и, следовательно, ставит дополнительные ограничения на возможности такого подхода при работе с большими данными.

Появление модели эмерджентного паттерна, заимствованного из классической теории индуктивного обучения [63], стало существенным шагом вперед в области поиска ассоциативных правил классификации. В этой модели сама задача поиска правил была сформулирована с учетом прагматики задачи ассоциативной классификации. Её суть заключается в построении правила, которые могли бы отделить экземпляры данных одного класса от экземпляров данных других классов. Это позволило авторам в дальнейшем сосредоточиться на эффективности алгоритмов поиска ЭП. Другое большое достижение в этой части – это введение структуры CPT–дерева для удобного представления данных обучения и построения эффективных алгоритмов их использования в процессах генерации ССЭП. Несмотря на то, что первоначальная идея использования такого дерева не принадлежит авторам работ в области алгоритмов генерации ССЭП, структура CPT–дерева является весьма продуктивной идеей в области обучения классификации. По всей видимости, такое дерево может быть использовано в алгоритмах поиска минимальных правил в задачах индуктивного обучения при решении задач типа оптимального покрытия, а также найти другие применения.

2.3 Причинные структуры и ассоциативные связи

Несмотря на определенные ограничения при работе с гетерогенными данными сложной структуры, методы ассоциативного анализа обычно оказываются удобными для того, чтобы справиться с особенностями, характерными для больших данных [28, 36, 38, 39, 44]. Однако в работах [28, 39] теоретически и экспериментально показано, что среди ассоциативных связей наиболее полезными с точки зрения информативности в задачах принятия решений являются связи причинного характера. Именно ассоциативно-причинный анализ позволяет наиболее эффективно решать классические задачи анализа данных, в частности, задачи по-

иска и отбора признаков. Отметим, что ассоциация, по определению, является ненаправленной связью, и следовательно ее нельзя напрямую интерпретировать как причинно–следственную связь, если не обосновывать это специальными методами или не использовать метрики, которые должны быть специально для этого разработаны.

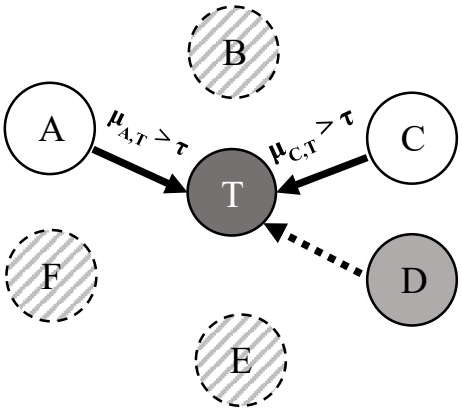
Активная работа над формальной теорией причинных связей и их извлечением ведется с 1980х годов многими исследователями [40, 70, 71, 72].

Наиболее распространённой моделью представления причинных связей в данных на сегодня является *байесовская сеть доверия*, предложенная Д. Пирлом в 1988 году [40]. Байесовская сеть – это математическая структура, которая компактно представляет объединенное распределение вероятностей $\boldsymbol{P}$ вектора случайных переменных $\boldsymbol{V}$ с помощью направленного ацикличного графа **G** с таблицами условных распределений вероятностей переменной, поставленной в соответствие каждому узлу графа для всех экземпляров его родителей. Таким образом, байесовская сеть (БС) представляет собой триплет $<\boldsymbol{V}, \boldsymbol{G}, \boldsymbol{P}>$ [40, 73]. Причинная байесовская сеть $<\boldsymbol{V}, \boldsymbol{G}, \boldsymbol{P}>$ является байесовской сетью с дополнительной семантикой: если $X \in \boldsymbol{V}$ и $Y \in \boldsymbol{V}$, то узел X является родителем узла Y в $\boldsymbol{G}$, если случайная величина X является непосредственной причиной случайной величины Y. Причинная байесовская сеть $\boldsymbol{G}$ и объединенное распределение $\boldsymbol{P}$ являются корректными относительно друг друга тогда и только тогда, когда каждая условная независимость, представленная в графе $\boldsymbol{G}$, имеет также место и в распределении в $\boldsymbol{P}$ [40].

Однако построение Байесовской сети доверия требует решения достаточно сложных задач обучения, которые имеют экспоненциальную сложность относительно множества атрибутов [40, 28]. Так как в больших данных число атрибутов может измеряться десятками и сотнями тысяч, использование модели Байесовской сети доверия в задачах причинного анализа больших данных бесперспективно. Поэтому актуальной задачей является поиск методов, которые позволили бы упростить поиск причин, например, путем существенного снижения числа атрибутов, которые являются кандидатами в искомое множество причин той или

иной целевой переменной или использованием более простых критериев оценки причинности связей.

В настоящее время наиболее перспективной представляется схема поиска множества причинных связей, в которой на первом шаге для целевой переменной находится множество атрибутов, связанных с ней значимой ассоциативной связью, а потом оценивается по некоторой мере, является ли эта связь причинной. При этом на первом шаге, как обычно, используется некоторая модель поиска ассоциативной связи (см. раздел 2.1). С ее помощью находится множество атрибутов, часть из которых потенциально может быть связана с целевой переменной причинной связью (рисунок 2.3). На втором шаге для каждого такого атрибута проверяется гипотеза о том, является ли найденная для него ассоциативная связь с целевой переменной причинной связью. Такая схема используется, например, в работе [70]. Объем вычислений при этом зависит, в частности, от того, насколько удачно выбрана мера ассоциативной связи, поскольку разные меры могут генерировать разные множества атрибутов и обладать разной вычислительной эффективностью.



A	B	C	D	E	F	T
0	0	1	0	1	0	1
1	1	0	0	1	1	0
1	1	1	0	1	1	1
1	1	1	1	1	1	1
0	1	0	0	1	1	0
0	1	1	0	1	1	1
1	1	0	1	1	1	1
…	…	…	…	…	…	…
0	0	0	1	0	0	0

Рисунок 2.3 – Упрощенная схема поиска причин, где T – целевая переменная; A, B, C, D, E, F – некоторые переменные, соответствующие атрибутам данных, из которых D связана с T ассоциативной связью, а A и C – причинной, μ – некоторая мера причинной связи, τ – пороговое значение меры μ

Отметим, что в качестве причин могут выступать не только отдельные атрибуты данных, но и паттерны данных. Для паттернов, как и для одиночных атрибу-

тов данных, численная мера их ассоциации с целевой переменной может характеризовать численную оценку перспективности паттерна при решении той или иной задачи.

2.4 Исследование численных мер оценки ассоциативно-причинных связей в данных

2.4.1 Роль численных мер связи в ассоциативно-причинном анализе больших данных и необходимые свойства мер причинной связи

Для практической реализации схемы ускоренного поиска множества причинных связей необходимо, в первую очередь, обоснованно выбрать подходящую численную меру для оценки «силы» причинной связи между переменными. Так как причинная связь переменных является частным случаем ассоциативной связи, то естественно попытаться найти хорошую численную меры оценки причинной связи среди мер, предложенных для оценки «силы» ассоциаций. Некоторые примеры таких мер были рассмотрены ранее в обзоре, приведенном в главе 2. Примерами их являются поддержка, уверенность, мера интереса. Однако имеет смысл не ограничивать поиск только такими мерами.

При анализе мер ассоциативной связи на предмет пригодности их для оценки силы причинной связи нужно, прежде всего, выделить те меры, которые формально можно квалифицировать как меры оценки причинной связи, однако выбор среди них наилучшей можно сделать только на основе их экспериментальной оценки.

Рассмотрим содержательно те свойства численных мер оценки причинной связи, на основании которых можно было бы их сравнивать и определять среди них более предпочтительные. В качестве таких свойств в работе рассматриваются следующие свойства:

(1) какова вычислительная сложность вычисления меры (насколько быстро она может быть вычислена для всего множества пар атрибутов больших данных);

(2) отражает ли эта мера причинный характер найденной связи между парой переменных, которые являются аргументами этой меры;

(3) насколько полон список причин целевой переменной, которые та или иная мера позволяет находить.

Применительно к каждой мере ассоциативной связи, предложенной к настоящему времени, вопрос относительно способности этой меры выявлять причинные связи между атрибутами в данной работе решается в два этапа. Сначала анализируются формальные свойства меры для того, чтобы определить, может ли та или иная мера отобрать атрибуты, которые потенциально могут быть причинами целевой переменной. На втором этапе проводятся экспериментальные исследование для выбора наилучшей меры из числа тех, которые отобраны на первом этапе.

2.4.2 Обзор численных мер ассоциативной связи и их формальный анализ с точки зрения пригодности для оценки причинных связей

Для оценки свойств ассоциативной связи между переменными/паттернами могут быть использованы объективные и/или субъективные меры связи. Субъективные меры принимают во внимание априорные знания пользователя о связях аргументов с целевой переменной и о семантике аргументов. Субъективные меры [75], такие как нетривиальность, способность выявлять правила принятия решений (англ. *actionability*) и новизна, или способность отражать неочевидное знание) обычно невозможно определить формально, поскольку они отражают знание, интерпретация которого дается экспертом и которое может быть проверено лишь экспериментально. В данной работе субъективные меры не рассматриваются. Однако свойства, используемые экспертами для формирования субъективных мер, могут привлекаться для анализа семантических свойств причинных связей, полученных на основе тех или иных численных мер.

Объективные меры ориентированы на данные и принимают во внимание только их. Объективная мера есть некоторая функция $\mu = F(A,B)$, вычисляемая по выборке данных, содержащих атрибуты/паттерны A и B, которые интерпретируются как случайные события, а потому функция $F(A,B)$ является некоторой выборочной статистикой. В качестве аргументов функции $F(A,B)$ обычно рассматриваются выборочные оценки вероятностей событий A и B, так что формальное представление объективной меры связи атрибутов A и B имеет вид некоторой функции

$\mu\left(n, p_A, p_B, p_{\bar{A}}, p_{\bar{B}}, p_{AB}, p_{\bar{A}B}, p_{A\bar{B}}, p_{\bar{A}\bar{B}}\right) \in R$. В этой формуле n – объем выборки, а символом p обозначены выборочные вероятности событий, указанных в его нижнем индексе. Важно также отметить, что значение меры связи может зависеть от порядка следования аргументов в ней, т.е. в общем случае $F(A,B) \neq F(B,A)$. Объективные меры в общем случае могут принимать положительные и отрицательные значения, и они обычно выбираются так, чтобы большее значение ее абсолютной величины соответствовало *«более сильному»* правилу.

Для проверки ассоциативной связи на принадлежность к классу причинных связей предполагается использовать те меры, которые удовлетворяют некоторым формальным требованиям к мерам оценки причинности, которые, по существу, можно назвать *аксиомами* меры причинной связи. Эти требования перечислены ниже.

В силу того, что причинная связь является направленной [76], мера, которая сможет оценивать силу причинной связи, должна быть некоммутативной (критерий 1), указывая направление связи. Обычно требуется, чтобы мера принимала значения в интервале [-1, 1] либо в интервале [0, 1] (критерий 2). И наконец, нулевое значение меры должно указывать на отсутствие причинной связи между ее аргументами (критерий 3).

Список объективных мер связи пары атрибутов, предложенных к настоящему времени в статистике, социологии, машинном обучении и литературе по интеллектуальному анализу данных, представлен в таблице 2.1.

Анализ мер, приведенных в таблице 2.1, показывает, что некоммутативными являются следующие меры: *уверенность*, *мера Лапласа*, *J-мера*, *убеждение*, *добавочное значение*, *индекс Джини*, *мера Клозгена*, *мера Сибега и Шонауера*, *фактор определенности* и *коэффициент регрессии*. Остальные меры являются коммутативными, т.е. для них $F(A \rightarrow B) = F(B \rightarrow A)$.

Следующие меры имеют нулевое значение при отсутствии связи, но их диапазон значений отличен от [–1, 1] ([0, 1]): *убеждение*, *добавочное значение*, *мера Клозгена*, *мера Сибега и Шонауера*.

Таблица 2.1 - Численные меры ассоциативной связи

№	Мера	Определение	Область значений	Источник
1	2	3	4	5
1	ϕ-коэффициент (ϕ)	$\frac{p_{AB} - p_A \cdot p_B}{\sqrt{p_A \cdot p_B \cdot (1-p_A) \cdot (1-p_B)}}$	[-1; 1]	[77]
2	Соотношение шансов (α)	$\left(p_{AB} \cdot p_{\bar{A}\bar{B}}\right) / \left(p_{A\bar{B}} \cdot p_{\bar{A}B}\right)$	[0; ∞]	[78]
3	Q-коэффициент ассоциации Юла (Q)	$\frac{p_{AB} \cdot p_{\bar{A}\bar{B}} - p_{A\bar{B}} \cdot p_{\bar{A}B}}{p_{AB} \cdot p_{\bar{A}\bar{B}} + p_{A\bar{B}} \cdot p_{\bar{A}B}}$	[-1; 1]	[79]
4	Y-коэффициент ассоциации Юла (Y)	$\frac{\sqrt{p_{AB} \cdot p_{\bar{A}\bar{B}}} - \sqrt{p_{A\bar{B}} \cdot p_{\bar{A}B}}}{\sqrt{p_{AB} \cdot p_{\bar{A}\bar{B}}} + \sqrt{p_{A\bar{B}} \cdot p_{\bar{A}B}}}$	[-1; 1]	[79]
5	к-коэффициент (к)	$\frac{p_{AB} + p_{\bar{A}\bar{B}} - p_A \cdot p_B - p_{\bar{A}} \cdot p_{\bar{B}}}{1 - p_A \cdot p_B - p_{\bar{A}} \cdot p_{\bar{B}}}$	[-1; 1]	[80]
6	J-мера (J)	$p_{AB} \cdot \log\left(p_{B\|A} / p_B\right) + p_{A\bar{B}} \cdot \log\left(p_{\bar{B}\|A} / p_{\bar{B}}\right)$	[0; 1]	[80]
7	Индекс Джинни (G)	$p_A \cdot \left(p^2_{B\|A} + p^2_{\bar{B}\|A}\right) + p_{\bar{A}} \cdot \left(p^2_{B\|\bar{A}} + p^2_{\bar{B}\|\bar{A}}\right) - p^2_B - p^2_{\bar{B}}$	[0; 1]	[80]
8	Поддержка (sup)	p_{AB}	[0; 1]	[81]
9	Уверенность (conf)	$p_{B\|A}$	[0; 1]	[81]
10	Мера Лапласа (L)	$\left(n \cdot p_{AB} + 1\right) / \left(n \cdot p_B + 2\right)$	[0; 1]	[82]
11	Убеждение (V)	$p_A \cdot p_{\bar{B}} / p_{A\bar{B}}$	[0.5; ∞]	[83]
12	Фактор интереса (I)	$p_{AB} / \left(p_A \cdot p_B\right)$	[0; ∞]	[84]
13	Косинус (IS)	$p_{AB} / \sqrt{p_A \cdot p_B}$	[0; 1]	[84]
14	Мера Пятецкого-Шапиро (PS)	$p_{AB} - p_A \cdot p_B$	[-0.25; 0.25]	[45]

1	2	3	4	5
15	Фактор определенности (F)	$(p_{B\|A} - p_B)/(1 - p_B)$	[-1; 1]	[80]
16	Добавочное значение (AV)	$p_{B\|A} - p_B$	[-0.5; 1]	[80]
17	Коллективная сила (S)	$\frac{p_{AB} + p_{\overline{AB}}}{p_A p_B + p_{\bar{A}} \cdot p_{\bar{B}}} \times \frac{1 - p_A p_B - p_{\bar{A}} p_{\bar{B}}}{1 - p_{AB} - p_{\overline{AB}}}$	[0; ∞]	[85]
18	Мера Жаккара (ζ)	$p_{AB}/(p_A + p_B - p_A \cdot p_B)$	[0; 1]	[86]
19	Мера Клозгена (K)	$\sqrt{p_{AB}} \cdot (p_{B\|A} - p_B)$	$[\sqrt{2/\sqrt{3}-1} \cdot (2 - \sqrt{3} - 1/\sqrt{3}); 2/(3 \cdot \sqrt{3})]$	[77]
20	Информационная выгода (IG)	$\log(p_{AB}/(p_A \cdot p_B))$	$[-\infty; \log(1/p_A)]$	[78]
21	Мера Сибега и Шонауера (SEB)	$p_{AB}/p_{A\bar{B}}$	[0; ∞]	[87]
22	Коэффициент регрессии	$(p_{AB} - p_A \cdot p_B)/(p_A \cdot (1 - p_A))$	[-1, 1]	[88, 89]

Итак, можно сделать вывод, что всем трем критериям удовлетворяют только шесть мер, а именно *уверенность*, *мера Лапласа*, *J-мера*, *индекс Джини*, *фактор определенности* и *коэффициент регрессии*. Однако, в ходе экспериментального исследования, помимо перечисленных мер, для выяснения дополнительных свойств имеет смысл проанализировать также *убеждение*, *добавочное значение*, *меру Клозгена* и *меру Сибега и Шонауера*, так как они могут быть нормированы таким образом, чтобы удовлетворять критериям 2 и 3. Результаты формального исследования для перечисленных мер представлены в таблице 2.2.

2.4.3 Экспериментальное исследование мер связи

2.4.3.1 Методика экспериментального исследования

Можно использовать несколько способов экспериментальной оценки способности ассоциативной меры выявлять причинные зависимости между переменными. *Первый* из них – это сравнить множество причин, выявленных некоторой

мерой, со списком причин, которые заранее известны для некоторого набора данных, например, из других исследований. *Второй* способ – это сравнить причинные связи, выявленные с помощью анализируемой меры, со связями, полученными с помощью Байесовской сети доверия [40] для одного и того же набора данных, если следовать общепринятому мнению о том, что причинными связями являются те и только те связи, которые выявляются этой сетью. Последнее утверждение, хотя оно и является спорным, может быть использовано, если опираться на «мнение большинства» экспертов, а также на результаты работ [28, 39], в которых показано, что байесовская сеть позволяет выделить связи, весьма полезные для восстановления модели целевой переменной. Существуют, однако, работы, которые предостерегают от интерпретации причинности, устанавливаемой по свойствам связей в Байесовской сети доверия, например, [41].

Таблица 2.2 – Значения критериев для мер, участвующих в экспериментальном исследовании.

Обозн.	Мера	Критерий 1	Критерий 2	Критерий 3
J	J-мера	Да	Да	Да
G	Индекс Джини	Да	Да	Да
conf	Уверенность	Да	Да	Да
L	Мера Лапласа	Да	Да	Да
V	Убеждение	Да	Нет	Да
AV	Добавочное значение	Да	Нет	Да
K	Мера Клозгена	Да	Нет	Да
SEB	Мера Сибега и Шонауера	Да	Нет	Да
F	Фактор определенности	Да	Да	Да
R	Коэффициент регрессии	Да	Да	Да

Добавим, что экспоненциальная сложность построения Байесовской сети дополнительно снижает ценность ее практического использования как эталона для

построения причинных связей. Более того, она исключает возможность ее использования при работе с большими данными. Этот способ исследования не рассматривается в данной работе.

Конечной целью поиска причинных связей в большинстве задач анализа данных бывает их последующее использование в моделях предсказания значений целевой переменной. С другой стороны, имеются результаты, которые говорят о том, что причинные связи лучше других связей работают в моделях предсказания. На основании этих двух посылок можно выдвинуть гипотезу о том, связь, которая удовлетворяет аксиомам, перечисленным выше, и которая хорошо работает в задаче предсказания целевой переменной, является причинной связью. Тогда можно предложить еще один чисто прагматический способ оценки свойств меры связи с позиций ее способности выявлять связи причинного характера: если правила, построенные для предсказании значений целевой переменной, эффективно работают в построенной модели, а описываемые используемые ими связи удовлетворяют перечисленным выше аксиомам, то эти связи с большой вероятностью имеют причинный характер. Это *третий* способ выявления причинности.

Четвертый способ сравнения мер в рассматриваемом здесь смысле – это участие в соревновании по выявлению причинных связей и сравнение своих результатов с результатами, полученными с использованием других мер. Оказывается, что такие соревнования проводятся на некоторых международных конференциях.

Далее для сравнения мер используются все перечисленные подходы за исключением методики, основанной на построении Байесовской сети. Далее описываются результаты экспериментальных исследований мер и сравнения , выделенных в предыдущем разделе по формальным свойствам, для различных наборов данных.

2.4.3.2 Причинные правила для набора данных Adult (Способ 1)

В ходе исследования удалось найти только одну работу [90], в которой приводится список (точнее его фрагмент) правил, отражающих направленную причинную связь переменных и целевой переменной (назовем их для краткости при-

чинными правилами) для модифицированного открытого набора данных *Adult* [91].

Этот набор данных представляет собой фрагмент базы данных переписи населения, проведенной бюро переписи населения США в 1994 году. Задача предсказания состоит в том, чтобы с помощью модели, построенной на основании этих данных, можно было бы для любого индивида предсказать, зарабатывает ли он более $50,000 в год. Модифицированный набор данных Adult, использованный в работе [90], содержит 30160 примеров без пропущенных значений, каждый из которых описан 99 бинарными признаками.

Для выявления причинных правил в работе [90] используется методика CAR, разработанная авторами, основанная на ретроспективном *когортном* (англ. *cohort*) исследовании, широко применяемом в медицине и социологии. С деталями методики и алгоритмами извлечения причинных правил можно ознакомиться в первоисточнике [90].

Причинные связи, найденные с помощью методики CAR, представляются авторами в виде однолитерных правил с меткой класса в качестве заключения (эта метка и является целевой переменной):

Если <атрибут>=true, то класс = <метка класса>, (3.1)

причем метка класса принимает значение 1, если индивид зарабатывает более $50,000, и значение 0 – в противном случае. Полученный список правил авторы работы [90] сравнивают с правилами, выявленными ими же с помощью двух других методик построения локальных причинных структур, а именно, методики CCC [92] и методики CCU [70].

Эксперимент, проведенный в рамках данной работы, включает формирование наборов причинных правил с помощью каждой из анализируемых мер двумя способами (вариант 1 и вариант 2) и их сравнение с результатами, полученными в работе [90].

Опишем кратко алгоритм извлечения причинных правил из данных с помощью мер причинной связи. Сначала для каждого бинарного атрибута в данных формируются два правила следующего вида: (1) «*если <атрибут>=true, то класс*

= 0» и (2) «*если <атрибут>=true, то класс = 1»*. Затем для каждого из этих правил вычисляется одна из исследуемых мер μ. После этого правило (1 или 2), которое имеет меньшее значение исследуемой меры μ, отбрасывается. Списки правил, полученные для каждой исследуемой меры в отдельности, являются итоговыми списками первого варианта данного способа исследования. Затем списки правил, полученные для каждой исследуемой меры, фильтруются с использованием некоторого порога θ для значения этой исследуемой меры[1]. Полученные в итоге сокращенные списки правил рассматриваются как итоговые для второго варианта данного метода исследования. Детали эксперимента и все полученные численные результаты представлены в приложении А.

При формировании правил в ходе первого варианта полное совпадение с CAR в эксперименте продемонстрировали *коэффициент регрессии*, *убеждение*, *добавочное значение*, *фактор определенности* и *мера Клозгена*. При сравнении со списками правил, полученных с помощью методик CCC [92] и CCU [70], полное совпадение по части наличия правил в списке также продемонстрировали *коэффициент регрессии*, *убеждение*, *добавочное значение*, *фактор определенности* и *мера Клозгена*. В таблице 2.3 представлены обобщенные результаты эксперимента для первого варианта формирования правил. В последней строке приведена суммарное число совпадений правил, сформированных с помощью соответствующих метрик, с правилами, полученными методиками CAR, CCC, CCU.

Приведем результаты эксперимента при формировании правил в ходе второго варианта. Наилучшее совпадение со списком правил из работы [90] в этом случае демонстрирует *мера Клозгена* (18 правил из 30), далее следуют *коэффициент регрессии* и *добавочное значение* (12 правил из 30). Сравнение со списками правил, построенных методиками CCC и CCU показывает, что наилучшее совпадение вновь показывает *мера Клозгена*, далее следуют *коэффициент регрессии*, *добавочное значение, фактор определенности* и *убеждение*. В таблице 2.4 пред-

[1] Выбор значения порога θ зависит от решаемой задачи и конкретного набора данных. Методика выбора этого значения выходит за рамки данного исследования

ставлены обобщенные результаты эксперимента для второго варианта формирования правил.

Таблица 2.3 - Обобщенные результаты эксперимента (способ 1) для первого варианта формирования правил

Методика	Всего	J	G	conf	L	V	AV	K	SEB	F	R
CAR	30	22	14	20	20	30	30	30	20	30	30
CCC	20	16	8	15	15	20	20	20	15	20	20
CCU	18	15	8	13	13	18	18	18	13	18	18
Кол-во совпадений		53	30	38	38	68	68	68	38	68	68

Таблица 2.4 – Обобщенные результаты эксперимента (способ 1) для второго варианта формирования правил

Методика	Всего	J	G	conf	L	V	AV	K	SEB	F	R
CAR	30	8	10	8	8	10	12	18	8	10	12
CCC	20	5	6	8	8	10	10	11	8	10	10
CCU	18	4	6	7	7	9	8	10	7	9	8
Кол-во совпадений		17	22	23	23	29	30	39	23	29	30

Следует отметить, что перечисленные меры, помимо правил, генерируемых методами CAR, CCC и CCU, генерируют и другие правила. Их можно отнести как к лишним правилам, так и к правилам, которые являются причинными, но пропущены методами CAR, CCC и CCU. Заметим также, что в работе [90] представлены только 30 правил из 50 правил, найденных авторами, что несколько затрудняет сравнение мер в ходе исследования.

Однако, простого сравнения количества общих правил в списках, полученных разными методами, недостаточно для того, чтобы сделать окончательные выводы, тем более, если сравнение выполнено только на одном наборе данных. Более убедительные результаты можно получить с помощью способа 3, который рассматривается в следующем подразделе.

2.4.3.3 Сравнение мер на основании их классификационных возможностей (Способ 3)

На основе наборов правил, полученных с помощью мер, и набора правил из работы [90] были построены простые классификаторы для модифицированного набора данных Adult следующим образом:

– каждое правило из списка, сформированного для каждой меры, рассматривалось как отдельный классификатор, который может голосовать только за тот класс, который представлен в заключении соответствующего правила;

– объединение решений этих отдельных классификаторов выполняется по схеме простого голосования.

Для оценки качества исследуемых классификаторов использованы следующие данные: матрица неточностей (англ. *confusion matrix*); среднеквадратическое отклонение; чувствительность (англ. true positive rate), коэффициент ложной тревоги (англ. *false positive rate – FPR*), точность (англ. *precision*), полнота (англ. *recall*), F-мера (англ. *F-measure*) классификатора для каждого класса; площадь под ROC-кривой для классификатора (англ. AUC) [93, 94]. Тестирование классификаторов выполнялось с помощью процедуры скользящего контроля (англ. cross-validation) с 10 блоками (англ. 10-fold cross-validation) [95]. Детальное представление результатов размещено в приложении А, где приведены оценки построенных классификаторов по всем перечисленным метрикам. Дополнительно для сравнения использовался алгоритм классификации BayesNet [41], реализованный в системе Weka [96]. Напомним, что речь идет о классификации модифицированного набора данных Adult. Часть результатов эксперимента представлен в таблице 2.5.

Таблица 2.5 – Результаты анализа классификаторов (фрагмент)

	J	G	V	AV	K	F	R	CAR	Bayes-Net
RMSE	0.785	0.722	0.479	0.482	0.474	0.479	0.479	0.507	0.3659
AUC	0.527	0.5	0.799	0.799	0.792	0.797	0.798	0.6	0.745
Precision	0.658	0.629	0.835	0.836	0.83	0.835	0.835	0.761	0.81
FPR	0.331	0.476	0.177	0.169	0.193	0.176	0.175	0.366	0.326

По результатам анализа результатов эксперимента, полностью приведенных в приложении А, можно сделать следующие выводы:

1. Классификаторы, использующие набор правил, полученных с помощью мер *уверенность*, *мера Лапласа* и *мера Сибега и Шонауэра*, являются неэффективными и непригодны для классификации, так как не позволяют классифицировать экземпляры одного из классов набора данных, поэтому далее они не рассматриваются.

2. Наихудшими по всем параметрам являются меры *индекс Джини и J-мера*.

3. Наименьшее *среднеквадратичное отклонение* показал классификатор на основе *меры Клозгена*. Эта мера также оказалась лучшей среди оставшихся по показателям *чувствительность, полнота* и *F-мера.*

4. Классификаторы на основе мер *коэффициент регрессии*, *добавочное значение*, *фактор определенности* и *убеждение* незначительно хуже классификатора на основе *меры Клозгена* по показателям *среднеквадратичное отклонение, чувствительность, полнота* и *F-мера,* однако, лучше его по показателям *коэффициент ложной тревоги* и *точность*, что представляется очень важным.

5. Классификаторы на основе мер *коэффициент регрессии*, *добавочное значение, убеждение*, *фактор определенности* и *меры Клозгена*, превосходят классификатор, построенный на основе правил, представленных в [Li13], *по всем характеристикам*.

6. Наилучшее значение *площади под ROC-кривой* продемонстрировали классификаторы, построенные на основе *коэффициента регрессии*, *добавочного значения* и *убеждения*, *фактора определенности* и *меры Клозгена*.

Отметим, что классификатор *BayesNet*, построенный в инструментальной среде *Weka*, смог классифицировать все экземпляры и оказался наилучшим по показателям *среднеквадратичное отклонение, чувствительность, полнота* и *F-мера*. Однако, его построение заняло несоизмеримо большее время, и он уступил классификаторам на основе *добавочного значения*, *убеждения*, *коэффициента регрессии*, *фактора определенности* и *меры Клозгена* по показателям *коэффициент ложной тревоги, точность* и *площадь под ROC-кривой*.

Полученные результаты позволяют сделать вывод о том, что классификатор *BayesNet* не подходит *для работы с большими данными*, в отличие от алгоритмов классификации на основе правил, полученных с помощью мер *коэффициент регрессии*, *добавочное значение*, *убеждение*, *фактор определенности* и *меры Клозгена*.

2.4.3.4 Соревнование «Causality Challenge №1: Causation and Prediction» (Способ 4)

Соревнование *«Causality Challenge №1: Causation and Prediction»* [97] проходило с 15 декабря 2007 г. по 30 апреля 2008 г. Участникам были предложены четыре задачи в различных прикладных областях. Целью соревнования было как можно точнее предсказать значение бинарной целевой переменной для тестовых выборок. Кроме того, участникам было предложено представить список переменных (признаков), которые они использовали в предсказании. Задачи были разработаны таким образом, что знание причинно-следственных связей между переменными и целевой переменной должно было повышать точность предсказания (классификации). Эффективность поиска причинных связей соответствовала оценке, которая показывала, насколько хорошо признаки, отобранные участниками, совпадают с Марковским покрытием[2] [28, 40] для целевой переменной в тестовой выборке. Частью анализа результатов соревнования было исследование вопроса о том, насколько эта оценка коррелирует с точностью прогнозирования. Несмотря на то, что конкурс закончился в апреле 2008 года, платформа открыта для загрузки внеконкурсных результатов и просмотра всех результатов соревнования.

Для экспериментов с мерами связи, рассмотренными в данной работе, использован набор данных *CINA*, содержащий 16033 обучающих и 10000 тестовых примеров, каждый из которых описывается 132 бинарными признаками. Он был сформирован на основе набора данных *Adult*, имена атрибутов в котором были скрыты. К исходным 44 атрибутам было добавлено 88 искусственно созданных

[2] Напомним, это понятие относится к связям, которое находятся с помощью Байесовской сети доверия.

переменных, которые не являются причинами целевой переменной. В обучающих данных некоторые из этих искусственных переменных являются следствиями целевой переменной и/или других исходных переменных. Другие переменные не имеют ассоциативных связей с ними. Таким образом, некоторые из искусственных переменных могут коррелировать с целевой переменной в обучающих данных, но они не являются ее причинами.

Внеконкурсные результаты соревнования для набора данных *CINA0* были получены для *меры Клозгена*, *коэффициента регрессии*, *добавочного значения*, *убеждения* и *фактора определенности*. Для каждой из перечисленных мер были сформированы списки выявленных ими причин, а также построены классификаторы. Списки причин и результаты классификации загружены на сайт соревнования.

Для сравнения результатов участников соревнования использовались следующие оценки:

1. *Fscore* – оценка соответствия причинно-следственных связей, представленных участником, реальным причинно-следственным связям в экспериментальном наборе данных. Список всех истинных причинно-следственных связей между признаками (переменными) и целевой переменной известен только организаторам. *Fscore* представляет собой площадь под *ROC*-кривой для классификации «причинных» и «не причинных» признаков набора данных, представленного участникам соревнований. При этом «не причинными» организаторы считают признаки, которые не относятся к Марковскому покрытию целевой переменной.

2. *Dscore* – оценка, соответствующая площади под *ROC*-кривой для классификации примеров обучающего набора.

3. *Tscore* – оценка, соответствующая площади под *ROC*-кривой для классификации примеров тестового набора.

Отметим, что для ранжирования участников в турнирной таблице соревнования использовалась только оценка *Tscore*. Но так как целью данной работы является выявление причинных связей в данных, то в рамках проводимого исследования более важной является оценка *Fscore*.

Внеконкурсные результаты соревнования для набора данных *CINA0* и перечисленных выше мер представлены в таблице 2.6.

Отметим, что значения *Fscore* для всех пяти мер, анализируемых в работе, попали в список 5% лучших результатов. Результаты классификации (*Dscore* и *Tscore*) в списки лучших не попали, но целью этого исследования в данной работе не было построение наилучших классификаторов. Напомним, что в данной работе для оценки качества мер причинности использованы классификаторы, построенные по схеме простого голосования.

Таблица 2.6 – Внеконкурсные результаты соревнования [98]

Мера	Поиск причин	Точность классификации	
	Fscore	Dscore	Tscore
Мера Клозгена	0.9276	0.9015	0.9045
Коэффициент регрессии	0.8893	0.9009	0.9035
Добавочное значение	0.8813	0.8903	0.8952
Убеждение	0.8665	0.9136	0.9136
Фактор определенности	0.8665	0.9118	0.9132

2.5 Выводы: обоснование направления исследований в области ассоциативно-причинной классификации и выбор причинной меры связи

В главе рассмотрена формальная постановка задачи ассоциативной классификации, а также приведен обзор основных результатов, моделей и методов в этой области, которые, как полагают их авторы, ориентированы на обработку данных большого объема. В ходе сравнительного анализа дается оценка научного вклада работ, посвящённых ассоциативно-причинной классификации.

Модели ассоциативной классификации, получившие основное развитие в течение последних пятнадцати-двадцати лет, предлагают подход, интегрирующий в себе механизмы поиска ассоциативных правил и традиционные методы классификации. Целью является повышение вычислительной эффективности и точности решения задач классификации при использовании новых моделей в анализе больших данных. Первые модели ассоциативной классификации рассматривали эту

модель как частный случай задачи поиска ассоциативных правил с тем отличием, что в ассоциативной классификации заключением правила может быть одна из меток класса или ее отрицание. Значительный шаг в развитии этого направления был сделан благодаря разработке группы методов и алгоритмов поиска ассоциативных правил классификации с использованием понятия эмерджентного паттерна. Этот подход определил новое направление в области ассоциативной классификации, которое активно развивается и в настоящее время. Его результаты позволяют преодолеть ряд существенных недостатков начальных моделей и алгоритмов ассоциативной классификации. Работы в этом направлении во многом способствовали более глубокому пониманию специфики задач ассоциативной классификации и путей ее эффективной алгоритмизации.

Следует отметить, что модели и алгоритмы ассоциативной классификации в задачах анализа больших данных имеют также и ряд недостатков. *К основному недостатку* такой модели следует отнести ограниченные возможности при работе с гетерогенными данными сложной структуры, с данными, представленными текстами на естественном языке, изображениями и т.п. Модель ассоциативной классификации хорошо подходит для работы с дискретными данными (булевыми, номинальными и целочисленными). В случае других типов данных принципиальной проблемой становится проблема дискретизации реальных данных. Известные способы такого преобразования приводят к существенному увеличению размерности данных. А проблема размерности является ключевой проблемой больших данных и без их дискретизации.

Другая проблема использования ассоциативных правил обусловлена тем фактом, что ассоциация как связь, по определению, является ненаправленной связью, и ее нельзя интерпретировать как причинно–следственную связь, если не обосновывать это специальными методами или не использовать метрики, которые специально предназначены для обнаружения причинных связей.

Зачастую, способы построения классификаторов на основе ассоциативных правил базируются на эвристиках, на введении специальных метрик для оценки разделяющей способности классификаторов. Для каждого варианта выбора мо-

дели объединения решений, даваемых различными правилами, всегда можно построить пример, когда предложенный вариант совсем не подходит, или когда предложенная эвристика не работает. Эта проблема анализируется в теории объединения решений, которые вырабатываются множеством классификаторов (а каждое ассоциативное правило может интерпретироваться как простейший классификатор), и называется *проблемой разнообразия классификаторов* [74]. От успешности решения этой проблемы во многом зависит успешность выбранной модели ассоциативной классификации при работе с большими данными.

Анализ работ в области ассоциативной и ассоциативно-причинной классификации, который выполнен в текущей главе, позволяет сделать следующие *выводы*:

– методы ассоциативно-причинного анализа оказываются более эффективными в области больших данных, чем методы ассоциативного анализа. Экспериментальные и формальные исследования, выполненные, например, в работах [28, 39], убедительно показали, что в задачах принятия решений именно причинные связи между атрибутами в данных оказываются наиболее перспективными атрибутами для построения правил классификации;

– вычислительная сложность построения байесовских сетей доверия, которые являются наиболее популярной моделью поиска и представления причинных связей, не позволяет использовать их в задачах поиска причинных зависимостей в больших данных;

– задача разработки новых методов и алгоритмов поиска причинных зависимостей в данных имеет особую актуальность. Такие методы должны, прежде всего, обладать вычислительной эффективностью, что может быть достигнуто за счет простых алгоритмов фильтрации множества атрибутов, которые потенциально могут быть кандидатами в число причин для той или иной целевой переменной.

– для практической реализации описанной выше схемы поиска причинных связей в данных необходимо, в первую очередь, выбрать численную метрику для оценки силы причинной связи между переменными, которая должна быть семантически корректной и вычислительно эффективной.

Так как причинная связь переменных является частным случаем ассоциативной связи, то естественно численные меры оценки причинной связи искать среди мер, предложенных для численной оценки силы ассоциативных связей.

В разделе 2.4 решена одна из задач исследований, сформулированная в разделе 1.6, а именно задача выбора наилучшей формальной меры, оценивающей «силу» причинной связи между переменными, которая является семантически корректной и вычислительно эффективной;

В процессе решения этой задачи использованы «аксиомы» для мер причинной связи [89], и из множества мер ассоциативной связи выделены те, которые потенциально могут быть использованы в качестве мер для оценки «силы» причинных связей. Эти меры детально исследованы экспериментально с привлечением различных методологий их сравнения, в частности, для каждой меры

- на заданном тестовом наборе данных проведено сравнение найденных с её помощью правил, с теми правилами, которые получены для этого же набора данных другими исследователями в области причинного анализа данных;

- для заданного тестового набора данных построен простой классификатор, использующий правила, полученные с помощью исследуемой меры, и проведено сравнение по точности решения задачи классификации;

- получены внеконкурсные результаты соревнования «*Causality Challenge №1: Causation and Prediction*» для классификатора, построенного с помощью исследуемой меры, и определено «место», которое этот результат занял бы в соревновании.

На основании анализа экспериментальных результатов сделан вывод о том, что наиболее перспективными с точки зрения способности выявлять правила, представляющие причинные связи в данных, являются следующие меры:

– коэффициент регрессии;
– мера Клозгена;
– убеждение;
– фактор уверенности.

Две наилучшие метрики, а именно*, коэффициент регрессии* и *мера Клозгена*, выбраны для дальнейшего использования при построении семантического профиля пользователя в рекомендательных системах третьего поколения. Окончательный выбор будет сделан по результатам экспериментального исследования на конкретном наборе обучающих данных.

Напомним, что при анализе больших данных *критически важной* характеристикой используемых алгоритмов является их вычислительная *сложность*. Отметим, что вычисление значений перечисленных мер выполняется за *один проход* по данным. По этой причине предложенный вариант оценки правил, которые потенциально могут отражать причинные связи между переменными и целевой переменной, хорошо подходит для решения различных задач анализа больших данных.

3 МЕТОДИКА ПОСТРОЕНИЯ СЕМАНТИЧЕСКОЙ МОДЕЛИ БОЛЬШИХ ДАННЫХ

3.1 Рекомендательные системы и основные проблемы обучения профиля пользователя

Напомним, что рекомендательными системами называют класс систем принятия решений, которые, используя разнородную информацию о предпочтениях человека в различных контекстах, пытаются спрогнозировать его предпочтения, реакцию на предложение некоторой внешней системы, рекомендующей ему некоторый контент, товар или некоторую услугу [29]. Краткие сведения о рекомендательных системах и проблемах, существующих в этой области, представлены в разделе 1.4.

Как было отмечено в разделе 1.4 данной работы, построение современных рекомендательных систем относится к классу задач обработки больших данных, который, кроме того, объединяет в себе и черты многих других задач области машинного обучения и принятия решений. Поэтому многие модели и алгоритмы, которые используются в задаче построения рекомендательных систем, могут быть адаптированы для использования и в других классах приложений, причем не только в области больших данных.

В рекомендательных системах третьего поколения информация о пользователе, которая используется как обучающая информация, как правило, извлекается из всех доступных источников, в которых так или иначе отражаются, так называемые, «следы» его деятельности. В качестве таких источников обычно выступают социальные сети (Facebook, LinkedIn, Twitter и т.д.), история посещенных вебстраниц браузера, история покупок и т.п. Кроме того, в процесс построения профиля пользователя зачастую вовлекается дополнительная информация, объёмы которой в сотни и тысячи раз превосходят начальный объём данных о нем. В качестве источников такой информации обычно выступают облачные ресурсы и глобальные онтологии, такие как WordNet, Wikipedia, DBpedia и ресурсы Linked Data Web. Эти данные, как правило, являются гетерогенными, обладают разной

степенью достоверности и полноты, содержат данные на естественных языках и другую неструктурированную информацию.

Следует заметить, что в рекомендательных системах третьего поколения акцент делается на семантических моделях представления и использования всех компонент знаний о пользователе и о продуктах, которые ему могут быть предложены. Рекомендательные системы третьего поколения должны вырабатывать решения на основе семантических моделей интересов и предпочтений пользователя, принимать во внимание мотивацию и причины, которые побуждают конкретного пользователя делать тот или иной выбор. Такие системы должны давать мотивированные объяснения принятым решениям. Они должны учитывать психологические и другие факторы, связанные к конкретным пользователем [30]. Поэтому технология разработки рекомендательных систем третьего поколения акцентирует внимание на построении и использовании персонифицированного профиля пользователя рекомендательной системы. Важными в них являются также и экономические факторы, связанные с практическим использованием рекомендательных систем.

В настоящее время наиболее разработанной формальной моделью представления семантики данных является онтология [100]. По этой причине в рекомендательных системах третьего поколения именно онтология рассматривается в качестве общей модели представления всех компонент знаний, таких как знания о предметной области рекомендаций, знания об интересах и эмоциональном и психологическом состоянии пользователя, о контексте принятия решений и т.п.

Основная идея онтологической модели профиля пользователя состоит в том, что каждый интерес пользователя рассматривается как экземпляр некоторого понятия онтологии, которое семантически связано с другими ее понятиями. При этом, в зависимости от потребностей задачи, онтология может представляться или как сложная структура понятий с разнообразными бинарными отношениями между ними, или в более бедной форме, когда она задается только иерархической моделью понятий с единственным типом отношения на их множестве– отношением порядка типа «общее - частное», или, что формально одно и то же, отноше-

нием «родитель» - «потомок». Важно отметить еще раз, что понятиями онтологии данных в рекомендательных системах третьего поколения являются понятия, которые потенциально могут отвечать тем или иным интересам пользователя. В этом случае профиль интересов пользователя есть некоторая подструктура общей структуры понятий онтологии предметной области выбора. Можно говорить, что проблема построения персонального профиля пользователя состоит в том, чтобы в потенциальном множестве интересов, представленных иерархией понятий той или иной предметной онтологии, найти некоторое их подмножество, которое определяет его предпочтения при принятии решений в пользу выбора того или иного предлагаемого объекта, например, товара или услуги. Иначе говоря, определение мотивации того или иного выбора пользователя есть одна их ключевых задач рекомендательных систем третьего поколения.

Естественно, что онтология интересов пользователя должна строиться так, чтобы они были представлены в ней наиболее полным образом. Поэтому в рекомендательных системах построение онтологии данных и поиск в ней интересов конкретного пользователя – это базовые, причем тесно связанные задачи, которые решаются путем интеллектуальной обработки одного и того же множества экспериментальных данных.

Подчеркнем, что онтологическая модель профиля пользователя позволяет существенно упростить решение задачи кросс–доменных рекомендаций, выработка которых рассматривается как функциональность рекомендательных систем третьего поколения, так как она позволяет представить интересы пользователя из различных предметных областей в рамках единой структуры понятий.

Методы построения онтологии предметной области и профиля пользователя варьируются от простого опроса экспертов и самого пользователя, до полностью автоматического (машинного) обучения профиля путем обработки всей доступной информации, полученной из разных источников.

Однако, в задачах построения рекомендательных систем третьего поколения, как и других в задачах из области больших данных, число понятий онтологии дан-

ных может исчисляться десятками тысяч и больше, при этом данные, представленные на естественном языке, сильно осложняют задачу. Эти обстоятельства делают ручное построение онтологии данных практически невозможным. Поэтому при создании персонифицированной рекомендательной системы третьего поколения остро стоит проблема автоматизации процессов построения онтологии.

В последующих разделах описан разработанная методика автоматического построения онтологии, управляемая от данных, которая комбинирует методы построения структуры понятий на базе онтологического подхода и методы формального построения структуры понятий на основе данных, которые введены в теории анализа формальных понятий [99]. Методика названа *Семантическим анализом понятий*. Онтология данных, построенная с помощью этой методики, позволяет структурировать понятия - интересы пользователя и вычислять количественную меру важности каждого из них для конкретного пользователя, отражая тем самым его предпочтения.

Предлагаемая далее методика семантического анализа понятий реализует подход к машинному обучению, обеспечивающий интеграцию алгебраической и статистической моделей данных, а также представления модели знаний в рамках единой структуры.

3.2 Семантический анализ понятий

3.2.1 Автоматизированные методы и средства построения онтологий

Как известно, существуют, по крайней мере, два варианта онтологии. Первая из них задает структуру понятий и отношений предметной обрасти до какого-то уровня глубины. Она строится экспертом или в автоматизированном режиме, начиная с понятий самого верхнего уровня (абстракции) [100]. В ней полностью отсутствуют примеры. Другой тип онтологии – это онтология, которая строится по схеме *от данных* (англ. *data –driven ontology*), и по сути своей она является онтологией данных, но не онтологией какой-то конкретной предметной области. Для ее построения существуют подходы и различные инструменты, которые поддерживают автоматизированный режим ее построения. Если речь идет об онтологии в различных задачах обработки больших данных [101], то в этом случае, как

правило, имеется в виду онтология второго типа, а именно онтология данных. То же самое касается онтологий, используемых в рекомендательных системах, поскольку они строятся н основе данных о привычках и предпочтениях пользователя.

Онтология данных как структура для представления знаний, представленных в данных, используется при построении профиля пользователя рекомендательных систем уже достаточно давно (около 20 лет). В ранних работах в этой области (1990-е годы) использовался вариант ручного или частично автоматизированного построения онтологии предметной области, который требовал значительных усилий и временных затрат со стороны экспертов [102, 103, 104].

Первые активные исследования в области автоматизации построения онтологии данных относятся к началу 2000-х годов. Эти усилия привели к разработке ряда крупных лексических онтологий (семантических словарей), таких как WordNet [108], и инструментов извлечения структурированного контента, например, DBpedia [109]. Последний инструмент вместе с инструментами обработки естественных языков и в сочетании с иерархией категорий статей Википедии постепенно стал мощным инструментом извлечения понятий из разнородных данных, включая неструктурированные тексты. Дополнительный источник информации для полуавтоматической разработки онтологий появился в связи с реализацией проекта Open Linked Data Web [35], в рамках которого существует множество компонент онтологий повторного использования для различных наборов данных, представляющих различные предметные области. Различные варианты применения переиспользуемых компонент онтологий из Linked Data Web также рассматриваются в работах [105, 106, 107].

Выделим некоторые этапы развития технологии построения онтологий, а также примеры успешных реализаций технологии полуавтоматической разработки онтологий применительно к рекомендательным системам. Более подробный анализ этих технологий имеется в работе [110].

В работе [111] представлена технология полуавтоматического построения формальной модели персонального профиля пользователя в терминах онтологии,

характеризующей его интересы и намерения. В качестве входной информации в разработанной модели используются URL-адреса, извлеченные из сообщений пользователя в Twitter. Верхний уровень онтологии, используемой авторами, который отвечает категоризации веб-сайтов, формируется вручную с использованием понятий, извлеченных из дополнительных источников. Примерами таких источников являются базы коллективных знаний OpenDNS [112], таксономии рекламных объявлений и онтологии DBpedia. Далее, для каждого URL-адреса, найденного в твитах пользователя, из баз коллективных знаний OpenDNS и DBpedia автоматически извлекаются категории и понятия, которые добавляются к онтологии профиля пользователя. Отметим, что методика, предложенная в [111], позволяет извлечь только достаточно общие интересы пользователя с невысокой точностью. Этот подход может быть использован только для первичного извлечения областей интересов пользователя, которые затем могут быть утонены, например, путем анализа содержания страниц, размещенных по URL-адресу из твита.

В работе [113] описывается подход к разработке онтологий предметных областей на основе *фолксономий*[3] тегов с последующим их обогащением с помощью данных и онтологий проекта Linked Open Data. На выходе авторы получают облегченную версию онтологии предметной области, которая объединяет в себе эмерджентные знания, появляющиеся в ходе совместной категоризации произвольно выбираемых ключевых слов (тегов) и формальных знаний из существующих онтологий. Авторы демонстрируют работу предложенной методики на примере предметной области «Финансы» для тегов из набора данных Delicious [115] с использованием баз данных понятий DBpedia, OpenCyc [116] и UMBEL [117] в качестве дополнительных источников знаний. Разработанную авторами методику можно также описать как извлечение знаний о предметной области из масштаб-

---

3 *Фолксономия* (англ. *folksonomy*, от *folk* - народный + *taxonomy* таксономия, от гр. расположение по порядку) - «народная» классификация, практика совместной категоризации информации (текстов, ссылок, фото, видео клипов и т. п.) посредством произвольно выбираемых меток, называемых тегами [114].

ных баз знаний с помощью терминологии предметной области, полученной из фолксономии. Построенная онтология может быть использована инженерами по знаниям как отправная точка при разработке подробной онтологии предметной области. Данную работу можно считать важным шагом в сторону автоматизации построения онтологии. Работа хорошо иллюстрирует, как можно использовать публичные (разработанные сообществом специалистов) базы знаний, поддерживаемые проектом Linked Open Data, для обогащения информации о пользователе, например, путем привлечения фолксономии тегов. Существует несколько других работ, предлагающих подобный подход с некоторыми вариациями.

Авторы [118] предлагают использовать метод, основанный на анализе формальных понятий, для построения онтологии в виде таксономии (иерархии) понятий. Авторы считают, что таксономия является «скелетом» онтологии и именно с нее начинается построение полноценной онтологии. Эта идея выглядит достточно перспективной. Метод, предложенный в [118], заключается в следующем. Эксперт с помощью программного средства для анализа формальных понятий строит решетку понятий, соответствующую имеющимся данным. Затем, анализируя эту решётку, эксперт вручную проектирует готовую онтологию, исключая из нее понятия или добавляя в нее новые понятия. Однако работа с интерпретацией формальных понятий является достаточно трудоемкой ввиду их потенциально огромного количества. Кроме того, представляется, что авторы [118] лишь частично используют потенциальные возможности формального анализа понятий для автоматизированного построения онтологии данных в интересах обучения профиля пользователя и принятия решений в рекомендующих системах.

Сходное понимание проблемы отражено в работе [119], авторами которой разработан плагин для Protégé 2000, который позволяет использовать анализ формальных понятий как вспомогательное средство при построении онтологии, используя ту же технологию, что и описанная в работе [118].

Та же идея обсуждается и в работе [120]. Авторы [120] считают, что два формализма, а именно онтология и решетка формальных понятий могут дополнять друг друга в прикладном аспекте. Авторы [120] полагают, что подход с позиций

формального анализа понятий может быть использован в качестве метода обучения для построения онтологии и для анализа уже построенной онтологии. Онтологии же, в свою очередь, могут быть использованы для улучшения приложений, использующих формальный анализ понятий. И в том, и в другом случае авторы [120] предполагают активное участие эксперта.

Таким образом, идея объединения анализа формальных понятий и модели данных на основе онтологии, которая добавляет семантику к формальным понятиям, обсуждается уже длительное время. Тем не менее, эта идея не была ранее разработана до такой глубины, которая позволила бы полностью автоматизировать технологию построения онтологий.

3.2.2 Анализ формальных понятий как средство извлечения понятий из данных

Рассмотрим некоторые аспекты модели формального анализа понятий для того, чтобы в дальнейшем использовать его в процессах автоматического построения онтологии данных. Заметим, что особенностью задачи автоматического построения онтологии в рекомендующих системах является то, что единственным источником информации является выборка данных. В случае рекомендательной системы такая выборка может представлять собой логи мобильного устройства пользователя, историю посещения веб-сайтов, историю покупок и т.д. Таким образом, эта задача должна решаться в режиме управления от данных.

Проблема извлечения и анализа понятий с управлением от данных является предметом научного направления в области интеллектуального анализа данных, которое принято называть анализом формальных понятий (англ. *Formal Concept Analysis, FCA*) [99]. Хотя базовые идеи этого направления были сформулированы еще в 1960-х годах, анализ формальный понятий как отдельное научное направление появилось совсем недавно, в середине 1990-х годов. На данный момент исследования в этой области представлены внушительным количеством публикаций, в которых, в основном, исследуются его математические аспекты.

Анализ формальный понятий (АФП) - это теория и технология, предназначенная для извлечения агрегированных сущностей из данных. Они называются

формальными понятиями и каждому из них ставится в соответствие подмножество экземпляров данных обучающей выборки, которое называется объемом понятия. Если говорить неформально, то каждое понятие (англ. *intent*) характеризуется одним или несколькими свойствами, а объем этого понятия (англ. *extent*) – это множество примеров данных, которые всеми этими свойствами обладают.

Формально, пусть $\boldsymbol{A}=\{A_i\}|_{i=1}^{m}$ есть множество, в общем случае, свойств (в терминологии машинного обучения - понятий, признаков), $\boldsymbol{B}=\{B_j\}|_{i=1}^{n}$ есть множество объектов (примеров выборки обучающих данных в нашем случае), и задано бинарное отношение $\boldsymbol{I}\subseteq\boldsymbol{B}\times\boldsymbol{A}$ из множества объектов в множество свойств, называемое отношением инцидентности. Оно интерпретируется следующим образом: для любого объекта $B_j\in\boldsymbol{B}$ и свойства $A_i\in\boldsymbol{A}$ имеет место отношение $\boldsymbol{I}$, если B_j обладает свойством A_i. Формально это факт отражается выражением $B_j\ \boldsymbol{I}\ A_i$. Тройка $\boldsymbol{K}=<\boldsymbol{B},\boldsymbol{A},\boldsymbol{I}>$, представляющая бинарное отношение $\boldsymbol{I}$ на множестве $\boldsymbol{B}\times\boldsymbol{A}$, называется формальным контекстом данных.

Пусть $\boldsymbol{A}_i\subseteq\boldsymbol{A}$ и $\boldsymbol{B}_j\subseteq\boldsymbol{B}$ – некоторые подмножества свойств и объектов соответственно, и задана пара отображений:

$\varphi:\boldsymbol{2}^{\boldsymbol{A}}\to\boldsymbol{2}^{\boldsymbol{B}}$, которое каждому подмножеству свойств $\boldsymbol{A}_i\subseteq\boldsymbol{A}$ ставит в соответствие подмножество объектов $\boldsymbol{B}_j\subseteq\boldsymbol{B}$, которые этим свойством обладают, и

$\psi:\boldsymbol{2}^{\boldsymbol{B}}\to\boldsymbol{2}^{\boldsymbol{A}}$, которое каждому подмножеству объектов $\boldsymbol{B}_j\subseteq\boldsymbol{B}$ ставит в соответствие подмножество свойств $\boldsymbol{A}_i\subseteq\boldsymbol{A}$, которыми эти объекты обладают.

Заметим, что множества $\boldsymbol{2}^{\boldsymbol{A}}$ и $\boldsymbol{2}^{\boldsymbol{B}}$ (множества всех подмножеств для множеств, указанных в степени $\boldsymbol{2}$) являются частично упорядоченными множествами по отношению включения $\subseteq$. Пара указанных отображений задает так называемое соответствие Галуа [121] между элементами частично упорядоченных множеств $<\boldsymbol{2}^{\boldsymbol{A}},\subseteq>$ и $<\boldsymbol{2}^{\boldsymbol{B}},\subseteq>$.

Доказано, что композиция этих отображений $\varphi\psi$ является операцией замыкания (оно называется замыканием Галуа), т.е.

$$\varphi\,\psi(A_i)=A_i \text{ и } \psi\,\varphi(B_j)=B_j.$$

С учетом введенных понятий и их обозначений формальным понятием контекста $K=<B, A, I>$ называется пара $<A_i, B_j>$, $A_i \subseteq A$, $B_j \subseteq B$, для элементов которой справедливо $B_j=\varphi(A_i)$ и $A_i=\psi(B_j)$. Подмножество A_i, которое соответствует некоторому подмножеству свойств, принято называть содержанием формального понятия $<A_i, B_j>$ (англ. *intent*), а подмножество B_j, которое соответствует множеству примеров обучающих данных, обладающих всем набором свойств A_i, принято называть объемом понятия (англ. *extent*).

АФП используется для того, чтобы помочь специалистам обнаружить в множестве данных некоторые формально присутствующие агрегаты экземпляров данных с общими свойствами, которые потенциально могут соответствовать семантически интерпретируемым понятиям предметной области, скрытым в данных. Если говорить применительно к рекомендательным системам, то некоторые из этих формальных понятий могут отвечать интересам пользователя.

К сожалению, АФП не содержит каких-либо формальных средств интерпретации понятий. Поэтому построение смысловой интерпретации выявленных формальных понятий, если она существует, остается за специалистом в предметной области, что сильно усложняет процесс извлечения знаний из данных с использованием АФП.

Другой трудной проблемой АФП применительно к рассматриваемой задаче является его вычислительная сложность. АФП может генерировать огромный набор потенциальных понятий, анализ семантики которых и отбор полезных понятий для формирования онтологии данных остается задачей специалиста-эксперта в предметной области. Как правило, эта задача очень трудоемка.

Еще один существенный недостаток АФП связан с тем, что он может работать только с хорошо структурированными данными, в которых каждый экземпляр данных задается в терминах атрибутов (признаков), имеющих смысл

свойств, т.е. заданных в шкале наименований[4]. Однако в реальности такой случай встречается достаточно редко. Обычно большие данные, включая данные, которые пользователь генерирует с помощью своего мобильного устройства, содержат, наряду с данными типа свойств, также и другие структурированные и неструктурированные и/или плохо структурированные данные, такие, например, как тексты на естественных языках. Это означает, что даже если игнорировать вычислительную сложность АФП, для его использования в задаче построения онтологии данных последние необходимо предварительно привести к одному типу, а именно, к номинальному типу.

Более подробная информация об анализе формальных понятий на русском языке может быть найдена в работе [122].

3.2.3 Содержательная характеристика семантического анализа понятий и его базовой идеи

Рассмотрим методику семантического анализа понятий применительно к построению персонифицированного профиля интересов пользователя мобильного устройства для ее использования в персональной рекомендательной системе. Очевидно, что эта задача ввиду ее массовости (сколько мобильных устройств на руках, столько требуется и персонифицированных рекомендательных систем) не может решаться с привлечением экспертов персонально для каждого пользователя. Эта задача будет иметь практический и коммерческий смысл только в том случае, если она решается автоматически и средствами самого мобильного устройства. Последнее позволит решить проблему конфиденциальности данных пользователя. По сути, речь идет об автоматическом обучении персонифицированной рекомендательной системы в автономном режиме.

Будем полагать, что выборка данных для построения профиля интересов пользователя включает в себя логи, записанные на его мобильном устройстве. Это могут быть, например, логи истории посещения им веб-сайтов. Поскольку в таких логах представлены URL-адреса, то получить тексты страниц, которые посещал

[4] Другие названия этой шкалы – номинальная, или кардинальная шкала

пользователь и собрать информацию, которая является хорошим источником знаний о его интересах, не представляет трудностей.

Однако, извлечение знаний из таких источников затруднено тем, что содержащаяся в них информация не структурирована. Отметим также, что логи, как правило, не содержат информации об отношении пользователя к той или иной посещенной им веб-странице. По этой причине задача извлечения знаний для обучения профиля пользователя в рассматриваемой ситуации является задачей машинного обучения без контр-примеров, или, другими словами, без учителя, что дополнительно осложняет её решение.

Подчеркнем, что основные проблемы семантического моделирования больших данных, в том числе, данных, которые используются для построения профиля пользователя в рекомендательных системах, обычно связаны именно с неструктурированными данными, чаще всего, с текстами на естественном языке. По этой причине рассматриваемая далее методика построения онтологии данных на основе семантического анализа понятий демонстрируется на примере использования выборки, которая содержит множество текстов.

Основная идея семантического анализа понятий (САП) – это совместное использование возможностей и преимуществ онтологической спецификации семантики понятий предметной области, извлеченных средствами анализа понятий естественного языка, дополненных вспомогательными средствами поиска типа DBpedia, с одной стороны, и технологии, близкой к АФП, реализующей генерацию формальных понятий для имеющегося множества данных. Дадим содержательное описание этой идеи.

Известно, что иерархия понятий онтологии средствами DBpedia строится по принципу *обобщения* («от примеров к понятиям и от них – к более общим понятиям»), а АФП строит формальные понятия по принципу *специализации* («от множества признаков данных и исходных формальных понятий к более частным понятиям»). САП комбинирует оба эти процесса, выполняя их параллельно и итеративно. Более точно, иерархия понятий онтологии стро-ится параллельно в построением структуры формальных понятий и их взаимодействие используется для

управления процессом построения онтологии данных. При этом используется тот факт, что эти структуры, иерархия понятий онтологии и иерархия формальных понятий, как будет показано далее, являются двойственными структурами. Поэтому получаемые в итоге некоторые численные характеристики иерархии понятий онтологии и иерархии формальных понятий оказываются связанными некоторыми зависимостями, что и используется для обеспечения эффективности и результативности итогового процесса построения онтологии данных.

В САП оба подхода стартуют от одного и того же множества понятий (далее они будут называться *базовыми понятиями*) и реализуют на каждом шаге попеременно сначала их обобщение (при построении понятий очередного уровня онтологии данных) и затем специализацию (при построении такого же уровня формальных понятий).

Как было отмечено ранее, основные проблемы практического применения АФП состоят в том, что в нем отсутствуют формальные средства семантической интерпретации генерируемых понятий, а сам процесс их генерации обладает экспоненциальной сложностью. Обычно процесс генерации структуры формальных понятий останавливается тогда, когда все минимальные элементы решетки формальных понятий будут иметь пустой объем, т.е. когда для каждого из них в множестве данных не найдется ни одного примера. Установление порога по минимальной мощности объема формального понятия является другим средством останова процесса генерации понятий в АФП. Но обычно последнее ограничение мало помогает ввиду экспоненциального роста количества формальных понятий, поскольку в итоговой решетке формальных понятий среди огромного их количества только небольшая часть обычно отвечает семантически интерпретируемым понятиям естественного или проблемно-ориентированного фрагмента языка, которым оперирует человек в предметной области. Например, в задаче обучения профиля пользователя в рекомендательных системах такими семантически интерпретируемыми понятиями являются понятия, представляющие потенциальные интересы пользователя.

Как уже отмечалось, в современной концепции использования АФП задачу отбора семантически осмысленных формальных понятий должен выполнять эксперт в предметной области, к которой относятся данные. А это обычно очень трудоемкая задача прямого перебора формальных понятий для означивания их семантики. Такой подход совсем не подходит для рекомендательных систем, устанавливаемых на мобильных устройствах, где вмешательство эксперта практически исключено.

Выходом из этой ситуации является поиск автоматического способа семантической интерпретации формальных понятий. Действительно, если можно было бы установить, что очередное сгенерированное формальное понятие не является семантически осмысленным, то его удаление из решетки формальных понятий привело бы к резкому уменьшению числа генерируемых понятий, т.е. к существенному повышению вычислительной эффективности и к исключению эксперта из процесса построения решетки формальных понятий. Последнее является особенно важным. Другое желательное свойство методики формирования понятий онтологии данных состоит в том, чтобы она была в состоянии определять необходимый и достаточный уровень обобщения понятий онтологии, избегая тем самым генерации избыточного их количества. Избыточность их определяется тем фактом, что они могут оказаться бесполезными с точки зрения улучшения качества решений рекомендательной системы.

В разработанной методике оба свойства обеспечиваются именно за счет пошагового итеративного взаимодействия процессов обобщения понятий онтологии и процессов построения структуры формальных понятий. Поясним эту идею на содержательном уровне, а затем обоснуем ее использование более формально.

Пусть данные для обучения профиля пользователя представляются множеством текстов на естественном языке. Для извлечения понятий из текстов можно использовать любой существующий NLP[5]-инструмент, например, DBpedia Spotlight Service [109] или некоторый инструментарий компании IBM.

---

[5] NLP – это аббревиатура английского термина Natural Language Processing, обработка естественного языка.

Предположим, что с помощью NLP-инструмента для имеющихся данных построено множество базовых понятий. Каждое понятие обычно соответствует некоторому свойству (оно представлено в номинальной шкале), которым конкретный пример данных обучения может обладать, а может и не обладать. Полагаем что каждому понятию базового уровня (свойству данных) поставлено в соответствие множество примеров данных, соответствующих этому понятию. Напомним (см. подраздел 4.2.2), что это множество примеров в АФП называется объемом понятия. Далее, каждому понятию можно поставить в соответствие мощность множества его объема, а также значение эмпирической вероятности понятия по имеющейся обучающей выборке данных (относительную частоту, с которой это свойство появляется в выборке данных).

Перечисленные данные (множество базовых понятий, объем этого понятия и эмпирическая частота понятия в данных) вместе с выборкой данных представляют вход процедуры автоматического построения иерархии понятий онтологии данных.

3.2.4 Обоснование алгоритма автоматического построения онтологии данных на основе семантического анализа понятий

Рассмотрим кратко обоснование разработанного алгоритма решения поставленной задачи. Известно, что любое множество (например, в нашем случае речь идет о множестве понятий базового уровня) с двумя бинарными операциями, которые удовлетворяют аксиомам коммутативности, ассоциативности и поглощения, образуют структуру, называемую алгебраической решеткой (далее для краткости – решеткой), для которой исходное множество является системой образующих [123, 124].

Напомним некоторые простые и хорошо известные факты из теории решеток, которые имеют отношение к рассматриваемой далее формальной модели семантического анализа понятий. Обозначим множество всех базовых понятий (свойств), извлеченных из данных, символом $\boldsymbol{A}=\{A_k\}|_{k=1}^{n}$, где A_k, $k=1,\ldots,n$ – базовые понятия, формирующие множество свойств $\boldsymbol{A}$, а операции обобщения и

специализации понятий обозначим символами $\vee$ – для обобщения и $\wedge$ – для специализации. По определению операций обобщения и специализации, понятие $A_k \vee A_i$ представляет новое свойство примера, которым этот пример обладает, если он обладает хотя бы одним из названных свойств. Аналогично, понятие $A_k \wedge A_i$ представляет новое свойство примера, которым этот пример обладает при условии, что он обладает обоими названными свойствами. Из семантики операций обобщения и специализации видно, что они полностью соответствуют операциям дизъюнкции и конъюнкции, соответственно, если понятие A_k трактовать как предикат, утверждающий «объект $B_j \in \boldsymbol{B}$ обладает свойством A_k для заданного $j \in \{1,...,m\}$», где $\boldsymbol{B} = \{B_j\}|_{j=1}^{m}$ есть множество объектов (примеров) обучающей выборки данных. Поэтому алгебраическая структура $\mathcal{R} = <\boldsymbol{A}, \{\vee, \wedge\}>$ по определению является решеткой.

Известно [125], что эта решетка порождает две полурешетки $\mathcal{R}^{(\vee)} = <\boldsymbol{A}, \{\vee\}>$ и $\mathcal{R}^{(\wedge)} = <\boldsymbol{A}, \{\wedge\}>$, которые называются верхней полурешеткой и нижней полурешеткой соответственно. Верхняя полурешетка, построенная таким способом, содержит элементы множества подмножеств $\boldsymbol{2^A}$ множества $\boldsymbol{A}$, соединенные символом $\vee$.

Поставим в соответствие каждому элементу $\boldsymbol{A}_i^{(\vee)} = \vee_{k\in\{1,...,n\}}(A_k)$ верхней полурешетки подмножество примеров $\boldsymbol{B}_i^{(\vee)}$, каждый из которых обладает хотя бы одним из свойств множества свойств $\boldsymbol{A}_i^{(\vee)}$. Назовем это множество примеров $\boldsymbol{B}_i^{(\vee)}$ (*по аналогии* с соответствующим термином для решетки формальных понятий) объемом узла $\boldsymbol{A}_i^{(\vee)}$ верхней полурешетки $\mathcal{R}^{(\wedge)}$, а каждому узлу ее поставим в соответствие пару $<\boldsymbol{A}_i^{(\vee)}, \boldsymbol{B}_i^{(\vee)}>$.

На некотором уровне верхней полурешетки в ней могут появиться узлы, объем которых совпадает со всем множеством примеров данных. Все эти элементы будут иметь в качестве объема понятия все множество примеров данных и

совпадают по объему с единичным элементом верхней полурешетки, который обозначим символом E (его называют единицей полурешетки).

Далее, поставим в соответствие каждому узлу $< \boldsymbol{A}_i^{(\vee)}, \boldsymbol{B}_i^{(\vee)} >$ верхней полурешетки (полурешетки обобщения) значение эмпирической вероятности, которое равно мощности множества, представляющего объем понятия, деленной на общее число примеров m обучающей выборки $\boldsymbol{B}$. Построенную таким образом структуру принято называть нормированной полурешеткой. Ее важным свойством является то, что она полностью задает дискретное распределение эмпирических вероятностей на множестве узлов $< \boldsymbol{A}_i^{(\vee)}, \boldsymbol{B}_i^{(\vee)} >$ этой полурешетки в выборке данных. Назовем условно эти узлы обобщенными понятиями верхней полурешетки, или, для краткости, просто обобщенными понятиями.

Аналогично, нижняя полурешетка $\mathcal{R}^{(\wedge)}$ содержит в себе элементы множества подмножеств $\boldsymbol{2}^{\boldsymbol{A}}$, соединенные символом $\wedge$. Минимальный элемент этой решетки приято называть ее нулем. Будем обозначать его символом 0.

Поставим в соответствие каждому элементу $\boldsymbol{A}_i^{(\wedge)} = \bigwedge_{k\in\{1,\ldots,n\}}(A_k)$ нижней полурешетки подмножество примеров $\boldsymbol{B}_i^{(\wedge)}$, каждый из которых обладает набором свойств $\boldsymbol{A}_i^{(\wedge)}$ и назовем, как это принято в АФП, это множество объемом узла $\boldsymbol{A}_i^{(\wedge)}$. Можно видеть, что если в этой решетке сохранить только те элементы, которые имеют непустой объем $\boldsymbol{B}_i^{(\wedge)}$, то нижняя полурешетка превращается в решетку формальных понятий, и каждый ее элемент $\{< \boldsymbol{A}_i^{(\wedge)}, \boldsymbol{B}_i^{(\wedge)} >\}|_{i\in\{1,\ldots,2^n\}}$ представляет одно из формальных понятий множества примеров $\boldsymbol{B}_i^{(\wedge)}$ обучающей выборки.

Аналогично верхней полурешетке, поставим в соответствие каждому узлу нижней полурешетки формальных понятий $< \boldsymbol{A}_i^{(\wedge)}, \boldsymbol{B}_i^{(\wedge)} \}>$ (полурешетки специализации) значение эмпирической вероятности, которое равно мощности множе-

ства, представляющего объем понятия, деленной на общее число примеров *m* обучающей выборки $\boldsymbol{B}$. Построенную таким образом структуру называют нормированной полурешеткой. Она полностью задает дискретное распределение эмпирических вероятностей на множестве узлов нижней полурешетки.

Можно видеть, что для каждого элемента $< \boldsymbol{A}_i^{(\wedge)}, \boldsymbol{B}_i^{(\wedge)} >$ нижней полурешетке в верхней полурешетке существует двойственный элемент $< \boldsymbol{A}_i^{(\wedge)}, \boldsymbol{B}_i^{(\wedge)} >$, названный обобщенным понятием, и эмпирические вероятности, поставленные в соответствие этим узлам, связаны формулой включений и исключений.

Другая особенность построенной пары полурешеток состоит в следующем. Верхняя полурешетка $\mathcal{R}^{(\vee)} = < \boldsymbol{A}, \{\vee\} >$ содержит в себе все потенциально возможные обобщения базовых понятий онтологии, но не только их. Важно отметить, что все базовые понятия онтологии по построению имеют семантическую интерпретацию. В отличие от этого, не все обобщенные понятия введенной верхней полурешетки имеют семантическую интерпретацию в терминах понятий онтологии. Но семантически интерпретируемые понятия онтологии могут быть выделены из множества обобщенных понятий с помощью специальных программных инструментов, поддерживающих автоматизированный процесс построения онтологий. Примером такого инструмента является уже неоднократно упоминавшийся ранее инструмент DBpedia, который можно использовать для построения иерархии понятий онтологии для заданного множества базовых понятий.

Очевидно, что в результате этой процедуры все множество узлов верхней полурешетки разделится на два подмножества, а именно, на множество узлов, которые имеют интерпретацию в терминах онтологии и все остальные, которые не являются семантически интерпретируемыми. Если последнее множество узлов удалить из верхней полурешетки (вместе в отношениями, которыми они связаны со своими предшественниками и последователями, то оставшееся множество узлов, структурированное в соответствии с естественным порядком, который индуцируется операцией обобщения $\vee$, может рассматриваться в качестве стартовой версии иерархии понятий онтологии.

Отметим два важных условия, которые используются далее при обосновании алгоритма построения онтологии.

1. В рассматриваемом контексте строится онтология данных. Поэтому два понятия, которым отвечают одни и те же объемы, дублируют друг друга, и в итоговую онтологию следует включать только одно из них. Такие понятия в решетке обобщения могут встречаться на одном уровне иерархии, но, как показывает практика, чаще они соответствуют двум смежным уровням иерархии. По сути они отвечают одному и тому же понятию, названному разными именами и могут различаться лишь формально. В онтологии необходимо оставлять только одно из таких понятий.

2. Онтология данных строится для того, чтобы на ее основе далее синтезировать систему принятия решений. Поэтому с этой точки зрения в ней интерес представляют только те понятия, которые связаны между собой статистической связью. Поэтому, если каким-то образом выяснено, что два или больше понятия не содержат в своем объеме попарно общих примеров из множества данных $\boldsymbol{B}$, то в имеющейся обучающей выборке они независимы. Это означает, что между ними отсутствует выборочная статистическая связь, а потому любые понятия, которые являются их обобщением, могут не включаться в вычисляемую онтологию, поскольку они бесполезны для построения решающих правил. Достаточно использовать их предшественников в онтологии. Установить независимость пары понятий онтологии можно с помощью анализа объема двойственного понятия в решетке формальных понятий. Действительно, если некоторое формальное понятие имеет нулевой объем, то это значит, что его непосредственные предшественники в решетке формальных понятий независимы. Само формальное понятие $\boldsymbol{A}_i^{(\wedge)}=(\bigwedge_{k\in\{1,\ldots,n\}} A_k)$ с нулевым объемом в решетку формальных понятий не входит. Не входят в нее и все остальные формальные понятия, которые сравнимы с $\boldsymbol{A}_i^{(\wedge)}$ и строго меньше него по естественному порядку нижней полурешетки $\mathcal{R}^{(\wedge)}$. Таким образом, второе правило управления процессом построения онтологии дан-

ных для принятия решений состоит в том, что если формальное понятие, двойственное по отношению к рассматриваемому понятию онтологии, имеет нулевой объем, то это понятие и все строго большие понятия верхней полурешетки в онтологию не включаются.

Разработанная методика автоматического построения онтологии данных базируется на этих двух правилах, которые кратко формулируются следующим образом:

Правило 1. Из любой пары понятий $<\boldsymbol{A}_i^{(\vee)}, \boldsymbol{B}_i^{(\vee)}>$ и $<\boldsymbol{A}_j^{(\vee)}, \boldsymbol{B}_j^{(\vee)}>$ верхней полурешетки, для которых $\boldsymbol{B}_i^{(\vee)} = \boldsymbol{B}_j^{(\vee)}$, в онтологию включается только одно из них, предпочтительно меньшее по естественному порядку верхней полурешетки.

Правило 2. Если некоторое обобщенное понятие $<\boldsymbol{A}_i^{(\vee)}, \boldsymbol{B}_i^{(\vee)}>$ верхней полурешетки таково, что двойственный ему элемент $<\boldsymbol{A}_i^{(\wedge)}, \boldsymbol{B}_i^{(\wedge)}\}>$ полурешетки формальных понятий имеет пустой объем ($\boldsymbol{B}_i^{(\wedge)}=\{\emptyset\}$), то понятие $<\boldsymbol{A}_i^{(\vee)}, \boldsymbol{B}_i^{(\vee)}>$ и все равные ему или большие него по естественному порядку в онтологию не включаются.

Таким образом, алгоритм автоматического вычисления онтологии интересов пользователя должен строиться с учетом следующих положений:

1. Стартовый уровень процедуры обобщения понятий онтологии – это уровень базовых понятий онтологии (уровень k=1).

2. Стартовый уровень специализации формальных понятий - тот же самый, т.е. при k=1 все понятия онтологии базового уровня являются также и формальными понятиями уровня 1 (они семантически интерпретируемы в онтологии по построению).

3. Процедура обобщения понятий онтологии текущего уровня строит понятия онтологии для непосредственного следующего уровня иерархии, т.е. для понятий уровня k строит их обобщение на уровне k+1. Процедура может быть реализована с использованием баз данных понятий и инструментов их обобщения, например, с помощью баз данных и инструментов DBpedia.

4. Если новые понятия онтологии инструментом DBpedia не генерируются на уровне k+1, то процесс ее построения, как и процесс построения частично упорядоченного множества формальных понятий, останавливаются.

5. Если множество таких понятий не пусто, то для каждого нового потенциального понятия онтологии уровня $k+1$ (1) вычисляется объем, (2) строятся двойственные им элементы полурешетки формальных понятий и вычисляются их объемы. Далее используются правила, сформулированные в утверждениях 1 и 2 для выяснения, какие из потенциальных кандидатов верхней полурешетки уровня $k+1$ включаются в онтологию.

6. Повторяется шаг 4.

Нужно иметь в виду, что для одних и тех же данных $\boldsymbol{B}$ могут быть построены различные онтологии. Причины для этого могут быть различными. Во-первых, очевидно, что другие инструменты, которые могут быть использованы вместо инструмента DBpedia, могут привести к разным множествам понятий онтологии базового уровня и онтологии в целом. И это нормально в связи с особенностями естественного языка, который всегда далек от однозначности понятий и терминов. Однако из-за этого не должно возникать проблем при практическом их использовании в интересах систем принятия решений.

Во-вторых, онтология данных, построенная описанным способом, как показала практика, всегда избыточна, и одна и та же важная (для работы рекомендательной системы) причинно-следственная связь, реально присутствующая в данных, может отражаться в связях различных понятий онтологии. Практика показала, например, что если рассмотреть структурированное подмножество понятий онтологии, которые включает в себя максимальное понятие (в смысле, который вкладывается в термин «максимальный элемент» в теории частично упорядоченных множеств) и все сравнимые с ним по порядку элементы онтологии (все они строго меньше максимального элемента), то претендовать на включение в множество интересов пользователя для формирования правил принятия решений, зачастую, с равным правом могут несколько понятий описанного множества. При

этом наилучшим из них далеко не всегда оказывается максимальный элемент, соответствующий наибольшему обобщению некоторого множества базовых понятий. Заметим, что этот экспериментально установленный факт представляет большую практическую ценность, поскольку он указывает путь сокращения размерности модели целевой переменной практически без потери информации. Это одно из важнейших достоинств (наряду с другими) разработанной модели семантического анализа понятий применительно к задачам принятия решений.

В-третьих, множество базовых понятий тоже избыточно, поэтому «потерять» в итоговой онтологии все множество понятий, которые являются его обобщением также не является катастрофой.

Далее, в любом случае онтология интересов пользователя, реально используемая рекомендательной системой, «отстает» от актуальной онтологии, поскольку интересы пользователя эволюционируют, и эта эволюция находит отражение в динамике эволюции онтологии с некоторой задержкой. Это означает, что рекомендательная система всегда будет работать с использованием приближенной онтологии.

Однако самый убедительный аргумент в пользу семантического анализа понятий состоит в том, что практическое использование алгоритма автоматического построения онтологии, следующего положениям, описанными выше, показало хорошие результаты как в отношении качества работы исследовательских прототипов рекомендательных систем, построенных для разных множеств данных, так и в части вычислительной эффективности их работы. Соответствующие численные результаты приводятся в последующих разделах данной главы.

3.2.5 Алгоритм автоматического построения онтологии данных на основе семантического анализа понятий

На рисунке 3.1 приведен псевдокод алгоритма автоматического построения онтологии на основе модели семантического анализа понятий. Приведем краткие пояснения некоторых шагов алгоритма, которые не были описаны до этого.

SemanticConceptAnalisys (*B*) :

Вход: выборка данных $\boldsymbol{B} = \{ B_j \}|_{j=1}^{m}$

Выход: двойственная структура $\mathfrak{R}$

Начало

k =1;

$\boldsymbol{A}_k = \varnothing$; *// множество понятий первого уровня*

для всех экземпляров $B_j \in \boldsymbol{B}$:

$\boldsymbol{A}_k = \boldsymbol{A}_k + DBpediaSpotlightService\ (B_j)$; *// извлечь базовые понятия из текста*

для всех понятий $A_i^k \in \boldsymbol{A}_k$:

вычислить $\boldsymbol{B}_i^k$ и $\left|\boldsymbol{B}_i^k\right|$ */ множество экземпляров* выборки, *в которых встречается понятие* A_i^k *и его мощность*

$\hat{\boldsymbol{B}}_i^k = \boldsymbol{B}_i^k$; *// подмножества экземпляров базовых понятий и мощности «объёмов» двойственных им формальных понятий равны на 1 уровне*

Filter($\boldsymbol{A}_k$); *// отфильтровать базовые понятия*

пока $\boldsymbol{A}_k^{(\vee)} \neq \varnothing$: *// критерий останова разработки структуры* $\mathfrak{R}$; $\boldsymbol{A}_k^{(\vee)}$ *- множество понятий онтологии на уровне k*

для всех понятий $A_i^{(\vee),k} \in \boldsymbol{A}_k^{(\vee)}$

$\boldsymbol{A}_{k+1}^{(\vee)} = \boldsymbol{A}_{k+1}^{(\vee)} + DBpedia(\ A_i^{(\vee),k}\ )$; *// добавить обобщенные понятия онтологии*

для всех понятий $A_i^{(\vee),k+1} \in \boldsymbol{A}_{k+1}^{(\vee)}$:

вычислить $\boldsymbol{B}_i^{(\vee),k+1}$ и $\left|\boldsymbol{B}_i^{(\vee),k+1}\right|$;

построить $A_i^{(\wedge),k+1}$; *// двойственные формальные понятия*

вычислить $\boldsymbol{B}_i^{(\wedge),k+1}$ и $\left|\boldsymbol{B}_i^{(\wedge),k+1}\right|$; *// «объем» и его мощность для двойственных понятий*

если $F_1\ (A_i^{(\vee),k+1}) ==$ *ложь* **или** $F_2\ (A_i^{(\vee),k+1}) ==$ *ложь* */ критерии фильтрации*

то удалить понятия $A_i^{(\vee),k+1}$ из $\boldsymbol{A}_{k+1}^{(\vee)}$, и $A_i^{(\wedge),k+1}$ из $\boldsymbol{A}_{k+1}^{(\wedge)}$

добавить $\boldsymbol{A}_{k+1}^{(\vee)}$ и $\boldsymbol{A}_{k+1}^{(\wedge)}$ в $\mathfrak{R}$;

k = k+1;

вернуть $\mathfrak{R}$;

Конец.

Рисунок 3.1 – Псевдокод алгоритма, реализующего семантический анализ понятий

При условии, что выборка $B_j = \{ B_j \}|_{j=1}^{m}$ содержит неструктурированные данные, представленные текстами B_j на естественном языке, нулевой шаг разработанного алгоритма – это предварительная обработка текстов с целью извлечения базовых понятий. В работе в качестве NLP-инструмента использован сервис DBpedia Spotlight [109]. Такая предобработка данных не требует вмешательства эксперта и реализована в процедуре *DBpediaSpotlightService* (B_j).

Для каждого экземпляра текста выборки сервис возвращает набор URI-статей Википедии, которые соответствуют понятиям, найденным сервисом в этом тексте. Объединение понятий онтологии, извлеченных из всех текстов выборки данных $\boldsymbol{B}=\{B_j\}|_{j=1}^{m}$ формирует множество понятий онтологии базового уровня $\boldsymbol{A} = \{A_i\}|_{i=1}^{n}$. Подмножество $\boldsymbol{B}_{\mathrm{i}}^{1} = \{B_j\}|_{j=1}^{n_i}$ экземпляров выборки $\boldsymbol{B}$, содержащее понятие A_i, может быть поставлено в соответствие каждому понятию $A_i \in \boldsymbol{A}$. Мощности подмножеств $\left|\boldsymbol{B}_{\boldsymbol{i}}^{\boldsymbol{1}}\right|$, которые при делении на общее число примеров *m* обучающей выборки $\boldsymbol{B}$ будут равны значению эмпирической вероятности появления этого понятия в выборке, также отображаются на каждое базовое понятие A_i.

Напомним, что базовые понятия являются также и формальными понятиями уровня 1. После того, как построен уровень базовых понятий онтологии, эти понятия могут быть отфильтрованы с помощью выбранной метрики «значимости» понятия. Как правило, конкретное приложение определяет оптимальную стратегию фильтрации. Тем не менее, в ходе исследования было изучено несколько метрик для фильтрации. Все они учитывают значение эмпирической вероятности появления базовых понятиях в выборке в качестве наиболее важного атрибута. Процедура фильтрации базовых понятий не является обязательным шагом, однако в ходе экспериментального исследования он был выполнен. В псевдокоде эта процедура названа *Filter(* $\boldsymbol{A}_{\boldsymbol{k}}$ *)*.

Отметим, что простейшая метрика для фильтрации базовых понятий представляет собой само значение эмпирической вероятности для базового понятия с выбранным порогом фильтрации. Действительно, практика показала, что множество базовых понятий часто содержит избыточные понятия, и, по меньшей мере, 20% понятий, имеющих наименьшее значение этой метрики, могут быть отфильтрованы. Тем не менее, такой подход требует последующей дополнительной проверки, не остались ли экземпляры выборки, которым не соответствует ни одно базовое понятия в результирующем множестве.

Другой вариант фильтрации состоит в следующем. Во-первых, все основные понятия могут быть отсортированы в соответствии с их значением оценки эмпирической вероятности в порядке убывания. Затем понятия с наиболее высокими оценками, удовлетворяющими пороговому критерию, добавляются к результирующему множеству базовых понятий. Выбор порога зависит от набора данных и решаемой задачи и является предметом предварительной вычислительной настройки. После этого для каждого экземпляра текста выбирается одно (или более, если это необходимо) понятие, которое наиболее часто встречается в данном тексте. Это понятие(-я) включается в конечные множество понятий, если оно не было включено в него ранее. Именно такая стратегия фильтрации применена для формирования примеров онтологий данных в ходе эксперимента (раздел 3.4).

Последующие операторы псевдокода описывают шаги 3-6 алгоритма, содержательно описанные в предыдущем разделе работы. При этом процедура *DBpedia*($_{j=1}^{m}$) реализует обобщение понятий онтологии, построенных на предыдущем шаге, с использованием DBpedia и иерархии категорий Википедии. Как было упомянуто выше, каждое базовое понятие соответствует URI-статье в Википедии. Это означает, что для каждого понятия, можно получить категории Википедии, к которым принадлежит данная статья. URI этих категорий и их родительских категорий формируют верхние уровни разрабатываемой онтологии.

Критерий фильтрации $F_1(A_i^{(\vee),k})$, представленный в псевдокоде, реализует *Правило 1* из предыдущего раздела работы. Он может быть формально представлен следующим образом:

$$F_1(A_i^{(\vee),k}) = \begin{cases} \textit{ложь, если родительское понятие имеет одного потомка,} \\ \textit{истина, в остальных случаях.} \end{cases} \quad (3.1)$$

Аналогично, критерий фильтрации $F_2(A_i^{(\vee),k})$ соответствует *Правилу 2*:

$$F_2(A_i^{(\vee),k}) = \begin{cases} \textit{ложь, если } \left|\boldsymbol{B}_i^{(\wedge),k}\right| = 0, \\ \textit{истина} - \textit{в остальных случаях.} \end{cases} \quad (3.2)$$

где $\left|\boldsymbol{B}_i^{(\wedge),k}\right|$ - это мощность объема формального понятия $A_i^{(\wedge),k}$, двойственного понятию онтологии $A_i^{(\vee),k}$.

Критерий останова построения онтологии можно сформулировать так: остановить процесс формирования онтологии, если на текущем уровне с помощью двух правил фильтрации были отфильтрованы все сгенерированные понятия.

Описанный в псевдокоде алгоритм автоматического построения онтологии данных на основе модели семантического анализа понятий возвращает структуру $\mathcal{R}$, состоящую из двух полурешеток, одна из которых соответствует иерархической структуре понятий, другая – структуре формальных понятий. При этом каждому узлу обеих полурешёток приписаны значения мощности подмножеств экземпляров выборки, которые соответствуют этому узлу. Эти значения с помощью деления их на общее число примеров *m* обучающей выборки позволяют вычислить значение эмпирической вероятности появления понятия, соответствующего узлу, в выборке. Такая комбинированная структура названа семантической структурой понятий и может быть легко использована в системе принятия решений.

Схема программного обеспечения, реализующего САП, приведена на рисунке 3.2.

3.3 Выбор набора экспериментальных данных для тестирования разработанных алгоритмов

Для демонстрации и экспериментального исследования качества и вычислительной эффективности алгоритма семантического анализа понятий, а в дальнейшем и других разработанных алгоритмов, выбран набор данных из области рекомендательных систем набор данных Amazon [126].

Данные набора были собраны с сайта Amazon летом 2006 года и содержат информацию о 548552 различных продуктах из разделов Книги, Музыка, DVD, VHS видео и отзывы пользователей о них.

Для каждого продукта доступна следующая информация:

– название;

– показатель продаж;

– список похожих продуктов- продуктов, которые покупались вместе с ним;

– детальная категоризация продукта;

– отзывы: время, покупатель, рейтинг.

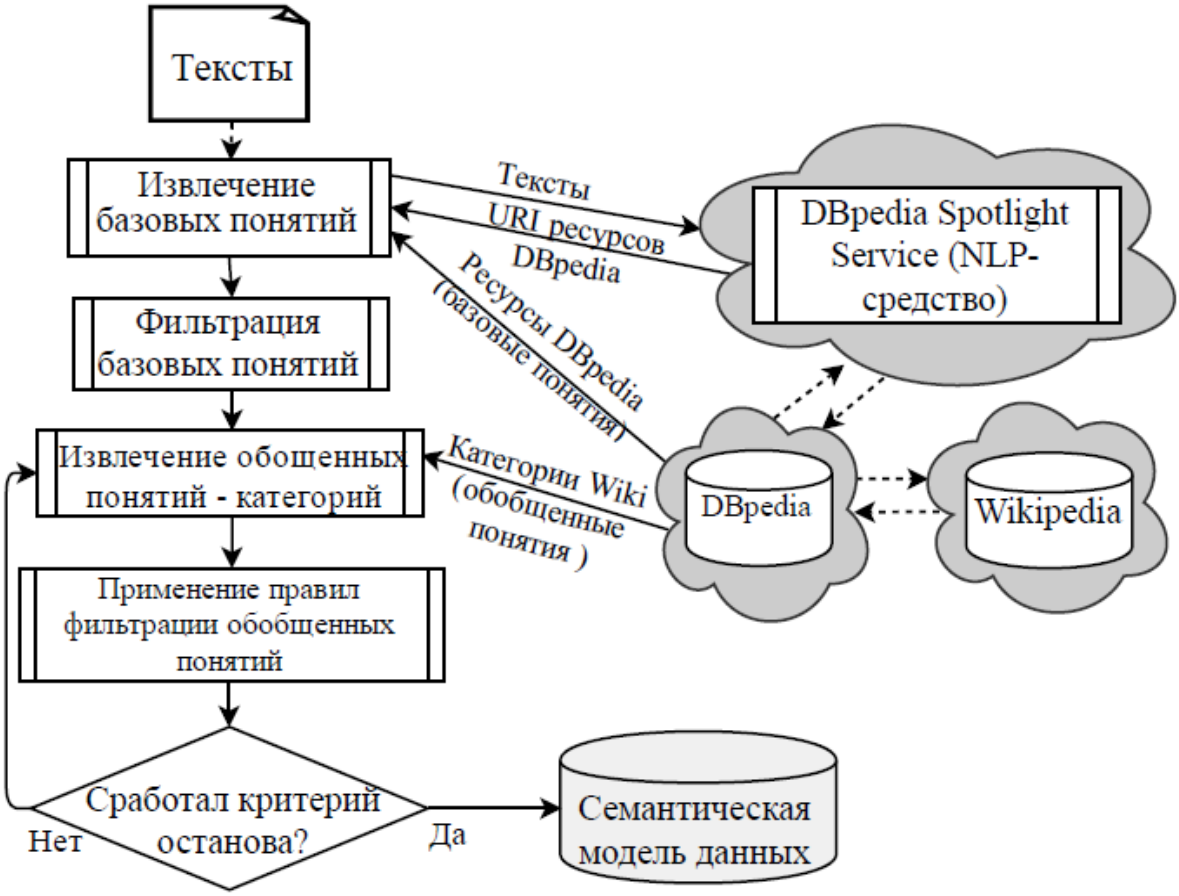


Рисунок 3.2 – Семантический анализ понятий

Мета-модель этого набора данных Amazon в виде диаграммы классов представлена на рисунке 3.3.

Основная сущность набора - «Продукт» (англ. *Product*) - определяется полями *id* – идентификационный номер, *ASIN* - стандартный идентификационный номер Amazon, *title* - название, *group* – группа продуктов, к которой он принадлежит, (Книги, DVD, VHS видео или Музыка), *salesrank* - показатель продаж продукции, *similar* - множество *ASIN* других продуктов, которые покупали вместе с этим продуктом, *categories* – категории, к которым относится продукт. Сущность «Категория» (англ. *Category*) определяется полями *category* – название категории и *parent* – ссылка на родительскую категорию. Продукт также связан с набором отзывов – *reviews*. Атрибутами сущности «Отзыв» (*англ*. *Review*) являются *time* –

время, *customer* – пользователь, оставивший отзыв, *rating* – выставленная им оценка. Последний атрибут сущности «Продукт» - *avgrating* - среднее значение рейтинга продукта, указанное пользователями, в своих отзывах. Сущность *Пользователь* (англ. *Customer*) имеет только атрибут *id* – идентификационный номер.

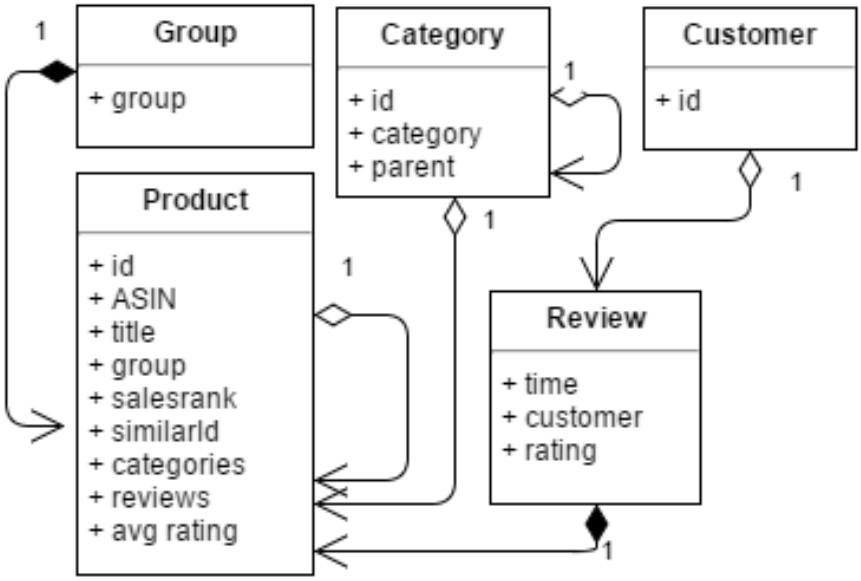


Рисунок 3.3 – Мета-модель набора данных Amazon

На рисунке 3.4 приведен пример экземпляра набора данных Amazon, представляющего собой продукт из группы *Книги* с указанием его категорий и сходных продуктов. Категории продукта приведены в виде иерархии последовательных узлов, разделенных символом «|». *Id* - номер категории, который указывается в квадратных скобках ([*]).

В таблице 3.1 приведены некоторые количественные характеристики набора данных Amazon.

Опишем некоторые ключевые особенности данного набора, повлиявшие на его выбор в качестве экспериментального. Во-первых, набор содержит данные о большом количестве продуктов из различных предметных областей, следовательно, на нем можно хорошо продемонстрировать возможности семантического анализа понятий при построении онтологии данных, объединяющей знания из различных областей в рамках одной семантической модели. Во-вторых, информация о каждом продукте может быть обогащена с помощью различных источников

и облачных ресурсов, таких как Википедия, DBpedia и др. Это дает основание говорить об этом наборе как о наборе больших данных, так как возможности такого обогащения практически безграничны. Напомним, что описываемый набор содержит отзывы пользователей на купленные ими продукты. Такая информация очень хорошо подходит для извлечения предпочтения и интересов пользователя с использованием онтологии, построенной с помощью семантического анализа понятий. Отметим, что наличие оценок по пятибалльной шкале, выставленных пользователями купленным продуктам, делает возможным дальнейшее использование набора данных не только для построения профиля пользователя, но для обучения моделей принятия решения, основанных на ассоциативно-причинной классификации, которые позволят давать пользователю мотивированные рекомендации относительно новых продуктов.

Таблица 3.1 – Количественные характеристики набора данных Amazon

Название	**Количество**
Продукты в категории Books	393561
Продукты в категории DVDs	19828
Продукты в категории Music CDs	103144
Продукты в категории Videos	26132
Всего продуктов	548552
Отношений продуктов к категориям	2509699
Всего пользователей	1555124
Пользователей с оценками {1,2,3,4,5}	14356
Всего отзывов	7781990

3.4 Экспериментальное исследование автоматического формирования онтологии данных на основе методов семантического анализа понятий

Рассмотрим несколько примеров формирования онтологии интересов пользователя с помощью семантического анализа понятий на основе данных Amazon.

На этапе предобработки данные набора Amazon были трансформированы в реляционную структуру. Продукты были разнесены в различные таблицы в зависимости от группы, к которой они принадлежат. На рисунке 4.5 приведена диаграмма сущность-связь для реляционной структуры, построенной для набора данных Amazon.

```
Id:   15
ASIN: 1559362022
title: Wake Up and Smell the Coffee
group: Book
similar: 5 1559360968 1559361247 1559360828 1559361018
         0743214552
categories: 3
   |Books[283155]|Subjects[1000]|Literature & Fiction [17]
   |Drama[2159]|United States[2160]
   |Books[283155]|Subjects[1000]|Arts& Photography[1]|
   Performing Arts [521000]|Theater[2154]|General[2218]
   |Books[283155]|Subjects[1000]|Literature & Fiction[17]|
   Authors, A-Z[70021]|( B )[70023]|Bogosian, Eric[70116]
```

Рисунок 3.4 - Пример экземпляра набора данных Amazon

Для того чтобы построить онтологию интересов пользователя, необходимо выбрать отзывы пользователя на различные продукты. В данном эксперименте были использованы все отзывы пользователя, независимо от того положительные они или отрицательные.

На рисунке 3.5 представлено процентное распределение пользователей со всеми оценками из множества {1, 2, 3, 4, 5} по количеству отзывов у каждого пользователя.

На гистограмме (рисунок 3.6) видно, что большинство пользователей имеет количество отзывов в диапазоне от 15 до 50.

Для эксперимента случайным образом были выбраны 3 пользователя с идентификаторами ANIHXYXKPZL2H, A3DMZKTBIJURE1, и A3IMNZSYDOTTU6 из набора данных Amazon с количеством отзывов в диапазоне от 15 до 100. Этому диапазону принадлежит 69% пользователей, представленных в наборе данных Amazon, которые имеют все отзывы со всеми оценками из множества {1, 2…, 5}.

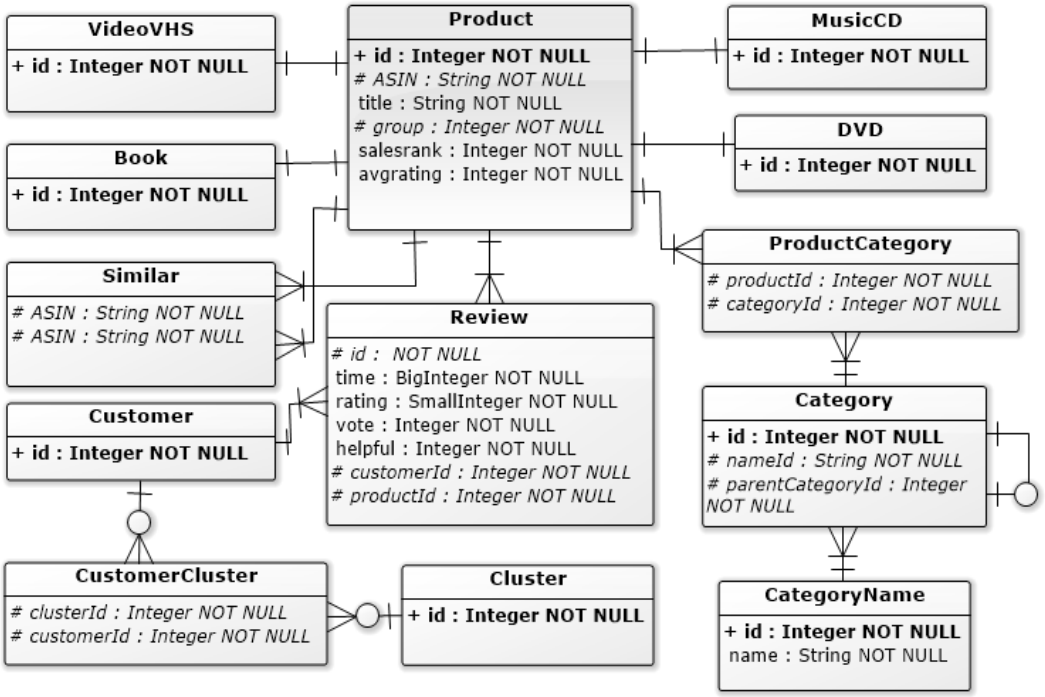


Рисунок 3.5 - Диаграмма сущность-связь реляционной структуры данных Amazon

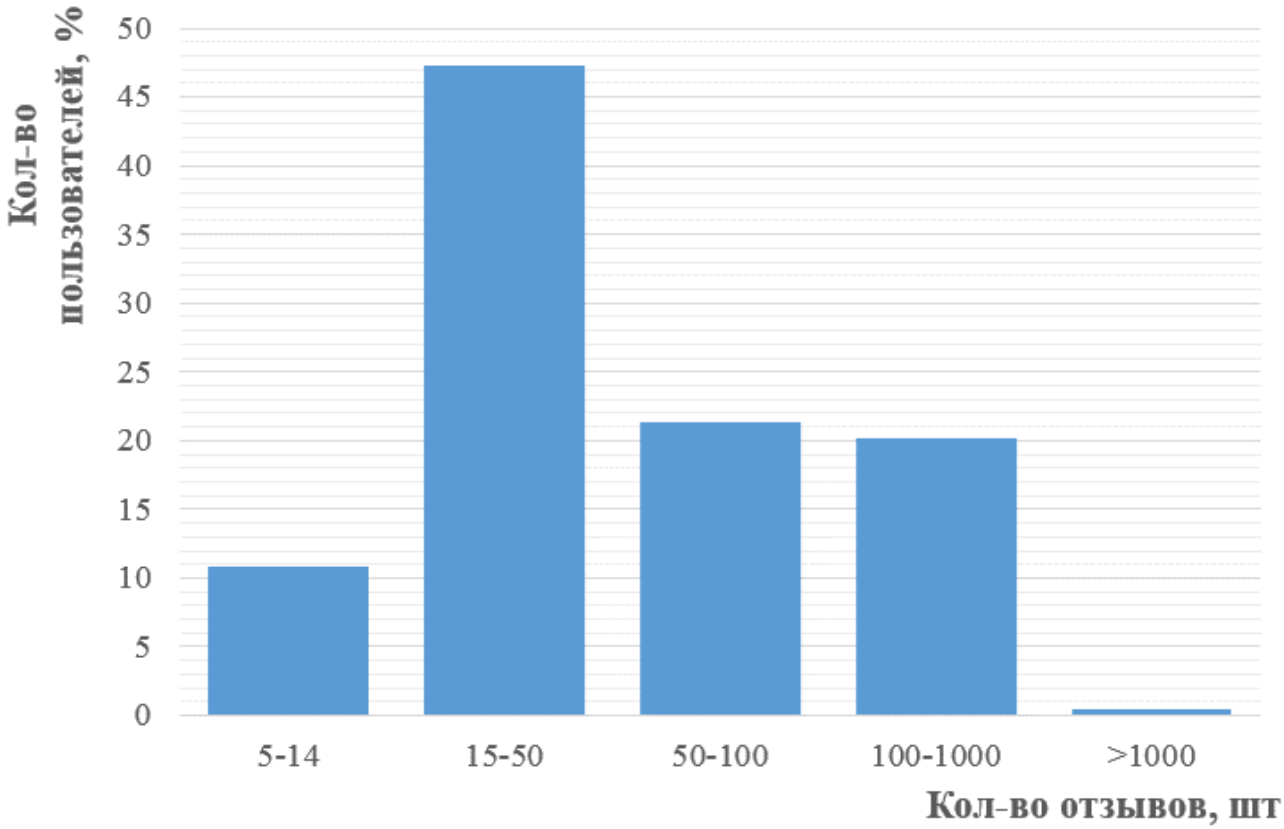


Рисунок 3.6 – Процентное распределение пользователей по количеству отзывов

Для каждого пользователя было сформировано множество продуктов, на которые он оставлял отзыв. Каждое такое множество рассматривается по отдельности и используется для построения онтологии интересов соответствующего пользователя. Списки продуктов, на которые пользователи оставляли отзывы, могут быть найдены в приложении Б (таблица Б.1 – Б.3).

С помощью сервиса DBpedia по имени каждого продукта можно получить краткую аннотацию соответствующей ему статьи в Википедии. Множество таких аннотаций далее используется в качестве выборки данных, подаваемой на вход алгоритма САП, описанного в подразделе 3.2.5, для построения онтологии интересов каждого пользователя.

На рисунке 3.7 представлена схема использования DBpedia и САП для построения онтологии каждого пользователя в отдельности.

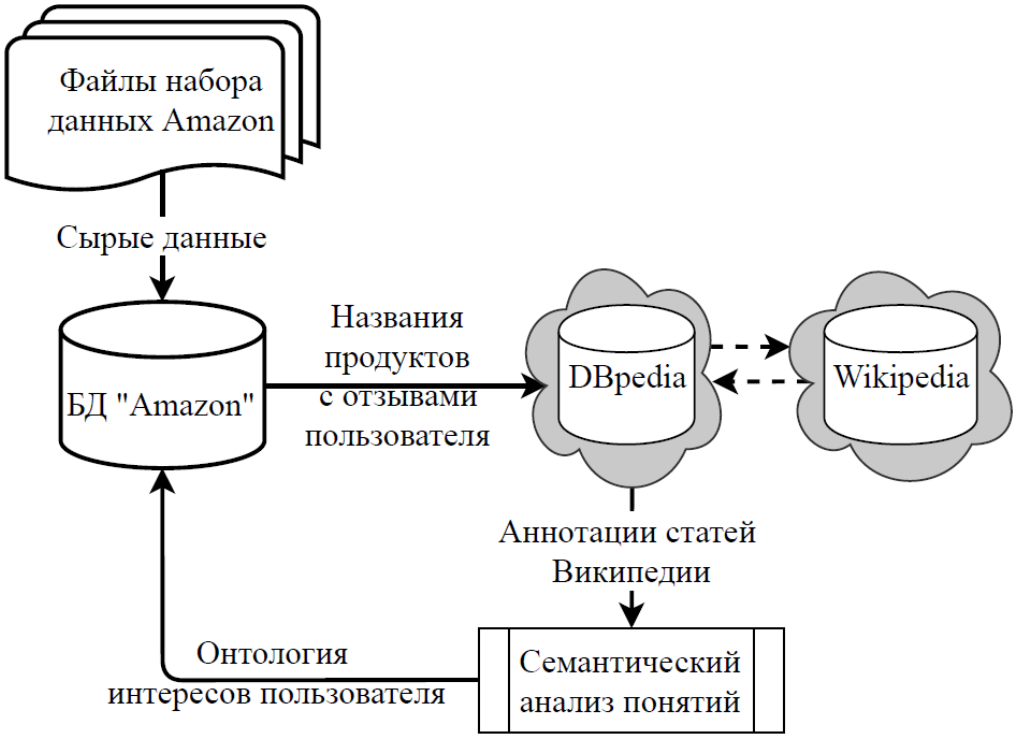


Рисунок 3.7 - Схема построения онтологии пользователя набора данных Amazon

Для каждого пользователя выполнялось два эксперимента. Численные характеристики этих экспериментов представлены в приложении Б (таблица Б.4). В первом случае построение семантической структуры понятий, представляющей

онтологию интересов пользователя, выполнялось без применения правил фильтрации на верхних уровнях. Целью этого эксперимента было показать максимальное количество понятий, которые могут быть извлечены из глобальной онтологии для конкретного пользователя. Отметим, что без использования правил фильтрации количество уровней онтологии может достигать нескольких десятков. При этом большинство полученных понятий бесполезны с точки зрения принятия решений, так как двойственные им формальные понятия имеют пустой объем. Во втором случае при построении онтологии интересов пользователей были использованы оба правила фильтрации.

Сравнение результатов двух серий экспериментов показывает, насколько важную роль предложенные правила фильтрации играют в сокращении бесполезных, с точки зрения выборки данных, понятий и повышении вычислительной эффективности построения онтологии интересов пользователя с помощью процедуры семантического анализа понятий.

На рисунках 3.8 – 3.10 представлены фрагменты семантических структур понятий, представляющих онтологии интересов выбранных пользователей «ANIHXYXKPZL2H», «A3DMZKTBIJURE1», «A3IMNZSYDOTTU6», полученные на основе их отзывов о продуктах набора данных Amazon с помощью семантического анализа понятий. В малых овалах для каждого понятия через косую черту указана мощность подмножества экземпляров данных и мощность «объёма» двойственного понятия. Информация обо всех понятиях онтологий интересов, построенных для пользователей, может быть найдена по адресу [128].

Следует отметить, что построение онтологии данных с помощью САП – это алгоритм обучения без учителя. Существует несколько способов оценить точность и эффективность алгоритмов обучения без учителя [127]. Для алгоритмов кластеризации, например, можно использовать некие «внутренние» метрики оценки полученных кластеров. Такие метрики основаны на вычислении размеров кластеров и расстояния между ними. Очевидно, что такой подход неприменим к оценке метода семантического анализа понятий. Вторая группа методов оценки

алгоритмов без учителя носит название «внешние» методы [127]. Для использования этих методов необходимо наличие размеченных данных (некого «золотого стандарта»), которые будут использованы в ходе оценки алгоритмов обучения без учителя. Однако, такие данные очень редко существуют в прикладных задачах, и соответственно, «внешние» методы, по сути, неприменимы в условиях реальных данных. В случае оценки результатов семантического анализа понятий применить такие методы также не представляется возможным.

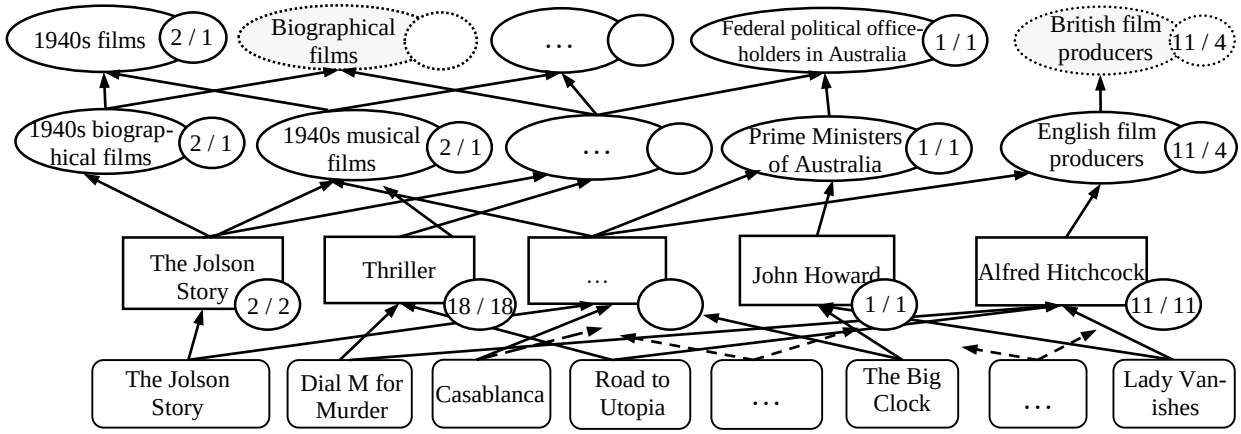


Рисунок 3.8 – Фрагмент семантической структуры понятий, представляющей онтологию интересов пользователя «ANIHXYXKPZL2H»

Таким образом, фактически, остается только один возможный вариант оценки результатов семантического анализа понятий – это использование полученных результатов в алгоритмах обучения с учителем и изучение влияния этих результатов на точность таких алгоритмов. Применение именно такого подхода будет описано в главе 4.

3.5 Выводы: рекомендации по использованию семантического анализа понятий

В данной главе решены следующие задачи работы:

1. Разработан масштабируемый механизм автоматического построения онтологии для семантического моделирования данных, используемых в задаче принятия решений.

2. Предложена выразительная модель обрабатываемых данных, которая позволяет совместно представить синтаксис и семантику данных.

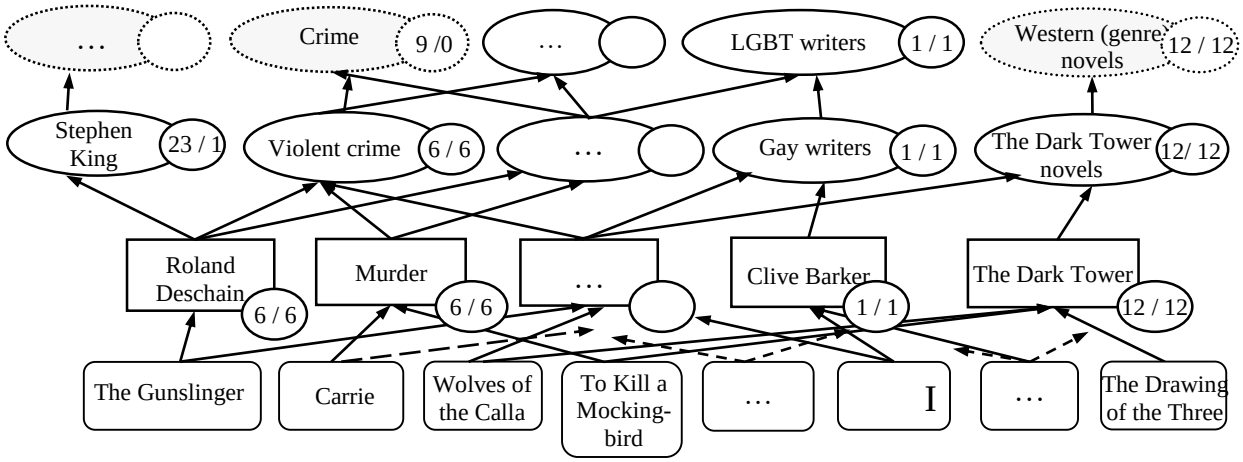


Рисунок 3.9 – Фрагмент семантической структуры понятий, представляющей онтологию интересов для пользователя «A3DMZKTBIJURE1»

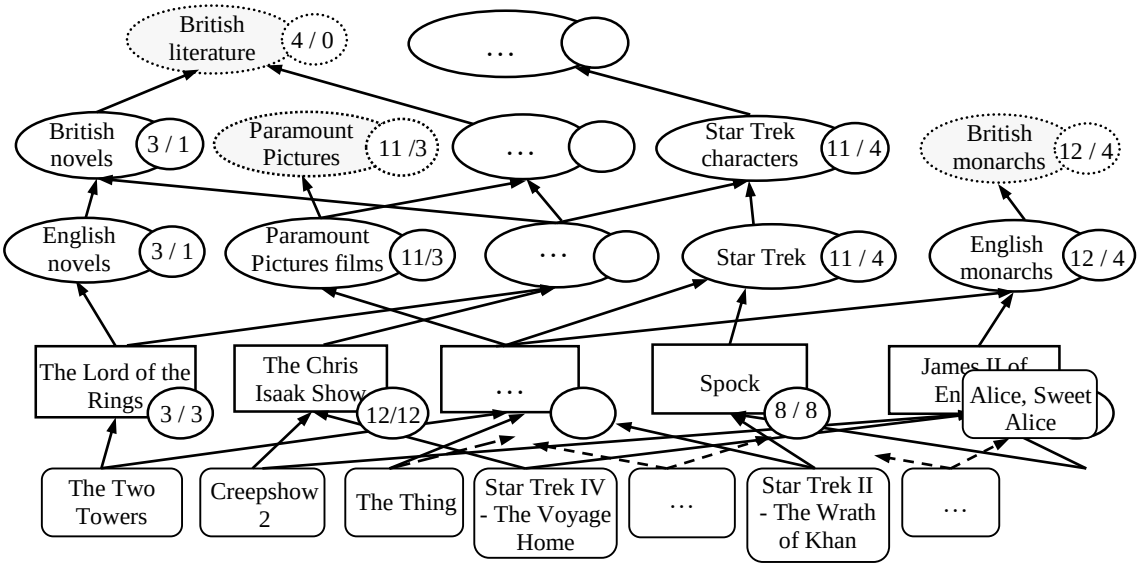


Рисунок 3.10 – Фрагмент семантической структуры понятий, представляющей онтологию интересов для пользователя «A3IMNZSYDOTTU6»

Новая методика автоматизированного построения онтологии данных, которая предложена в данной главе, назван семантическим анализом понятий. Основная идея семантического анализа понятий – это объединение преимуществ онтологической спецификации семантики понятий предметной области, извлеченных с помощью средств DBpedia, и управление процессом генерации этих понятий с помощью формальных понятий, двойственных к ним.

Структура данных, которая может быть получена в результате семантического анализа понятий, состоит из двух полурешеток, сформированных от данных и двойственных друг другу. Одна из этих полурешёток соответствует иерархической структуре понятий онтологии, другая – структуре формальных понятий. Такая комбинированная структура названа семантической структурой понятий и может быть интерпретирована с двух точек зрения:

– с онтологической точки зрения, семантическая структура понятий – это онтология данных, обогащенная иерархией формальных понятий, которые формируют структуру, которая относится к компоненте принятия решения (англ. *actionable component*);

– с точки зрения АФП, семантическая структура понятий – это структура формальных понятия, построенная от данных, содержащая только понятия, обладающие семантикой, наличие которой в классическом АФП должен проверять эксперт.

Предложенная выразительная модель данных не только позволяет представить совместно синтаксис и семантику данных, но также содержит в себе метаинформацию (оценки эмпирических вероятностей появления понятий), которую далее можно эффективно использовать при разработке алгоритмов принятия решений.

Работоспособность алгоритма семантического анализа понятий была экспериментально продемонстрирована на нескольких примерах из набора данных Amazon. В ходе экспериментов выполнялось построение онтологий интересов нескольких пользователей, информация о которых была извлечена из набора данных Amazon.

Анализ результатов экспериментального исследования показал, что помимо отсечения бесполезных, с точки зрения выборки данных, понятий онтологии, семантический анализ понятий позволяет уменьшить время построения онтологии интересов пользователя на несколько порядков по сравнению с простым извлечением понятий будущей онтологии из глобальной базы знаний DBpedia.

Онтология интересов пользователя, построенная с помощью семантического анализа понятий, может далее быть использована для обучения его профиля. Соответствующая технология рассматривается в следующей главе. Такой профиль позволяет вырабатывать рекомендаций относительно новых для пользователя продуктов. Алгоритмы выработки рекомендаций, в отличие от алгоритма семантического анализа понятий, являются алгоритмами обучения с учителем. При использования онтологий данных, построенных методом семантического анализа понятий, в алгоритмах выработки рекомендаций можно будет экспериментально оценить эффективность, и в некотором смысле, точность семантического анализа понятий. Применение именно такого подхода будет описано в главе 4 данной работы.

4 АЛГОРИТМЫ ПОСТРОЕНИЯ И ОПТИМИЗАЦИИ АССОЦИАТИВНО-ПРИЧИННЫХ МОДЕЛЕЙ ДЛЯ ПРИНЯТИЯ РЕШЕНИЙ (НА ПРИМЕРЕ ОБУЧЕНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ)

4.1 Особенности семантических моделей рекомендательных систем третьего поколения

В настоящее время особый научный интерес для исследователей представляют рекомендательные системы третьего поколения [30]. Напомним, что рекомендательные системы третьего поколения рассматриваются как очередной шаг в сторону лучшего понимания интересов пользователя и выработки более адекватных и разносторонних рекомендаций. Они ориентируются на принятие решений на основе семантических моделей интересов и предпочтений пользователя, на более глубокий анализ и понимание мотивации и причин, которые побуждают его выбирать тот или иной товар/услугу. Поэтому разработка современной рекомендательной системы третьего поколения, ориентированной на пользователя, должна начинаться с построения (обучения) его профиля, который будет эти интересы отражать.

Рассмотрим подробнее данные, которые могут быть использованы для обучения профиля пользователя в рекомендательной системе третьего поколения. Чаще всего такие данные содержат информацию о товарах/услугах, которые может выбрать пользователь или выбирал их прошлом (будем их далее называть продуктами), и при этом предполагается, что каждый прошлый выбор пользователя имеет его оценку в виде рейтинга соответствующего продукта.

Обычно рейтинг – это величина, заданная в целочисленной балльной шкале, и чем выше его значение, тем более высоко продукт оценен пользователем. Часто для рейтинга выбирают область значений от 1 до 5[6].

Помимо простого описания, дополнительные данные о продуктах могут быть доступны из внешних источников. Их семантика может быть обогащена, напри-

[6] В работе не рассматривается случай обучения рекомендательной системы без учета рейтингов, выставленных продукту пользователем.

мер, с помощью глобальной онтологии DBpedia или ресурсов Linked Data Web. Имеется также много источников информации о пользователе. Это могут быть данные, почерпнутые из социальных сетей. Они обеспечивают также информацией о его друзьях. В этих же целях может быть использована и история посещения им различных веб-сайтов. В результате рекомендательная система, имея доступ к большому объему распределенных гетерогенных данных о пользователе, потенциально имеет возможность использовать различные механизмы обучения для того, чтобы построить достаточно богатый и достоверный профиль конкретного пользователя в семантических терминах. Как уже отмечалось, эти данные обладают всеми свойствами больших данных, а потому задача обучения профиля пользователя относится к классу задач интеллектуальной обработки больших данных, и потому все проблемы, свойственные этой задаче, необходимо решать и при построении профиля пользователя.

На сегодняшний день наиболее разработанные семантические модели знаний, включая модели знаний о профиле пользователя, основаны на использовании онтологии. Такая модель принята и в данной работе (см. главу 4). Она позволяет описывать в рамках одной модели множество разнообразных интересов пользователя, оперируя понятиями онтологии из различных областей. В подразделах 3.2.3 – 3.2.5 был описан алгоритм автоматического построения такой онтологии данных, принятый в данной работе. Он основан на семантическом анализе понятий и позволяет эффективно обрабатывать большие объемы неструктурированной информации о пользователе и о продуктах, доступной из множества распределенных гетерогенных источников.

В этой технологии разработка самой онтологии данных и выделение в ней компонент, описывающих интересы и предпочтения пользователя, являются тесно связанными задачами. Профиль пользователя формируется как структурированное подмножество понятий онтологии, которые наиболее точно соответствуют интересам пользователя и являются причинами, определяющими его мо-

тивацию при выборе продукта и его оценки в терминах рейтинга Значения рейтинга, постав-ленные пользователем в соответствие тем или иным продуктам, в процессах машинного обучения играют роль интерпретации данных.

Итак, целью обучения профиля пользователя в рекомендательных системах третьего поколения является выявление причин, побудивших пользователя выставить тот или иной рейтинг продукту. В качестве такой причины может выступать как некоторое понятие построенной онтологии данных, так и значение некоторого структурированного атрибута данных. Например, выбор пользователя обычно сильно зависит от контекста, а атрибуты контекста также могут выступать в качестве косвенных причин выбора пользователем того или иного продукта и значения присвоенного ему рейтинга. Таким образом, вместе с атрибутами понятий, представляющими продукт, атрибуты контекста формируют некоторую структурированную ситуацию, которая отражает интересы пользователя. И эту ситуацию нужно рассматривать как структурированное множество причин, совместно определяющих выбор пользователя.

Далее описывается формальная постановка задачи обучения профиля пользователя в рекомендательных системах третьего поколения и представляются алгоритмы формирования рекомендаций на основе построенного профиля. Для экспериментального тестирования алгоритмов, оценки их точности и вычислительной эффективности используется набор данных Amazon, выбор которого был обоснован в разделе 3.3.

4.2 Построение структурированного множества потенциальных интересов пользователя в задачах обучения рекомендательных систем третьего поколения

4.2.1 Постановка задачи обучения профиля интересов пользователя

Рассмотрим краткую информацию о наборе данных Amazon, выбранном в качестве тестового множества и сформулируем задачу извлечения множества интересов пользователей по данным этого набора.

Напомним, что набор содержит информацию о большом количестве продуктов из различных предметных областей (видео, музыка, книги). Каждый продукт,

помимо названия, характеризуется набором внутренних категорий Amazon, к которым он принадлежит. Эти категории образуют иерархическую структуру и могут рассматриваться как некие бинарные атрибуты соответствующего продукта. Кроме этого, для каждого продукта дан список пользователей, которые его приобретали и выставили ему оценку по пятибалльной шкале (1 - наихудшая оценка, 5 - наилучшая). Это дает возможность структурировать информацию таким образом, чтобы можно было работать с отдельными пользователями по всему множеству продуктов Amazon, используя методы машинного обучения с учителем. Далее, для каждого пользователя, используя дополнительную информацию о приобретенных им продуктах, извлеченную из DBpedia, и модель семантического анализа понятий (см. далее) можно построить иерархию понятий, соответствующую онтологии интересов пользователя, как это описано в подразделе 3.4.

Сформулируем задачу обучения профиля пользователя на основе данных о продуктах, имеющихся в Amazon, и их оценках пользователями, совместно с которыми будет использоваться также информация из внешних источников. Пусть имеется некоторое множество экземпляров продуктов (товаров, фильмов, книг и т.п.) из набора Amazon с их оценками конкретным пользователем. Каждый экземпляр продукта из этого множества описывается некоторым набором атрибутов. В частности, для данных Amazon такими атрибутами являются внутренние категории Amazon, к которым относится этот экземпляр продукта, и понятия построенной онтологии данных. Отметим, что экземпляры продукта (далее они называются также экземплярами, или примерами данных Amazon) могут иметь дополнительные атрибуты различных типов, например, численные, номинальные и др.

Целью обучения профиля пользователя рекомендательной системы является поиск причин, которые определили ту или иную оценку пользователя. В качестве причины в данном контексте может выступать то или иное значение (или значения) некоторого атрибута (или атрибутов) экземпляра данных. Задача состоит в том, чтобы найти набор таких причин, предпочтительно минимального объема, который наилучшим образом характеризует интересы пользователя и, таким образом, описывает его профиль (мотивацию выбора пользователя).

В профиле пользователя рекомендательной системы они представляются в виде множества правил вида

$$P_k^i\left(x_j^i \in \tilde{\boldsymbol{X}}_s^i\right) \rightarrow \omega_k,\ k=1,\ldots q. \tag{4.1}$$

где $\tilde{\boldsymbol{X}}_s^i$ - подмножество значений атрибута X^i, найденное для некоторого пользователя и отражающее его интересы; ω_k – рейтинг (экземпляра данных) продукта, а предикаты вида $P_k^i\left(x_j^i \in \tilde{\boldsymbol{X}}_s^i\right)$ принимают истинное значение для некоторого экземпляра данных тогда, когда значение x_j^i атрибута X^i для этого экземпляра принадлежат подмножеству $\tilde{\boldsymbol{X}}_s^i$.

В том случае, когда атрибуты примеров предоставлены в понятиях онтологии, профиль пользователя включает в себя также дополнительные семантические связи между атрибутами.

4.2.2 Принятие решений в рекомендательной системе как задача ассоциативно-причинной классификации

Главной задачей любой рекомендательной системы является предсказание отношения пользователя к некоторому новому для него продукту в терминах рейтинга. В случае набора данных Amazon (и многих других наборов данных из области рекомендательных систем) пользователю предоставляется возможность присвоить купленному продукту рейтинг по пятибалльной шкале. Таким образом, основная задача рекомендательной системы третьего поколения, предназначенной для работы с данными, подобными данным Amazon, это предсказание рейтинга $\boldsymbol{\omega} \in \{1, 2, 3, 4, 5\}$, который целевой пользователь $\boldsymbol{U_t}$ присвоит некоторому новому для него продукту $\boldsymbol{I_t}$. При этом принятие решения о рейтинге должно приниматься на основе характеристик товара $\boldsymbol{I_t}$ и профиля пользователя Pr_{U_t}, содержащего его интересы в форме правил вида (4.1).

Рейтинг ω_k, $k \in \{1,\ldots,5\}$, выставляемый продуктам, может рассматриваться как метка класса. Тогда правила вида (4.1) можно рассматривать как ассоциативно-причинные правила, выступающие в роли элементарных решателей, а сама задача предсказания рейтинга формулируется как задача ассоциативно-причин-

ной многоклассовой классификации продуктов относительно предпочтений конкретного пользователя [129].

Одним из наиболее эффективных способов решения такой задачи классификации является ее декомпозиция на несколько задач бинарной классификации. При этом возможны различные варианты её разбиения, например, использование модели «один класс» - «все другие классы», использование бинарного дерева решений [130], и другие.

В работе для выработки рекомендаций используется метод бинарного дерева решений. На рисунке 4.1 представлен выбранный вариант дерева решений для значений рейтинга $\boldsymbol{\omega} = \{1, 2, 3, 4, 5\}$. В таком варианте принятия решений множество правил вида (4.1) необходимо построить для каждого дерева. В последующих подразделах описаны последовательные этапы алгоритма формирования множеств предикатов $P_k^i\left(x_j^i \in \widetilde{\boldsymbol{X}}_s^i\right)$, которые соответствуют посылкам правил принятия решений в каждом узле дерева, представленного на рисунке 4.1.

Совокупность всех этих правил для всего множества узлов есть структурированное представление профиля пользователя рекомендательной системы, в котором каждому узлу ставится в соответствие множество интересов, определяющих его выбор в этом узле. Принятие решений в каждом узле выполняется с помощью метаправил объединения решений, предлагаемых правилами узла. Такую модель знаний принято называть знаниями для действий (англ. *actionable knowledge*).

4.2.3 Агрегирование и первичная фильтрация множества потенциальных интересов пользователя

Рассмотрим процедуру формирования подмножеств значений атрибутов $\widetilde{\boldsymbol{X}}_s^i$, которые являются областью истинности предикатов $P_k^i\left(x_j^i \in \widetilde{\boldsymbol{X}}_s^i\right)$ пользователя $\boldsymbol{U_t}$, поставленных в соответствие каждому узлу принятия решений бинарного дерева (рисунок 4.1). Процедура использует идею агрегирования, предложенную в [131, 132].

Отметим, что значения атрибутов, характеризующих экземпляры данных (продукты), могут быть представлены в категориальной, булевой, порядковой или

числовой шкалах, а также могут быть текстами на естественном языке. Текстовые данные с помощью алгоритмов семантического анализа понятий (см. главу 3) могут быть охарактеризованы множеством понятий онтологии, т.е. множеством атрибутов в номинальной шкале. С помощью унарных предикатов $P_k^i\left(x_j^i \in \tilde{\tilde{\boldsymbol{X}}}_s^i\right)$ агрегированные атрибуты, представленные в исходных данных в любой другой шкале, также в итоге преобразуются к номинальной шкале (шкале свойств, или имен, что то же самое). В итоге такого преобразования данных все они преобразуются к единой шкале измерения, а именно, к номинальной шкале. Поясним содержательно основную идею этого преобразования.

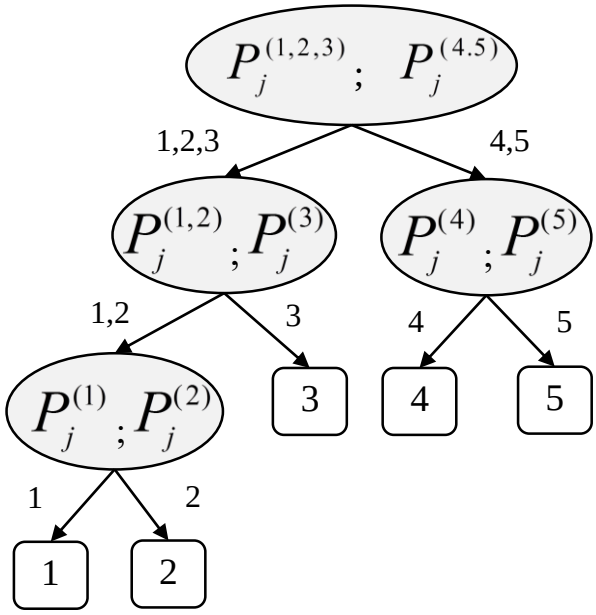


Рисунок 4.1 – Бинарное дерево решений для предсказания рейтинга $\boldsymbol{\omega} = \{1, 2, 3, 4, 5\}$

Это преобразование, по существу, является преобразованием агрегирования данных. В нем реализуется объединение отдельных значений атрибутов в агрегаты, которые обладают некоторым общим свойством по отношению к решаемой задаче классификации. Идея состоит в том, область значений $\tilde{\tilde{\boldsymbol{X}}}^i$ каждого атрибута $\boldsymbol{X}^i$, $i \in \{1, \ldots, m\}$, разделяется на три подобласти $\tilde{\tilde{\boldsymbol{X}}}_s^i \subseteq \tilde{\tilde{\boldsymbol{X}}}^i$, s = 1,2,3. К первой из них относятся те значения, которые чаще встречаются в одном из двух классов решаемой задачи классификации. Ко второй области относятся те значе-

ния, которые чаще встречаются в примерах другого класса. Третья область содержит те значения, которые встречаются в обоих классах примерно с одинаковой частотой. Эти области значений не содержат общих значений и $\bigcup_s \tilde{\boldsymbol{X}}_s^i = \tilde{\boldsymbol{X}}^i$. Некоторые из них могут быть пустыми. Напомним, что множества $\tilde{\boldsymbol{X}}_s^i \subseteq \tilde{\boldsymbol{X}}^i$, s = 1,2,3, определяют области истинности предикатов $P_k^i\left(x_j^i \in \tilde{\boldsymbol{X}}_1^i\right)$ и $P_k^i\left(x_j^i \in \tilde{\boldsymbol{X}}_2^i\right)$, которые ставятся в соответствие каждому узлу принятия решений бинарного дерева и играют роль *простейших классификаторов*, построенных с использованием всего лишь одного атрибута $\boldsymbol{X}^i$ из всего множества атрибутов. Заметим, что третья область значений $\tilde{\boldsymbol{X}}_3^i$ атрибута $\boldsymbol{X}^i$ далее не рассматривается.

Зададим формально правила (алгоритмы) формирования подмножеств $\tilde{\boldsymbol{X}}_s^i \subseteq \tilde{\boldsymbol{X}}^i$, s = 1,2,3. Процедура разбиения всего множества значений $\tilde{\boldsymbol{X}}^i$ атрибутов $\boldsymbol{X}^i$, $i \in \{1, \ldots, m\}$, на непересекающиеся подмножества $\tilde{\boldsymbol{X}}_s^i \subseteq \tilde{\boldsymbol{X}}^i$, s = 1,2,3, в данной работе основана на построении наивного байесовского классификатора[7] для двух альтернативных классов ω_a и $\omega_{\bar{a}}$. Для каждого значения $x_j^i \in \tilde{\boldsymbol{X}}^i$ по обучающим данным можно рассчитать эмпирическую (выборочную) вероятность $p(\omega_a | x_j^i)$ с которой экземпляр данных, относится к классу ω_a при условии, что атрибут $\boldsymbol{X}^i$ для этого экземпляра данных принимает значение x_j^i, и аналогичную вероятность $p(\omega_{\bar{a}} | x_j^i)$ для класса $\omega_{\bar{a}}$ Обозначим символами $p(\omega_a)$, $p(\omega_{\bar{a}})$ априорные вероятности классов ω_a и $\omega_{\bar{a}}$ в генеральной совокупности (точно они могут быть и неизвестны), а символом $p(x_j^i)$ – вероятность появления значения x_j^i в выборке данных.

Тогда, при условии, что классы ω_a и $\omega_{\bar{a}}$ альтернативны,

---

[7] В принципе, можно использовать и другой тип классификатора, но наивный байесовский классификатор представляется наиболее простым.

$$p(\omega_a \left| x_j^i \right.) + p(\omega_{\bar{a}} \left| x_j^i \right.) = 1 \tag{4.2}$$

и согласно теореме Байеса

$$p(\omega_a \left| x_j^i \right.) = \frac{p(x_j^i \left| \omega_a \right.) \cdot p(\omega_a)}{p(x_j^i \left| \omega_a \right.) \cdot p(\omega_a) + p(x_j^i \left| \omega_{\bar{a}} \right.) \cdot p(\omega_{\bar{a}})} \tag{4.3}$$

$$p(\omega_{\bar{a}} \left| x_j^i \right.) = \frac{p(x_j^i \left| \omega_{\bar{a}} \right.) \cdot p(\omega_{\bar{a}})}{p(x_j^i \left| \omega_a \right.) \cdot p(\omega_a) + p(x_j^i \left| \omega_{\bar{a}} \right.) \cdot p(\omega_{\bar{a}})} \tag{4.4}$$

При этом эмпирические вероятности, входящие в представленные выше формулы, вычисляются по данным обучающей выборки следующим образом:

$$p(x_j^i \left| \omega_a \right.) = \frac{n_a^i}{N_a}, \qquad p(x_j^i \left| \omega_{\bar{a}} \right.) = \frac{n_{\bar{a}}^i}{N_{\bar{a}}} \tag{4.5}$$

$$p(\omega_a) = \frac{N_a}{N}, \qquad p(\omega_{\bar{a}}) = \frac{N_{\bar{a}}}{N} \tag{4.6}$$

где n_a^i, $n_{\bar{a}}^i$ – количество экземпляров выборки, для которых атрибут X^i принимает значение x_j^i, и которые имеют метку класса ω_a и $\omega_{\bar{a}}$ соответственно; $N = N_a + N_{\bar{a}}$, где N – общее количество экземпляров в выборке, N_1, N_2 – количество экземпляров в классах ω_a и $\omega_{\bar{a}}$ соответственно.

Тогда

$$p(\omega_a \left| x_j^i \right.) = \frac{n_a^i}{n_a^i + n_{\bar{a}}^i}, \qquad p(\omega_{\bar{a}} \left| x_j^i \right.) = \frac{n_{\bar{a}}^i}{n_a^i + n_{\bar{a}}^i} \tag{4.7}$$

Таким образом, для каждого значения x_j^i атрибута X^i за один проход по данным можно рассчитать вероятности $p(\omega_a \left| x_j^i \right.)$ и $p(\omega_{\bar{a}} \left| x_j^i \right.)$ классов ω_a или $\omega_{\bar{a}}$ при условии, что атрибут X^i для этого продукта принимает значение x_j^i. Это справедливо в том случае, если выборка репрезентативна относительно апри-

орных вероятностей классов в том смысле, что в ней соблюдается пропорциональность количества представителей классов в соответствии с их априорными вероятностями.

По значениям этих вероятностей можно найти $\delta_{a,\bar{a}}^{i,j} = p(\omega_a | x_j^i) - p(\omega_{\bar{a}} | x_j^i)$. Выбрав некоторую константу $\delta_{a,\bar{a}}$, значение которой позволяет регулировать мощности подмножеств признаков, которые формируются в результате агрегирования, в ходе анализа значения $\delta_{a,\bar{a}}^{i,j}$, можно сформировать правила для выбора классов ω_a и $\omega_{\bar{a}}$ следующим образом:

– если $\delta_{a,\bar{a}}^{i,j} \geq \delta_{a,\bar{a}}$, то значение x_j^i атрибута $\boldsymbol{X}^i$ можно добавить в подмножество $\tilde{\boldsymbol{X}}_a^i$ истинности предиката $P_a^i\left(x_j^i \in \tilde{\boldsymbol{X}}_a^i\right)$, сформировать правило вида $P_a^i\left(x_j^i \in \tilde{\boldsymbol{X}}_a^i\right) \rightarrow \omega_a$ для соответствующего узла принятия решений и добавить его в предварительное множество правил профиля Pr_{U_t} пользователя $\boldsymbol{U_t}$; это правило формирует множество значений атрибута $\boldsymbol{X}^i$, обозначенное ранее символом $\boldsymbol{X}_1^i$;

– если $\delta_{a,\bar{a}}^{i,j} \leq -\delta_{a,\bar{a}}$, то в промежуточный набор правил можно добавить правило вида $P_{\bar{a}}^i\left(x_j^i \in \tilde{\boldsymbol{X}}_{\bar{a}}^i\right) \rightarrow \omega_{\bar{a}}$; это правило формирует множество значений атрибута $\boldsymbol{X}^i$, обозначенное ранее символом $\boldsymbol{X}_2^i$;

– если $-\delta_{a,\bar{a}} < \delta_{a,\bar{a}}^{i,j} < \delta_{a,\bar{a}}$, то значение x_j^i атрибута $\boldsymbol{X}^i$ нельзя добавить в множество истинности какого-либо предиката; это правило формирует множество значений атрибута $\boldsymbol{X}^i$, обозначенное ранее символом $\boldsymbol{X}_3^i$.

Напомним, что продукты набора данных Amazon имеют только бинарные атрибуты, описывающие принадлежность к той или иной внутренней категории Amazon. В качестве атрибутов выступают также понятия построенной онтологии данных, которые соотносятся с тем или иным продуктом. Такой характер данных множества Amazon сильно снижет вычислительную сложность описанной процедуры агрегирования в вычислительном отношении, поскольку в данном случае

область значений предметных переменных формируемых предикатов имеет только два значения.

Следует отметить, что значения n_a^i, $n_{\bar{a}}^i$ для значений x_j^i тех атрибутов, которые представлены понятиями онтологии данных, построенной с помощью семантического анализа понятий, вычисляются еще на этапе построения этой онтологии. Таким образом, если атрибуты представлены только понятиями онтологии, то повторного прохода по множеству данных для рассматриваемых данных Amazon вообще не требуется (см. подраздел 3.2.5). Особенности использования предложенной процедуры агрегирования гетерогенных данных в случае, когда среди атрибутов имеются атрибуты, измеренные в числовых, порядковых и номинальных шкалах, а также примеры ее использования для различных наборов гетерогенных данных можно найти в работах [7, 131, 132, 133].

В общем случае описанная процедура позволяет значительно снизить трудности, связанные с большой размерностью данных и гетерогенностью признаков. Построенные предикаты могут рассматриваться как компоненты нового редуцированного пространства признаков, а сформированные правила выступать в качестве простых правил классификации. Подчеркнем, что в этом подходе типы исходных атрибутов, значения которых подвергаются агрегированию, могут быть измерены в любой шкале (номинальной, числовой, порядковой, булевой) и включать в себя тексты на естественном языке, тогда как все результирующие атрибуты (признаки) будут булевыми утверждениями в форме одноместных предикатов.

Построенное множество правил представляет бинарные ассоциативные связи атрибутов данных и классов состояний, существующие в используемых обучающих данных. Заметим, что в описанной процедуре не используются традиционные метрики оценки силы ассоциативной связи типа поддержки и уверенности. Это преобразование данных может рассматриваться как первичная фильтрация множества потенциальных интересов пользователя. Следующим шагом обучения профиля пользователя является выделение из всего полученного множества пра-

вил тех, которые имеют причинный характер. Этот шаг осуществляется с помощью метрики причинной связи, обоснование выбора которой дано в главе 2.

4.2.4 Причинный анализ и вторичная фильтрация множества потенциальных интересов пользователя

Из всех правил, полученных на предыдущем этапе для каждого узла принятия решений, необходимо выделить те правила, которые отражают причины, побудившие пользователя поставить продукту тот или иной рейтинг.

Подчеркнем, что экспериментальные и формальные исследования, выполненные в работах [28, 39] убедительно показали, что в задачах принятия решений именно причинные связи между атрибутами в данных оказываются наиболее перспективными для построения правил классификации.

Помимо этого, причинная фильтрация правил позволит еще больше сократить размерность пространства признаков, что является очень важным с точки зрения вычислительной эффективности в задачах обучения на больших данных.

Напомним, что в главе 3 была экспериментально обоснована схема выделения причинных зависимостей между атрибутами данных. Для этих целей из всего множества метрик оценки силы связи между атрибутами на основе формального и экспериментального анализа были выбраны метрики, которые представляются наиболее перспективными с точки зрения способности выявлять правила, представляющие причинные связи в данных. Такими метриками оказались мера Клозгена [77]:

$$K(A,B) = \sqrt{p_{AB}} \cdot \left(p_{B|A} - p_B\right), \tag{4.8}$$

и коэффициент регрессии [88, 89]:

$$R(A,B) = \frac{\left(p_{AB} - p_A \cdot p_B\right)}{p_A \cdot \left(1 - p_A\right)}, \tag{4.9}$$

где A и B – это атрибуты, которые интерпретируются как случайные события, а символом p обозначены выборочные вероятности событий, указанных в его нижнем индексе. Заметим, что в результате предобработки данных и фильтрации правил, рассмотренных в предыдущем подразделе, все атрибуты представляются в

бинарной шкале свойств в виде утверждений о принадлежности значения атрибута некоторой области, т.е. $P_a^i\left(x_j^i \in \tilde{\boldsymbol{X}}_a^i\right)$.

Каждому правилу вида (4.1), полученному в результате агрегирования и первичной фильтрации для каждого узла принятия решений, можно поставить в соответствие значения метрик (4.11-4.12), вычисленных следующим образом:

$$K\left(P_k^i, \omega_k\right) = \sqrt{p\left(P_k^i \omega_k\right)} \cdot \left(p\left(\omega_k \middle| P_k^i\right) - p\left(\omega_k\right)\right), \tag{4.10}$$

$$R\left(P_k^i, \omega_k\right) = \frac{p\left(P_k^i \omega_k\right) - p\left(P_k^i\right) \cdot p\left(\omega_k\right)}{p\left(P_k^i\right) \cdot \left(1 - p\left(P_k^i\right)\right)}, \tag{4.11}$$

где $p\left(P_k^i\right)$ - это оценка вероятности появления какого-либо из значений $x_j^i \in \tilde{\boldsymbol{X}}_k^i$ в выборке данных, а $\tilde{\boldsymbol{X}}_k^i$ - это область истинности предиката P_k^i. Значение метрики, полученное для каждого правила, может рассматриваться как его вес, отражающий «силу» причинной связи, представляемой правилом.

После вычисления значений выбранной метрики $\mu(P_k^i, \omega_k)$ для каждого правила, представляющего потенциальный интерес пользователя, эти правила могут быть отсортированы в порядке убывания модуля значения метрики. Сортировка выполняется для каждого узла принятия решений и для правил каждого класса в этом узле отдельно.

Фильтрация правил профиля выполняется следующим образом: в профиле пользователя для каждого узла принятия решений на этом шаге остаются только правила вида (4.1), удовлетворяющие следующему условию:

$$\left|\mu\left(P_k^i, \omega_k\right)\right| \geq \delta_{\min}^{\mu} . \tag{4.12}$$

Значение $\delta_{\min}^{\mu}$ выбирается экспериментально, исходя из мощности множества правил и среднего значения модуля метрики для правил в каждом узле.

В результате такой фильтрации в профиле пользователя рекомендательной системы на этом шаге останутся только те правила, которые соответствуют наиболее «сильным» его интересам, т.е. причинам, побуждающим его выставлять тот или иной рейтинг продуктам.

Фильтрация, сохраняющая только самые «сильные» правила (с причинной точки зрения), позволяет в очередной раз значительно сократить размерность модели данных, улучшив тем самым вычислительную эффективность последующих процедур обучения и принятия решений без заметной потери точности принятия решений, как показали экспериментальные исследования (см. раздел 4.6).

4.2.5 Кластеризация интересов пользователя и сокращение размерности пространства описания профиля

После выполнения процедур, описанных в предыдущем подразделе, в каждом узле дерева, представляющего модель принятия решений пользователем (модель профиля пользователя), останутся только те правила, которые отражают причинные зависимости в данных.

Однако, для больших данных размерность признакового пространства может, по-прежнему, оставаться очень большой. Кроме того, обнаруженные причинные правила могут быть сильно коррелированными, а их совместное использование в моделях принятия решений может привести даже к негативному эффекту. Например, если использовать взвешенное голосование в качестве механизма объединения решений, даваемых разными правилами типа (4.1) одного узла дерева принятия решений (см. рисунок 4.1), то фактически одна и та же причина будет учитываться дважды, что может привести к появлению ошибочных решений. Этот явление называют проблемой недостаточного разнообразия классификаторов, которая подробно описана в [74]. Другое обоснование слабой полезности совместного использования сильно коррелированных правил состоит в том, что они, в основном, выдают одни и те же решения для большинства примеров данных, фактически повторяя решения друг друга.

Для обеспечения необходимого разнообразия классификаторов предлагается выполнять кластеризацию причинных правил в каждом узле принятия решений таким образом, чтобы внутри кластеров правила были сильно коррелированы, а правила разных кластеров - слабо коррелированы. Кластеризация выполняется отдельно для подмножества правил каждого класса в узлах принятия решений.

Коэффициент корреляции для каждой пары правил (представленных соответствующими предикатами) может быть вычислен по следующей формуле [88]:

$$Cor(P_k^i, P_k^j) = \frac{p(P_k^i P_k^j) - p(P_k^i) \cdot p(P_k^j)}{\sqrt{p(P_k^i) \cdot \left(1 - p(P_k^i)\right) \cdot p(P_k^j) \cdot \left(1 - p(P_k^j)\right)}}. \quad (4.13)$$

Отметим, что все оценки вероятностей, фигурирующие в формуле, уже были ранее вычислены для каждого правила на предыдущих шагах процедуры построения профиля пользователя, либо могут быть вычислены на основе уже имеющихся данных, и поэтому для подсчета коэффициентов корреляции не требуется дополнительных проходов по данным.

Далее можно сформировать матрицу $\boldsymbol{Cor}_k = \left\| Cor\left(P_k^i, P_k^j\right) \right\|_{i,j}$ корреляции правил внутри подмножества. Такая матрица может интерпретироваться как матрица расстояний между узлами графа, в качестве которых выступают анализируемые правила. В этом случае для кластеризации правил можно использовать алгоритмы кластеризации типа разрезания графа [134].

В работе кластерный анализ правил классификации выполняется с помощью метода корреляционных плеяд [134, 135].

Для выделения кластеров методом корреляционных плеяд необходимо задать некоторый пороговый параметр и обнулить в матрице расстояний (матрице корреляции) те значения, которые удовлетворяют условию:

$$1 - \left| Cor\left(P_k^i, P_k^j\right) \right| \geq \boldsymbol{\tau}. \quad (4.14)$$

Это процедура по сути реализует удаление из графа, узлы которого представляют правила классификации, тех ребер, длина которых больше расстояния $\boldsymbol{\tau}$.

Таким образом, связанными в графе остаются только те вершины, которые соответствуют наиболее коррелированным правилам классификации.

Затем с помощью алгоритма поиска в ширину (либо в глубину) можно выделить все компоненты связности графа. Полученные компоненты и будут представлять собой кластеры наиболее сильно коррелированных правил.

Для подбора параметра $\boldsymbol{\tau}$ необходимо построить гистограмму распределений значений $1 - \left| Cor\left(P_k^i, P_k^j\right) \right|$ всех правил. На полученной гистограмме можно выделить

два пика – один соответствует внутрикластерным расстояниям, второй – межкластерным расстояниям. Параметр $\boldsymbol{\tau}$ выбирается из зоны минимума между этими пиками [134]. Второй вариант выбора порогового значения $\boldsymbol{\tau}$ – это экспериментальный подбор.

Далее правила внутри кластеров упорядочиваются следующим образом.

Правило $\boldsymbol{P_k^j}$ предшествует правилу $\boldsymbol{P_k^i}$ если

- $\mu(P_k^i, \omega_k) > \mu(P_k^j, \omega_k)$,

или

- если $\mu(P_k^i, \omega_k) = \mu(P_k^j, \omega_k)$, но покрытие правила P_k^i больше покрытия правила P_k^j,

или

- если $\mu(P_k^i, \omega_k) = \mu(P_k^j, \omega_k)$ и покрытие правила P_k^i равно покрытию правила P_k^j, но правило P_k^i было сформировано раньше правила P_k^j;

- иначе правило P_k^i предшествует правилу P_k^j.

В итоговую модель принятия решений добавляется первое правило из упорядоченного списка правил в кластере.

Еще раз подчеркнем, что кластеризация правил позволяет удалить из итоговой модели принятия решений бесполезные правила – элементарные классификаторы, то есть те правила, использование которых не ведет к повышению точности алгоритма, более того их удаление позволяет повысить вычислительную эффективность работы системы в целом.

На рисунке 4.2 представлена блок-схема процедуры кластеризации правил в группе и выбора правила из кластера.

В приложении В приведен псевдокод всех компонент алгоритма обучения профиля пользователя.

В результате применения описанных процедур фильтрации в итоговом профиле пользователя в каждом узле принятия решений останутся только причинные

правила, представляющие его интересы, которые слабо коррелируют друг с другом, отражая тем самым разнообразие этих интересов.

Рисунок 4.2 - Блок-схема процедуры кластеризации правил в группе и выбора правила из кластера

На рисунке 4.3 представлена схема описанной выше технологии построения профиля пользователя применительно к набору данных Amazon.

Выделим важные преимущества полученной формальной модели представления профиля пользователя:

1. Предложенная формальная модель естественным образом реализует персонализацию рекомендательной системы.

2. Модель компактно, четко и однозначно интерпретируема с семантической точки зрения, поскольку каждое правило представляет отдельный интерес пользователя, выраженный понятием онтологии данных или подмножеством значений некоторого атрибута, которые интересны пользователю.

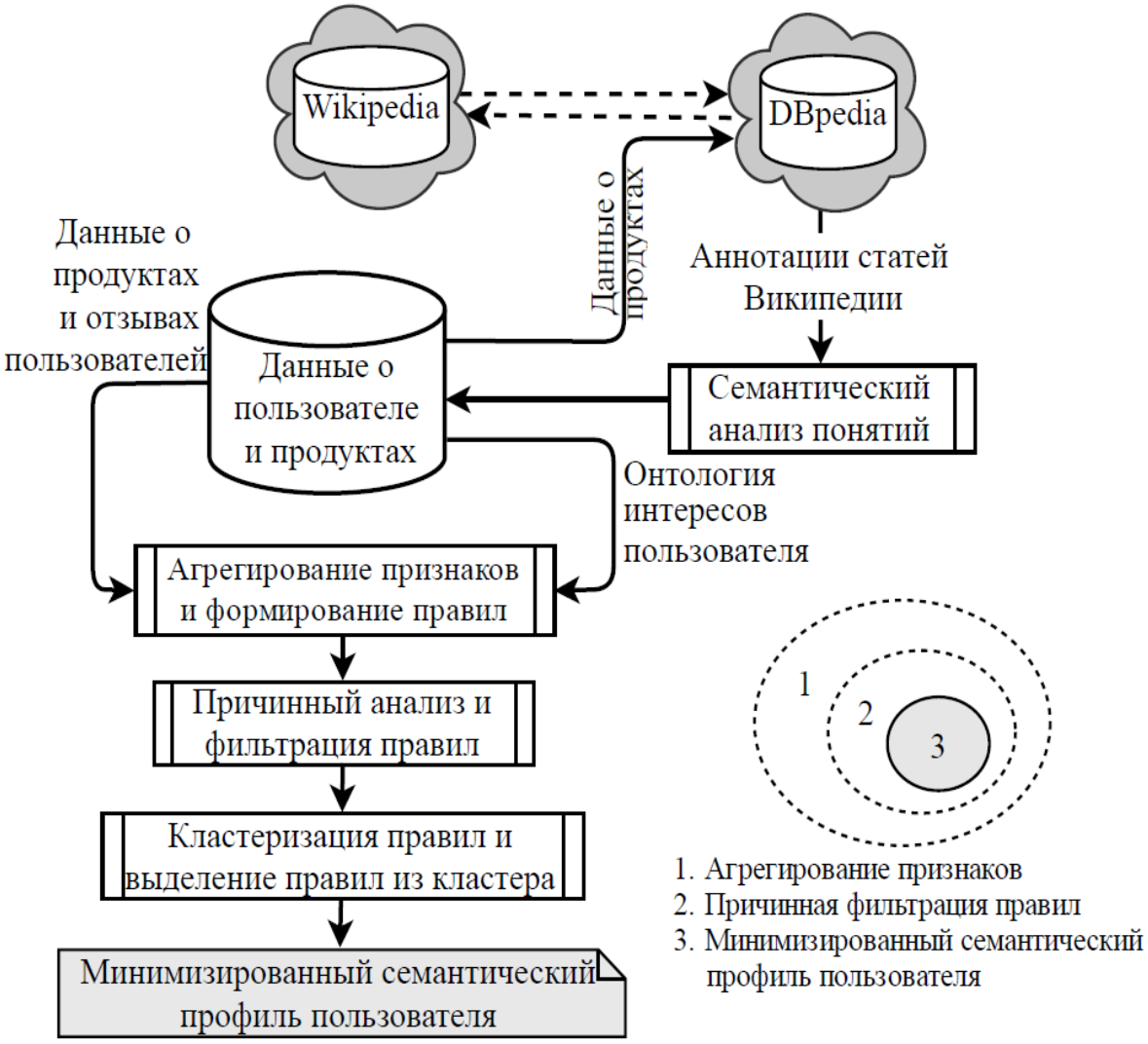


Рисунок 4.3 - Схема технологии построения профиля пользователя

3. Профили различных пользователей представлены в терминах одних и тех же понятий онтологии данных, даже если первичная информация о пользователях была получена из различных источников (если мы говорим о той части профиля, которая построена на основе понятий, извлеченных из DBpedia) и поэтому они легко сопоставимы с помощью какой-либо семантической меры сходства.

4. Профиль пользователя представлен в форме, хорошо применимой на практике: механизм принятия решений, т.е. дерево решений, по сути, встроен в сам профиль пользователя. Необходимо лишь выбрать модель объединения решений элементарных классификаторов в каждом узле дерева.

5. Контекст может быть представлен в виде понятий той же онтологии предметной области, а контекстные атрибуты в этом случае будут агрегированы и подвергнуты причинному анализу точно так же, как любые другие атрибуты данных. Поэтому предлагаемая модель профиля пользователя может быть легко обобщена на случай контекстно-зависимых рекомендаций.

6. То же самое касается технологии кросс-доменных рекомендаций: в любом случае, кросс-доменные рекомендации возможны, когда соответствующие предметные области имеют что-то общее, что может быть выражено в терминах общих понятий онтологий обеих предметных областей. Конкретные методы кросс-доменного принятия решений на основе онтологии описаны в подразделе 4.3.4.

7. Наконец, слияние информации из нескольких, возможно, распределенных источников данных также может быть естественно выполнено, если использовать онтологию как мета-модель, определенную на верхнем уровне распределенных источников данных, что позволит использовать любые стратегии в области слияния данных, в том числе и те, которые обеспечивают сохранение конфиденциальности данных.

4.3 Алгоритмы выработки рекомендаций

Главной задачей любой рекомендательной системы является предсказание отношения пользователя к некоторому новому для него продукту.

Напомним, что в общем случае, выделяют три базовых метода, которые используются для принятия решений в рекомендательных системах [30]:

- метод фильтрации контента;
- метод коллаборативной фильтрации;
- гибридные методы, которые комбинируют оба названных выше подхода и, возможно, используют еще какие-то новшества.

Далее будут рассмотрены разработанные алгоритмы выработки рекомендаций, основанные на упомянутых методах. Предложенные алгоритмы выработки рекомендаций используют семантический профиль интересов пользователя, в той форме, которая описана в предыдущих разделах, а также идею ассоциативно-причинной классификации.

4.3.1 Формирование рекомендаций методом фильтрации контента

Наиболее простым и естественным способом выработки рекомендаций на основе семантического профиля интересов пользователя, разработка которого описана выше, является метод фильтрации контента.

К преимуществам метода фильтрации контента можно отнести «прозрачность» механизма рекомендаций – рекомендательная система, основанная на фильтрации контента, всегда может дать объяснения своим рекомендациям, то есть продемонстрировать пользователю те свойства рекомендованного продукта, которые соответствуют его интересам. В отличие от методов коллаборативной фильтрации, рекомендательной системе на основе фильтрации контента необходима информация, относящаяся только к одному пользователю, что позволяет сохранять конфиденциальность информации о всех пользователей рекомендательной системы. И наконец, такие рекомендательные системы способны рекомендовать пользователем новые продукты – для выработки рекомендаций не требуется информация о том, какие оценки этому продукту поставили другие пользователи, так как рекомендации основываются только на свойствах продукта [136].

Рассмотрим подробнее разработанный алгоритм выработки рекомендаций методом фильтрации контента на основе семантического профиля пользователя с помощью ассоциативно–причинной классификации.

Пусть для некоторого пользователя U_t (например, пользователя, чьи оценки представлены в наборе данных Amazon) рекомендательной системы на основе данных об оцененных им продуктах $\boldsymbol{I} = \{I_1, \ldots, I_z\}$, с привлечением дополнительной информации из глобальной базы знаний DBpedia построен семантический профиль интересов $\boldsymbol{Pr}_{U_t}$. Этот профиль содержит бинарное дерево решений (рисунок 4.1) с присвоенными каждому узлу наборами ассоциативно-причинных пра-

вил классификации вида $P_k^i\left(\widetilde{X}_k^i\right) \rightarrow \omega_k$. При этом каждому правилу поставлено в соответствие значение метрики оценки «силы» причинной связи $\mu(P_k^i, \omega_k)$.

С помощью алгоритма выработки рекомендаций необходимо предсказать оценку (рейтинг) ω_{U_t, I_t}, которую пользователь U_t поставит новому для него продукту I_t на основе профиля $\boldsymbol{Pr}_{U_t}$ и характеристик продукта I_t.

Для продукта I_t из набора данных, используемого для обучения, могут быть извлечены значения всех его атрибутов. Помимо этого, информация о продукте может быть обогащена информацией из DBpedia (эта процедура описана в разделе 4.3). Полученные понятия и категории Википедии могут быть добавлены к списку атрибутов продукта как бинарные атрибуты.

Далее необходимо выбрать те правила профиля пользователя, которые сработают на продукте I_t. Опишем процедуру выбора правил, которая должна выполняться для каждого узла бинарного дерева принятия решений.

Пусть в некотором узле дерева решений есть множество предикатов вида $P_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right)$, а для продукта I_t извлечены значения всех его атрибутов X^j, $j = 1,\ldots,m$.

1. Если все предикаты $P_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right)$ (посылки правил классификации) в узле дерева принятия решений рассмотрены, то переход к п. 11, иначе к п. 2.

2. Выберем следующий не просмотренный предикат $P_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right)$ из списка правил.

3. Если все атрибуты продукта I_t просмотрены, то выполняется переход к п. 1, иначе – к п. 4.

4. Выберем следующий атрибут X^j из списка атрибутов продукта I_t.

5. Если выбранный атрибут X^j соответствует атрибуту X^i из профиля пользователя, для которого построен предикат $P_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right)$, то выполняется переход к п.4, иначе – к п.2.

6. Если все значения $x_v^i \in \widetilde{\boldsymbol{X}}_k^i$ просмотрены, то п. 1, иначе п. 7.

7. Выбирается следующее не просмотренное значение $x_v^i \in \tilde{\boldsymbol{X}}_k^i$ (если атрибут X^i целочисленный, то $x_v^i \in \tilde{\boldsymbol{X}}_k^i$ будет представлять собой интервал значений атрибута X^i, полученный в ходе предварительной дискретизации всех значений атрибута X^i).

8. Если значение $x_t^j = x_v^i$ (или $x_t^j \in x_v^i$, если x_v^i – это некоторый интервал значений из $\tilde{\boldsymbol{X}}_k^i$), где x_t^j - это значение атрибута X^j для продукта I_t, то выполняется переход к п. 9, иначе – к п. 1.

9. Добавляется правило, соответствующее предикату $P_k^i\left(\tilde{\boldsymbol{X}}_k^i\right)$, в список правил, сработавших для продукта I_t.

10. Переход к п. 1.

11. Конец.

Напомним, что рейтинг продукта отождествляется с меткой класса, а потому определить метку класса или определить рейтинг продукта – это одно и то же.

Используя выбранные правила классификации вида $P_k^i\left(\tilde{\boldsymbol{X}}_k^i\right) \rightarrow \omega_k$, с помощью процедуры простого или взвешенного голосования простых классификаторов, представленных правилами, в каждом узле дерева можно принять решение, к какому классу – ω_a или $\omega_{\bar{a}}$, следует отнести продукт в этом узле: решение принимается в пользу класса, набравшего больше голосов. В случае взвешенного голосования в качестве веса классификатора выступает значение метрики $\mu(P_k^i, \omega_k)$, поставленной в соответствие посылке правила классификации. Процедура голосования применяется в каждом узле дерева принятия решений от корня дерева к листьям. На выходе формируется метка итогового класса продукта, соответствующая рейтингу ω_{U_t,I_t}, который, предположительно, пользователь U_t присвоит продукту I_t.

В приложении В приведен псевдокод всех компонент алгоритма выработки рекомендаций методом фильтрации контента.

Следует отметить, что все методы выработки рекомендаций с помощью фильтрации контента обладают несколькими недостатками. Перечислим

наибощественные из них. Во-первых, профиль пользователя может получится «переобученным». Если некоторый пользователь до момента обучения по какой-то причине оценивал только фильмы-комедии, то рекомендательная система с большой долей вероятности будет рекомендовать этому пользователю только фильмы этого жанра. Иначе говоря, «идеальный» рекомендательный алгоритм на основе метода фильтрации контента не способен порекомендовать пользователю что-то новое, но интересное для него. Второй существенный недостаток методов на основе фильтрации контента связан с «проблемой нового пользователя» [136]. Очень сложно построить профиль, хорошо отражающий интересы пользователя, если он оценил только небольшое число продуктов («чего не знаешь, в том нет потребности»). Отметим, однако, что построение семантического профиля пользователя и обогащение информации о продуктах с помощью глобальных баз знаний позволяет ослабить негативные последствия обоих этих недостатков методов фильтрации контента.

Экспериментальное исследование алгоритма выработки рекомендации методом фильтрации контента на основе семантического профиля пользователя с помощью ассоциативно-причинной классификации путем взвешенного и простого голосования на наборе данных Amazon представлено в подразделе 3.5.

4.3.2 Формирование рекомендаций гибридным методом коллаборативной фильтрации

Методы коллаборативной фильтрации основаны на предположении, что рейтинг пользователя для нового продукта будет сходен с рейтингами других пользователей с похожими интересами.

Приведем основные преимущества этого подхода, выделенные в работах [30, 137] и многих других. Первым в списке преимуществ подхода чаще всего называют то, что системы, основанные на коллаборативной фильтрации, учитывают только оценки, выставленные пользователями, и не требуют никакой другой информации о пользователе и продуктах. На наш взгляд, это скорее недостаток, чем преимущество подхода. Такой метод рекомендаций является чисто статистическим и, фактически, не учитывают персональные предпочтения конкретного поль-

зователя. Отметим, что предлагаемый ниже метод коллаборативной фильтрации не обладает описанным «преимуществом». Вторым, действительно важным, на наш взгляд, преимуществом подхода является возможность учета при оценке предпочтений не только рассматриваемого пользователя, но также предпочтений пользователей со сходными интересами. Отсюда вытекает еще одно значимое преимущество данного подхода, а именно возможность рекомендовать что-то совсем новое, но, возможно, интересное для пользователя (англ. *serendipity*), используя информацию о предпочтениях похожих пользователей.

Следует отметить, что рекомендательные системы, основанные на коллаборативной фильтрации, по данным [30], в общем случае, показывают более высокую точность, чем системы, основанные на фильтрации контента. Возможно, это связано с тем, что большинство разработанных ранее рекомендательных систем на основе фильтрации контента не учитывают в полной мере семантику интересов пользователя.

К основным недостаткам систем, основанных на коллаборативной фильтрации, можно отнести проблему разрежённости данных: количество рейтингов, которые известны рекомендательной системе всегда значительно меньше, чем общее количество продуктов, рейтинги которых потребуется предсказать. Чем меньше пользователей оценило продукт, тем меньше вероятность того факта, что этот продукт будет кому-то еще порекомендован, даже если имеющиеся рейтинги высоки. Такие системы, в отличие от систем фильтрации контента, не способны давать рекомендации относительно новых продуктов, так как для таких продуктов вообще отсутствует информация о рейтингах других пользователей. Системы, коллаборативной фильтрации, так же как и системы фильтрации контента, плохо решают «проблему нового пользователя». Кроме этого, к ней добавляется и проблема «белой вороны» – так говорят о ситуации, когда интересы пользователя не совпадают с интересами других пользователей рекомендательной системы, следовательно, в системе не оказывается пользователей, чьи рейтинги можно использовать при выработке рекомендаций.

Отметим, что описанный ниже подход лишен большинства описанных недостатков, благодаря использованию семантического профиля интересов пользователя и семантической метрики сходства пользователей, вычисляемой на основе этого профиля. Благодаря этому, подход скорее можно отнести к гибридным методам выработки рекомендаций с использованием в основе метода коллаборативной фильтрации.

Рассмотрим формальную постановку задачи выработки рекомендаций методом коллаборативной фильтрации.

Пусть для множества пользователей $U_1, \ldots, U_e$ рекомендательной системы (например, пользователей, чьи оценки присутствуют в данных Amazon) на основе данных об оцененных ими продуктах с привлечением дополнительной информации из глобальной базы знаний DBpedia построены семантические профили интересов пользователей $\boldsymbol{Pr}_{U_1}, \ldots, \boldsymbol{Pr}_{U_e}$, как описано в разделе 4.2. При этом каждому правилу в профилях поставлено в соответствие значение метрики оценки «силы» причинной связи $\mu(P_k^i, \omega_k)$.

С помощью алгоритма выработки рекомендаций необходимо предсказать оценку (рейтинг) ω_{U_t, I_t}, которую целевой пользователь U_t из множества пользователей $U_1, \ldots, U_e$ поставит новому для него продукту I_t на основе оценок $\omega_1, \ldots, \omega_e$, которые поставили продукту I_t остальные пользователи из множества $U_1, \ldots, U_e$.

4.3.2.1 Кластеризация пользователей

Для повышения эффективности процедуры выработки рекомендаций необходимо выполнить предварительную кластеризацию пользователей на основе сходства их интересов. Такие кластеры могут быть пересекающимися, то есть один пользователь может принадлежать к нескольким кластерам. Это отражает реальное положение вещей. Действительно, один и тот же пользователь может, например, одинаково сильно интересоваться комедийными фильмами и музыкой жанра рок. В этом случае его можно отнести к группам любителей комедий и любителей рок-музыки.

После выполнения предварительной кластеризации пользователей для вычисления оценки ω_{U_t,I_t} продукта I_t для пользователя U_t будут использованы только оценки $\omega_1,\ldots,\omega_c$, выставленные этому продукту пользователями U_1, ..., U_c, которые принадлежат тому же кластеру, что и пользователь U_t.

Отметим, что кластеризация пользователей является важным шагом в условиях больших данных, так как помогает снизить вычислительную сложность задачи подсчета рейтинга.

Прежде чем применять алгоритм кластеризации пользователей, необходимо нормировать их профили. Выполняется это следующим образом. Для каждого пользователя U_1, ..., U_e (1) его интересы, представленные предикатами P_k^i правил из узлов дерева решений нижнего уровня (для набора данных Amazon при ω_k = {1, 2, 3, 4, 5}), упорядочиваются по убыванию значения метрики причинной связи $\mu(P_k^i,\omega_k)$; (2) вычисляется сумма всех значений метрики для таких правил $\sum_i \mu(P_k^i,\omega_k)$; (3) последовательно вычисляется текущая сумма значений метрики по порядку следования интересов и (4) пока текущая сумма не достигла значения равного $0.8\times\sum_i \mu(P_k^i,\omega_k)$, текущие интересы добавляются в итоговый нормированный профиль, все последующие – отсекаются.

На основе нормированного профиля пользователя для каждой пары пользователей из U_1, ..., U_e можно рассчитать значение меры сходства их интересов. В качестве меры сходства пары пользователей используется мера Танимото, которая имеет следующий вид:

$$Sim'(U_k,U_l) = \frac{\left|\boldsymbol{Pr}'_{U_k} \cap \boldsymbol{Pr}'_{U_l}\right|}{\left|\boldsymbol{Pr}'_{U_k}\right| + \left|\boldsymbol{Pr}'_{U_l}\right|} \qquad (4.15)$$

где $\left|\boldsymbol{Pr}'_{U_k}\right|, \left|\boldsymbol{Pr}'_{U_l}\right|$ – количество интересов в нормированных профилях пользователей U_k и U_l соответственно; $\left|\boldsymbol{Pr}'_{U_k} \cap \boldsymbol{Pr}'_{U_l}\right|$ – количество общих интересов пользователей U_k и U_l. При этом интерес считается общим для пользователей U_k и U_l в том случае, когда у обоих пользователей в нормированном профиле есть правила, пре-

дикаты которых построены для одного и того же атрибута[8], и эти правила имеют в заключении одну и ту же метку класса.

Опишем теперь алгоритм кластеризации пользователей на основе нормированного профиля. Входными параметрами алгоритма являются: (1) множество нормированных профилей $\boldsymbol{Pr}'_{U_1}, \ldots, \boldsymbol{Pr}'_{U_e}$ пользователей $U_1, \ldots, U_e$, (2) матрица значений метрики сходства (4.15) для каждой пары пользователей $\mathbf{Sim}' = \|Sim'(U_k, U_l)\|_{k,l}$, (3) два значения $\Delta_{\max}$ и $\Delta_{\min}$ для порога отсечения, используемых при построении кластеров (другая стратегия – это вычислять эти значения на каждом шаге алгоритма, используя среднее значение меры сходства, и некоторые коэффициенты), (4) M_{min} – минимально допустимая мощность кластеров, которые будут формироваться.

1. Пусть $j = 1$, где j – счетчик итераций.

2. Если $\left|\boldsymbol{U}^j\right| > M_{\min}$, где $\left|\boldsymbol{U}^j\right|$ - это мощность множества пользователей на шаге j, то переход к выполнению п. 3, иначе – к п. 10 (если $j = 1$, то $\boldsymbol{U}^j = \{U_1, \ldots, U_e\}$).

3. Из множества $\boldsymbol{U}^j$ случайным образом выбрать некоторого пользователя U_j.

4. В матрице $\mathbf{Sim}' = \|Sim'(U_k, U_l)\|_{k,l}$ найти всех пользователей $U_i \in \boldsymbol{U}^j$, для которых $Sim'(U_i, U_j) \geq \Delta_{\min}$. Добавить их в новое множество $\boldsymbol{UC}^j$. Пользователя U_j также включить во множество $\boldsymbol{UC}^j$.

5. Извлечь из матрицы $\mathbf{Sim}' = \|Sim'(U_k, U_l)\|_{k,l}$ значения метрики сходства для всех пар пользователей из множества $\boldsymbol{UC}^j$.

6. С помощью алгоритма поиска связанных компонент (подобный подход описан в 4.2.4) и порога $\Delta_{\min}$ выделить в $\boldsymbol{UC}^j$ непересекающиеся кластеры поль-

---

[8] Если атрибут является номинальным или числовым, то необходимо дополнительное сравнение значений, которые входят в подмножество области истинности предиката. Для подсчета сходства двух понятий онтологии днных могут быть использованы графовые подходы, основанные на поиске кратчайшего пути между узлами [138]

зователей и обозначить их как $\boldsymbol{UC}_w^j$.

7. Оценить мощность $\left|\boldsymbol{UC}_w^j\right|$ каждого кластера $\boldsymbol{UC}_w^j$. Если $\left|\boldsymbol{UC}_w^j\right| > M_{\min}$, то этот кластер рассматривается как один из искомых кластеров пользователей. Иначе этот кластер отбрасывается.

8. Для каждого кластера, полученного в п. 7, строится множества пользователей, для которых парное сходство больше или равно $\Delta_{\max}$ («сильный» подкластер). Имена всех пользователей, которые вошли в «сильные» подкластеры, удаляются из списка $\boldsymbol{U}^j$, полагая, что в другие кластеры они, скорее всего, не будут входить, поскольку свое «сильное» сходство они уже проявили.

9. Полагается $\boldsymbol{U}^{j+1} = \boldsymbol{U}^j$, $j = j+1$. и выполняется переход к п. 2.

10. Конец алгоритма.

В приложении В приведен псевдокод всех алгоритма кластеризации пользователей на основе профилей их интересов.

4.3.2.2 Алгоритм вычисления рейтинга

Для вычисления оценки ω_{U_t,I_t} продукта I_t для пользователя U_t сначала необходимо выбрать всех пользователей U_i, которые находятся в одном или нескольких кластерах с целевым пользователем U_t. Только оценки этих пользователей (при их наличии) далее используются при подсчете значения оценки продукта I_t. Далее из этих пользователей выбираются те, которые имеют оценку для продукта I_t. Пусть эти пользователи образуют множество $\boldsymbol{U}^t$. Только оценки этих пользователей далее используются при подсчете значения оценки продукта I_t

В [30] предлагается три основных варианта вычисления рейтинга ω_{U_t,I_t}:

$$1.\ \omega_{U_t,I_t} = \frac{1}{N}\sum_{U_i \in \boldsymbol{U}^t} \omega_{U_i,I_t} \tag{4.16}$$

$$2.\ \omega_{U_t,I_t} = k\sum_{U_i \in \boldsymbol{U}^t} \left(sim\left(U_t,U_i\right)\times\left(\omega_{U_i,I_t}\right)\right) \tag{4.17}$$

$$3.\ \omega_{U_t,I_t} = \hat{\omega}_{U_t} + k \sum_{U_i \in \boldsymbol{U}^t} \left(sim(U_t, U_i) \times \left(\omega_{U_t,I_t} - \hat{\omega}_{U_i} \right) \right) \tag{4.18}$$

где N – мощность множества $\boldsymbol{U}^t$; $\hat{\omega}_{U_t}$ – средний рейтинг целевого пользователя, $\hat{\omega}_{U_i}$ - средний рейтинг пользователя U_i; $sim(U_t, U_i)$ – метрика сходства пользователей U_t и U_i, которая будет пояснена далее; k – это коэффициент нормализации, который вычисляется по следующей формуле [30]:

$$k = \frac{1}{\sum_{U_i \in \boldsymbol{U}^t} |sim(U_t, U_i)|} \tag{4.19}$$

В самом простом варианте вычисления рейтинга продукт, который соответствует случаю (4.16), значение ω_{U_t,I_t} может быть вычислено как среднее всех значений ω_{U_i,I_t} для пользователей $U_i \in \boldsymbol{U}^t$. Очевидно, что эта формула не позволяет учитывать семантическое сходство между пользователями.

Выражение (4.17) позволяет вычислить значение ω_{U_t,I_t} как взвешенное среднее значений ω_{U_i,I_t}. При этом в качестве веса выступает значение меры сходства $sim(U_t, U_i)$ между пользователями U_t и U_i: чем больше значение $sim(U_t, U_i)$, тем больший вклад в значение ω_{U_t,I_t} вносит рейтинг ω_{U_i,I_t} пользователя U_i. Следует отметить важный недостаток этого подхода: различные пользователи могут поразному (субъективно) использовать шкалу рейтингов. Для некоторого пользователя оценка 4 из 5 может означать, что ему очень понравился продукт, а для другого 4 – это посредственная оценка продукта. Преодолеть описанный недостаток позволяет формула (4.18), которая использует скорректированное взвешенное среднее значение рейтингов: вместо абсолютного значения рейтинга ω_{U_i,I_t} пользователя U_i при подсчете среднего значения рейтинга используется его отклонение от среднего значения рейтинга $\hat{\omega}_{U_i}$ пользователя U_i, т.е. $\omega_{U_t,I_t} - \hat{\omega}_{U_i}$.

4.3.2.3 Семантическая метрика сходства пользователей

Опишем процедуру вычисления разработанной семантической метрики сходства $sim(U_t,U_i)$ пользователей U_t и U_i.

В первую очередь необходимо сравнить значения атрибутов целевого продукта I_t и интересы профиля целевого пользователя U_t для каждого класса. В профиле целевого пользователя U_t оставляются только те интересы с присвоенными им значениями меры причинности, которые встречаются в описании целевого продукта I_t, то есть те правила, которые «сработают» на этом продукте. Обозначим полученную модель профиля пользователя целевого пользователя U_t символом $\boldsymbol{Pr}_{U_t}^{I_t}$.

В профиле каждого пользователя $U_i \in \boldsymbol{U}^t$ остаются только те правила, которые срабатывают на целевом продукте I_t и имеют в заключении метку класса ω_{U_i,I_t}, т.е. значение рейтинга, которое пользователь U_i присвоил целевому продукту I_t. Обозначим полученную модель профиля пользователя символом $\boldsymbol{Pr}_{U_t}^{I_t}(\omega_{U_i,I_t})$.

Далее для каждого пользователя $U_i \in \boldsymbol{U}^t$ с непустым профилем $\boldsymbol{Pr}_{U_t}^{I_t}(\omega_{U_i,I_t})$ вычисляется расстояние до целевого пользователя U_t, причем в каждом классе целевой пользователь U_t представлен различными множествами интересов по отношению к целевому продукту I_t. Другими словами, расстояние до пользователя U_i вычисляется только по подмножеству интересов $\boldsymbol{Pr}_{U_t}^{I_t}(\omega_{U_i,I_t})$ целевого пользователя U_t в том классе, который соответствует рейтингу ω_{U_i,I_t}, выставленному целевому продукту I_t пользователем U_i.

Расстояние между пользователями U_i и U_t в классе ω_{U_i,I_t} относительно некоторого целевого продукта I_t предлагается вычислять следующим образом:

$$D^{\omega_{U_i,I_t}}(U_t,U_i) = \frac{\sum\limits_{s\in \boldsymbol{S}^{U_t\cap U_i}} \left|\mu_s^{U_t} - \mu_s^{U_i}\right| + \sum\limits_{s\in \boldsymbol{S}^{U_i\setminus U_t}} \left|\mu_s^{U_t}\right| + \sum\limits_{s\in \boldsymbol{S}^{U_t\setminus U_i}} \left|\mu_s^{U_i}\right|}{\left|\boldsymbol{Pr}_{U_t}^{I_t}(\omega_{U_i,I_t})\right| + \left|\boldsymbol{Pr}_{U_i}^{I_t}(\omega_{U_i,I_t})\right| - \left|\boldsymbol{S}^{U_t\cap U_i}\right|} \qquad (4.20)$$

где $\left|\boldsymbol{Pr}_{U_t}^{I_t}\left(\omega_{U_i,I_t}\right)\right|$, $\left|\boldsymbol{Pr}_{U_i}^{I_t}\left(\omega_{U_i,I_t}\right)\right|$ – количество интересов для класса ω_{U_i,I_t} в сокращенных профилях пользователей U_t и U_i; $\boldsymbol{S}^{U_t\cap U_i} = \boldsymbol{Pr}_{U_t}^{I_t}\left(\omega_{U_i,I_t}\right)\cap \boldsymbol{Pr}_{U_i}^{I_t}\left(\omega_{U_i,I_t}\right)$ – множество общих интересов в сокращенных профилях пользователей $\underline{U_t}$ и U_i для класса ω_{U_i,I_t}; $\boldsymbol{S}^{U_i\setminus U_t} = \boldsymbol{Pr}_{U_i}^{I_t}\left(\omega_{U_i,I_t}\right)\setminus \boldsymbol{Pr}_{U_t}^{I_t}\left(\omega_{U_i,I_t}\right)$ – множество интересов в редуцированном профиле пользователя U_i в классе ω_{U_i,I_t}, которых (интересов) нет у U_t; $\sum_{s\in \boldsymbol{S}^{U_i\setminus U_t}}\left|\mu_s^{U_i}\right|$, – сумма модулей значений меры причинной связи таких интересов; $\boldsymbol{S}^{U_t\setminus U_i} = \boldsymbol{Pr}_{U_t}^{I_t}\left(\omega_{U_i,I_t}\right)/\boldsymbol{Pr}_{U_i}^{I_t}\left(\omega_{U_i,I_t}\right)$, $\sum_{s\in \boldsymbol{S}^{U_t\setminus U_i}}\left|\mu_s^{U_t}\right|$ – аналогично для пользователя U_t; $\sum_{s\in \boldsymbol{S}^{U_t\cap U_i}}\left|\mu_s^{U_t}-\mu_s^{U_i}\right|$ – сумма модулей разностей значений меры причинной связи для общих интересов пользователей U_i и U_t в классе ω_{U_i,I_t}.

Соответственно, значение меры близости пользователей U_i и U_t относительно продукта I_t будет равно

$$sim^{I_t}(U_t,U_i)=1-D^{\omega_{U_i,I_t}}(U_t,U_i) \tag{4.21}$$

Результаты экспериментального исследования алгоритма выработки рекомендации методом коллаборативной фильтрации на основе семантического профиля пользователя с использованием различных вариантов подсчета рейтингов на наборе данных Amazon представлено в разделе 4.5.

4.3.4 Кросс-доменные рекомендации

При решении задач из области интеллектуальной обработки больших данных практически всегда приходится работать с информацией из различных предметных областей. Современные рекомендательные системы третьего поколения не являются исключением: они должны быть способны давать пользователям кросс-доменные рекомендации.

Анализ современного состояния разработок в области кросс-доменных рекомендательных систем выявил следующее [139]:

– это направление в области рекомендательных систем развито очень слабо;

– до сих пор существуют противоречащие друг другу определения понятия кросс-доменной рекомендации;

– на данный момент нет исчерпывающего сравнения существующих подходов к кросс-доменным рекомендациям.

Рассмотрим общую постановку задачи выработки кросс-доменных рекомендаций.

Пусть $\boldsymbol{U^1}$ и $\boldsymbol{U^2}$ – это множества пользователей, которые имеют оценки для продуктов из множеств (доменов) $\boldsymbol{D^1}$ и $\boldsymbol{D^2}$. Общая постановка задачи кросс-доменных рекомендаций подразумевает выработку рекомендаций для пользователей из множества $\boldsymbol{U^1 \cup U^2}$ относительно продуктов из множества $\boldsymbol{D^1 \cup D^2}$. Различия в частных постановках задачи зависят от того, являются ли множества $\boldsymbol{U^1 \cap U^2}$ и $\boldsymbol{D^1 \cap D^2}$ пустыми.

Задача выработки кросс-доменных рекомендаций обычно решается с помощью коллаборативной фильтрации, если хотя бы одно из множеств - $\boldsymbol{U^1 \cap U^2}$ или $\boldsymbol{D^1 \cap D^2}$ - не является пустым и обладает достаточной размерностью. В противном случае точность предсказания рекомендаций, как правило, будет очень низкой.

Далее рассматривается частная постановка задачи выработки кросс-доменных рекомендаций на основе семантических моделей и подход к ее решению на основе семантического профиля интересов пользователя с использованием метода коллаборативной фильтрации.

Напомним, что подход к построению онтологии данных с помощью семантического анализа понятий (подразделы 3.2.3 – 3.2.5) и семантического профиля пользователя, основанного на этой онтологии, позволяет представить его интересы из разных предметных областей в рамках одной семантической модели.

Пусть для множества пользователей $U_1, \ldots, U_n$ рекомендательной системы (например, пользователей, чьи оценки присутствуют в наборе данных Amazon) построены семантические профили интересов $\boldsymbol{Pr}_{U_1}, \ldots, \boldsymbol{Pr}_{U_n}$, которые основаны на информации об оцененных ими продуктах в нескольких доменах $\boldsymbol{D^1}, \ldots, \boldsymbol{D^m}$ (например, Музыка, Книги, Видео, как в наборе данных Amazon) с привлечением

дополнительной информации из глобальной базы знаний DBpedia. Соответствующий подход описан в разделе 4.2.

Пусть некоторый целевой пользователь U_t из множества $U_1, \ldots, U_n$ оценивал продукты, как минимум, в одном из доменов $D^1, \ldots, D^m$. Пусть некоторый новый для пользователя U_t целевой продукт I_t принадлежит домену $\boldsymbol{D}^t$ из множества $\boldsymbol{D}^1, \ldots, \boldsymbol{D}^m$, о котором профиль $\boldsymbol{Pr}_{U_t}$ пользователя U_t информации не содержит.

Необходимо предсказать оценку (рейтинг) ω_{U_t,I_t}, которую целевой пользователь U_t поставит новому для него продукту I_t на основе оценок, которые поставили продукту I_t остальные пользователи из множества $U_1, \ldots, U_n$.

Подчеркнем, что в рамках рассматриваемой модели такая задача имеет решение только тогда, когда целевой пользователь имеет схожие интересы с некоторыми пользователями, которые имеют интересы в целевом домене. Продемонстрируем это на простом примере.

Если пользователи и домены взаимодействуют так, как показано в примере на рисунке 4.4, то с помощью гибридной коллаборативной фильтрации, описанной в подразделе 4.3.3, для целевого пользователя U_t с использованием профилей пользователей U_2, U_3, U_5 можно предсказать рейтинги для тех продуктов из доменов $\boldsymbol{D}^2$, $\boldsymbol{D}^3$, $\boldsymbol{D}^4$, которые были оценены пользователями U_2, U_3, U_5. В этом случае информация о профилях пользователей U_1, U_4 для рекомендаций использована быть не может. Рекомендации в домене $\boldsymbol{D}^1$ для пользователя U_t также получены быть не могут.

Описание экспериментального исследования алгоритма выработки кросс-доменных рекомендации методом коллаборативной фильтрации на основе семантического профиля пользователя на наборе данных Amazon и его результатов дано в разделе 4.5.

4.4 Программная реализация моделей и методов построения причинных моделей принятия решений в рекомендующих системах третьего поколения

Программный комплекс, реализующий построение онтологии данных с помощью семантического анализа понятий, алгоритмы построения семантического

профиля интересов пользователя и выработки рекомендаций различными способами, представляет собой консольное объектно-ориентированное приложение на языке Java. На рисунке 4.5 представлена UML-диаграмма основных его компонент, реализующих алгоритм построения онтологии данных путем семантического анализа понятий.

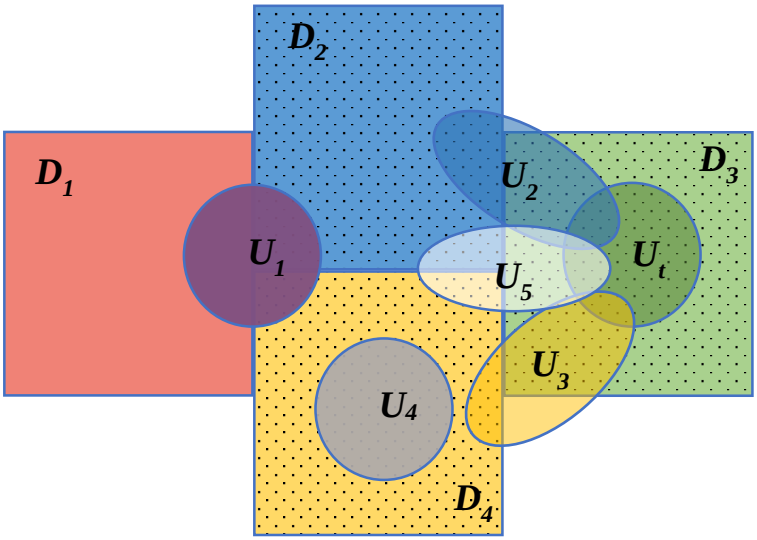


Рисунок 4.4 – Графическая интерпретация примера взаимодействия пользователей U_1, U_2, U_3, U_4, U_5, U_s доменов $\boldsymbol{D_1}$, $\boldsymbol{D_2}$, $\boldsymbol{D_3}$, $\boldsymbol{D_4}$. Целевой пользователь U_t содержит в профиль только информацию о домене $\boldsymbol{D_3}$; U_1 - о доменах $\boldsymbol{D_1}$, $\boldsymbol{D_2}$, $\boldsymbol{D_3}$; U_2 - домнах $\boldsymbol{D_2}$, $\boldsymbol{D_3}$; U_3 - о доменах $\boldsymbol{D_3}$, $\boldsymbol{D_4}$; U_4 – о доменах $\boldsymbol{D_4}$; U_5 – о доменах $\boldsymbol{D_2}$, $\boldsymbol{D_3}$, $\boldsymbol{D_4}$

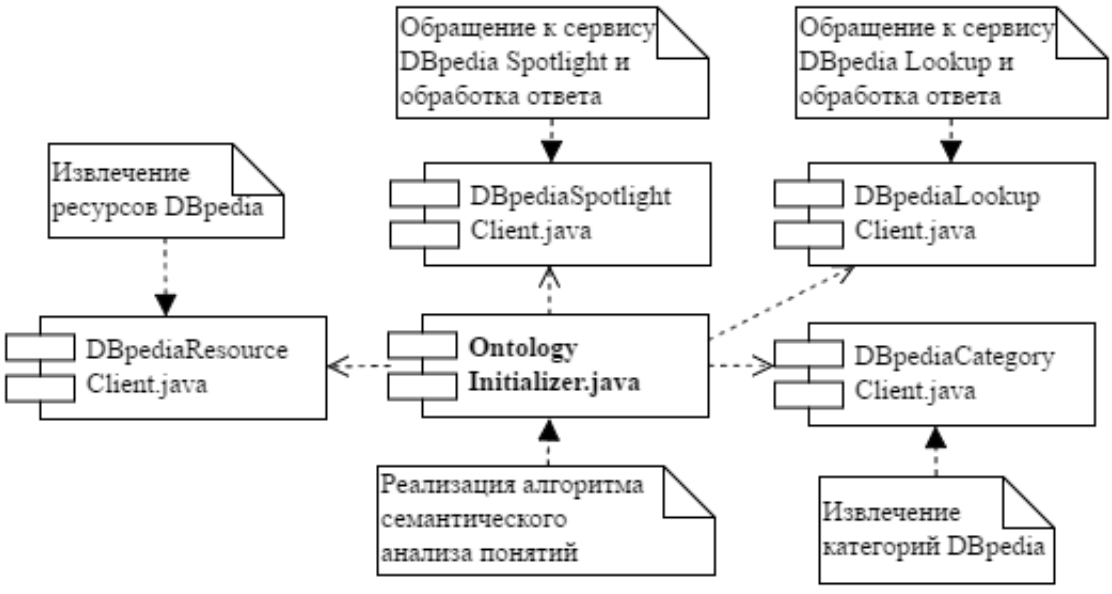


Рисунок 4.5 –UML-диаграмма основных компонент, реализующих алгоритм построения онтологии данных путем семантического анализа понятий

Обогащение информации и извлечение данных из DBpedia выполняется классами *DBpediaLookupClient.java*, *DBpediaSpotlightServiceClient.java*, *DBpediaResourceClient.java*, *DBpediaCategoryClient.java*. Алгоритм семантического анализа понятий (псевдокод представлен на рисунке 4.3) реализован классом *OntologyInitializer.java*.

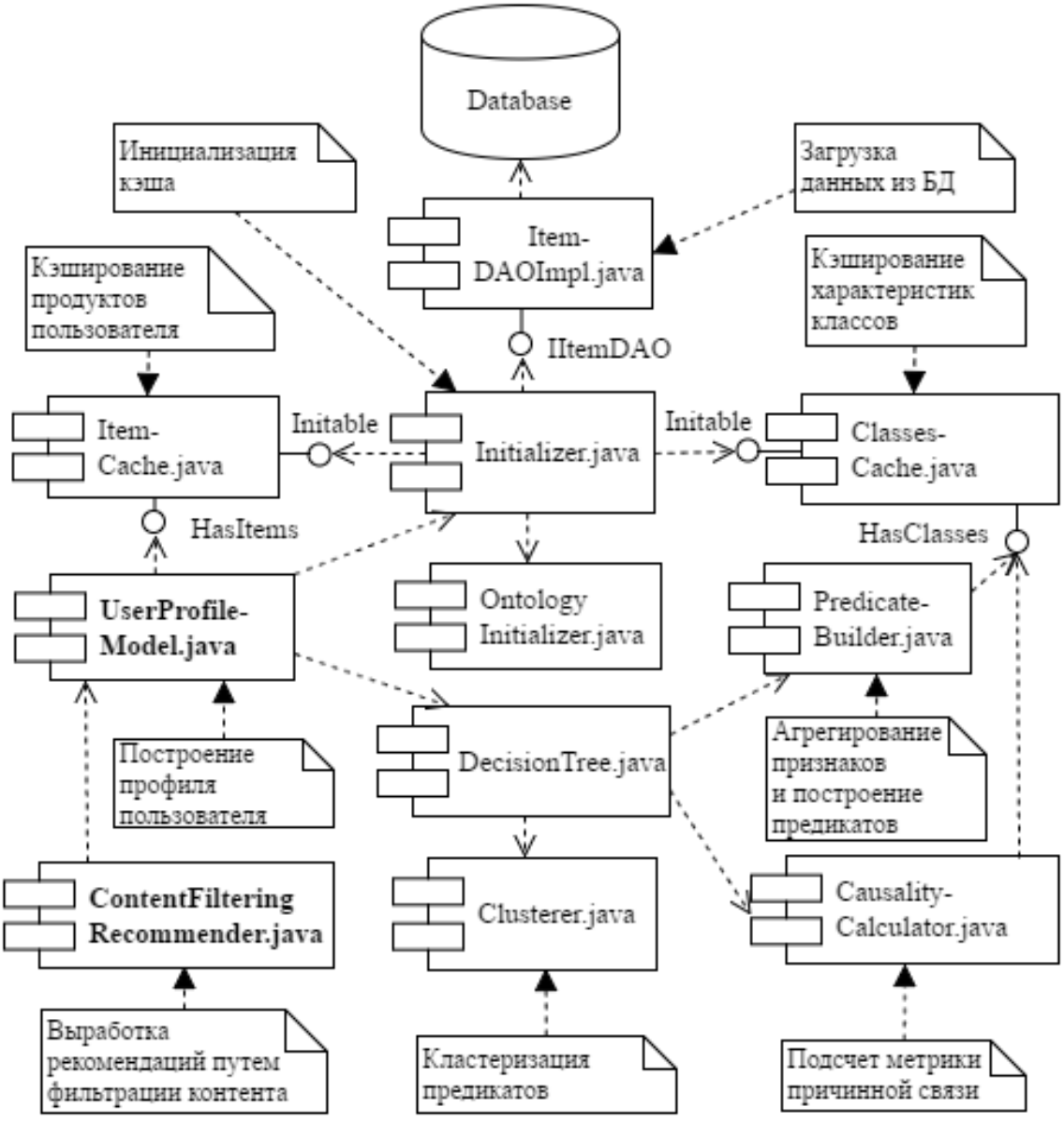


Рисунок 4.6 –UML-диаграмма основных компонент программного продукта, реализующего алгоритмы построения семантического профиля интересов пользователя и выработки рекомендаций путем фильтрации контента

На рисунке 4.6 представлена UML-диаграмма основных компонент программного продукта, реализующего алгоритмы построения семантического профиля интересов пользователя и выработки рекомендаций путем фильтрации кон-

тента. Основные шаги построения семантического профиля пользователя реализованы в классе *UserProfileModel.java* (псевдокод представлен на рисунке В.1 в приложении В). На первом этапе работы программы выполняется загрузка данных пользователя из базы данных. За этот процесс отвечает класс *ItemDAOImpl.java*, реализующий интерфейс *ItemDAO*. Полученные данные пользователя кэшируются классами *ItemCache.java* и *ClasssesCache.java*, реализующими интерфейс *Initable*. Инициализация кэшей и запуск загрузки данных пользователя выполняется классом *Initializer.java*. Это класс также запускает процесс инициализации онтологии интересов (класс *OntologyInitializer.java*). Логика построения дерева решений профиля инкапсулирована в классе *DecisionTree.java*. Агрегирование значений атрибутов, формирование предикатов и построения ассоциативно-причинных правил выполняет класс PredicateBuilder.java (псевдокод представлен на рисунке В.3 в приложении В). За вычисление метрики причинной связи и значений корреляции отвечает класс *CausalityCalculator.java*. Причинная фильтрация правил выполняется в классе *DecisionTree.java* (псевдокод представлен на рисунке В.2 в приложении В). Кластеризация предикатов выполняется в классе *Clusterer.java* (псевдокод представлен на рисунке В.4 в приложении В). Извлечение минимального набора правил из кластера выполняется в классе *DecisionTree.java*.

Выработка рекомендаций методом фильтрации контента на основе профиля пользователя выполняется в классе *ContentFilteringRcommender.java* (псевдокод представлен на рисунке В.5 в приложении В). На рисунке 4.7 представлена UML-диаграмма основных компонент, реализующих алгоритмы выработки рекомендаций путем гибридной коллаборативной фильтрации. Класс *UserClustering.java* реализует алгоритм предварительной кластеризации пользователей на основе семантических профилей их интересов (псевдокод представлен на рисунках В.6, В.7 в приложении В). Класс *CollaborativeRatingPredictor.java* реализует механизм подсчета коллаборативного рейтинга с помощью метрики сходства пользователей (класс *SimilarityCalculator.java*). Класс *CollaborativeFilteringRecommender.java* реализует механизм выработки коллаборативных рекомендаций.

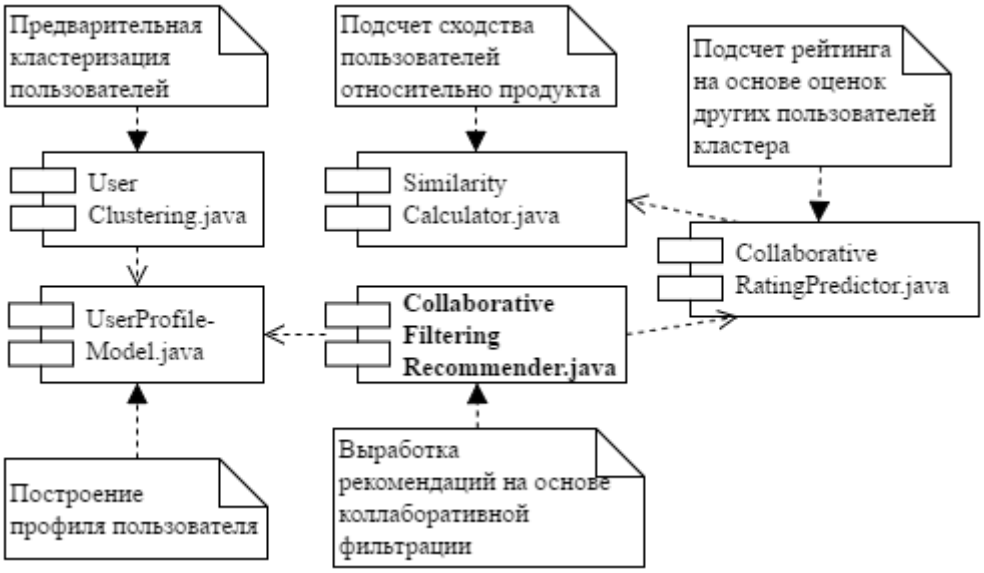


Рисунок 4.7 – UML-диаграмма основных компонентов, реализующих алгоритмы выработки рекомендаций путем коллаборативной фильтрации

На рисунке 4.8 представлена UML-диаграмма основных компонент программного продукта, реализующего алгоритмы выработки кросс-доменных рекомендаций.

Класс *CrossDomainRecommender.java* реализует выработку кросс-доменных рекомендаций с помощью алгоритмов коллаборативной фильтрации в тех случаях, когда это возможно.

4.5 Экспериментальные оценки точности разработанных алгоритмов в задачах построения рекомендательных систем третьего поколения

Прежде чем приступить, к описанию экспериментов, необходимо сделать несколько важных замечаний.

Алгоритмы выработки рекомендаций и рекомендательные системы в целом (как и любые другие системы принятия решений) обладают рядом свойств, которые могут быть экспериментально исследованы и оценены. К таким свойствам относят точность, робастность, масштабируемость, уровень значимости рекомендаций (англ. *confidence*), уровень покрытия (англ. *coverage*), степень новизны рекомендаций (англ. *novelty*), адаптивность системы и многие другие [140].

В общем случае, прежде чем приступать к экспериментальному исследова-

нию алгоритмов, следует выделить те свойства, которые более важны для прикладного приложения, в котором эти алгоритмы будут использованы.

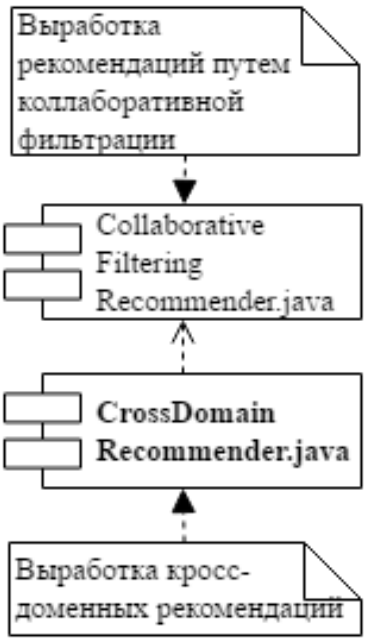


Рисунок 4.8 –UML-диаграмма основных компонент программного продукта, реализующего алгоритмы выработки кросс-доменных рекомендаций

Обычно, при выборе того или иного алгоритма выработки рекомендаций в первую очередь учитывается точность его работы, то есть способность хорошо предсказывать выбор пользователя. Тем не менее, многие исследователи сходятся во мнении, что точные прогнозы являются необходимым, но не всегда достаточным условием разработки эффективной рекомендательной системы [140].

В данной работе в рамках экспериментального исследования будет выполнена только оценка точности алгоритмов выработки рекомендаций.

Наиболее распространённой метрикой оценки точности любых алгоритмов принятия решений, в том числе алгоритмов классификации, является средняя квадратическая ошибка (англ. *root mean square error*, RMSE) [140]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\omega_i - \ddot{\omega}_i\right)^2} \qquad (4.22)$$

где n – общее число классифицированных экземпляров; ω_i – подлинный класс экземпляра, $\ddot{\omega}_i$ – предсказанный класс экземпляра (англ. *confusion matrix*)..

Еще одним распространенным способом оценки точности системы принятия решений является анализ матрицы неточностей.

Экспериментальное исследование точности разработанных алгоритмов выработки рекомендаций проводилось на наборе данных Amazon. Основные характеристики этого набора приведены в разделе 3.3.

В тестировании всех алгоритмов участвовали только те пользователи, которые имеют минимум 15 отзывов на различные продукты.

Для каждого пользователя тестирование выполнялось с помощью процедуры скользящего контроля с четырьмя блоками (англ. *4-fold cross-validation*) [41] следующим образом: все продукты, оцененные пользователем, разбивались на четыре блока, каждый из блоков поочередно выступал в качестве контрольной выборки, а остальные три – в качестве обучающей.

На основе обучающей выборки для каждого пользователя выполнялось построение онтологии интересов с помощью семантического анализа понятий (подраздел 3.2.5) и производилось обучение профиля, как описано в разделе 4.2. Продукты из контрольной выборки выступали в качестве новых для исследуемого пользователя.

Результаты такого тестирования для каждого пользователя могут быть представлены в виде матрице неточностей, построенной для пяти классов (рейтингов) – $\omega = \{1, 2, 3, 4, 5\}$ (рисунок 4.9а). Если следовать предположению о том, что оценки «4» и «5» - положительные, а «1», «2», «3» - отрицательные, то дополнительно может быть построена матрица для двух классов – «положительного» и «отрицательного» – $\omega = \{(1, 2, 3), (4, 5)\}$ (рисунок 4.9а). На рисунке 4.9б «TP» – это количество правильно классифицированных экземпляров «положительного» класса (англ. *true positive*); «TN» – это количество правильно классифицированных экземпляров «отрицательного» класса (англ. *true negative*); «FP» – это количество экземпляров «отрицательного» класса, классифицированных как экземпляры «положительного» класса, иначе – «ложная тревога» или ошибка первого рода (англ. *false positive*); «FN» – это количество экземпляров «положительного»

класса, классифицированных как экземпляры «отрицательного» класса, иначе – «пропуск сигнала» или ошибка второго рода (англ. *false negative*).

а)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	…	…	…	…	…
2	…	…	…	…	…
3	…	…	…	…	…
4	…	…	…	…	…
5	…	…	…	…	…

б)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	TP	FN
1,2,3	FP	TN

Рисунок 4.9 – Структуры матриц неточностей для а) классов $\omega = \{1, 2, 3, 4, 5\}$ и б) классов $\omega = \{(1, 2, 3), (4, 5)\}$

На основе обеих матриц неточностей для каждого пользователя могут быть подсчитаны значения среднеквадратического отклонения (RMSE) для случаев, когда рейтинг имеет пять значений и два значения. Могут быть вычислены также и другие показатели эффективности классификатора, которые тоже удобно представлять в форме матрицы. На рисунке 4.10 приняты следующие обозначения:

$TPR = \frac{TP}{TP + FN}$ – чувствительность классификатора (англ. *true positive rate, sensitivity*);

$TNR = \frac{TN}{TN + FP}$ – специфичность классификатора (англ. *true negative rate, specificity*);

$FPR = \frac{FP}{TN + FP}$ – показатель, отражающий уровень значимости классификатора (англ. *false positive rate, fall-out*);

$FNR = \frac{FN}{TP + FN}$ – коэффициент пропусков классификатора (англ. *false negative rate, miss rate*).

Отметим, что для задач выработки рекомендаций наиболее важными показателями точности алгоритмов являются показатели, RMSE, TPR и FPR. Показатели

TPR и FPR являются более важными, чем FNR и TNR, поскольку в рекомендательной системе необходимо максимально снизить количество «ложных тревог». Ложная тревога в данном контексте соответствует рекомендации пользователю того продукта, который с большой долей вероятности ему не понравится. Большой процент ложных тревог ведет к снижению уровня доверия пользователя к рекомендательной системе. В этом случае «пропуск сигнала», то есть «не рекомендация» какого-либо товара, который мог бы понравится пользователю, не является критическим.

		Предсказанный класс	
		4,5	**1,2,3**
Подлинный класс	**4,5**	**TPR**	**FNR**
	1,2,3	**FPR**	**TNR**

Рисунок 4.10 – Структура представления показателей эффективности классификаторов в форме матрицы неточностей

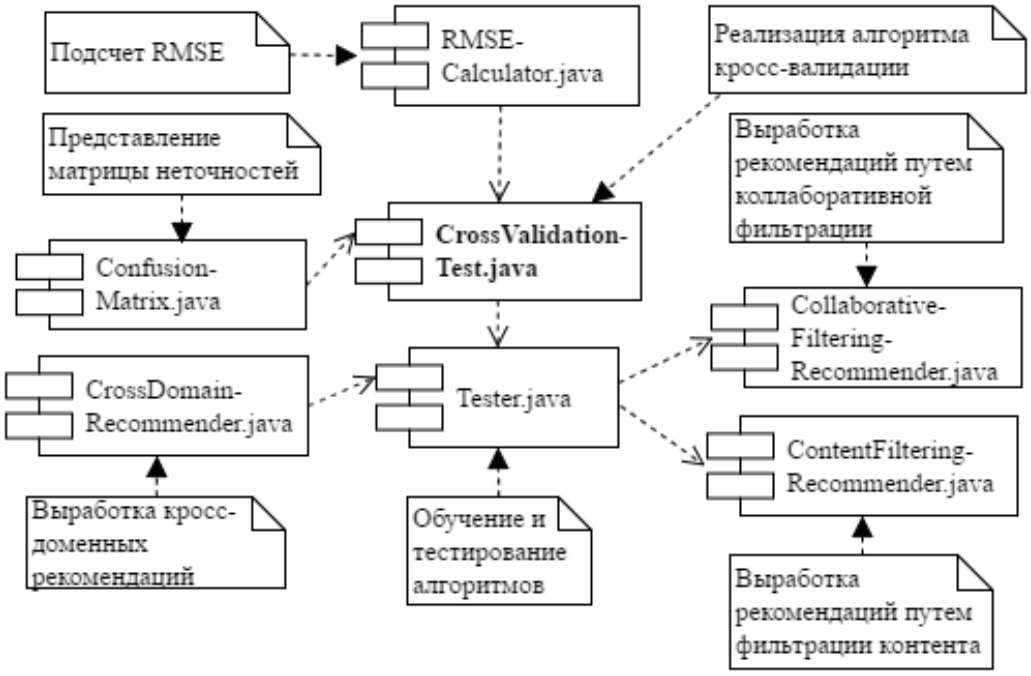


Рисунок 4.11 – UML-диаграмма основных компонент программного продукта, реализующего тестирование разработанных алгоритмов с помощью процедуры скользящего контроля

На рисунке 4.11 представлена UML-диаграмма основных компонент программного продукта, реализующего тестирование разработанных алгоритмов с помощью процедуры скользящего контроля.

По результатам анализа значений описанных показателей могут быть сделаны основные выводы о точности построенных классификаторов и разработанных алгоритмов. Они формулируются далее.

4.5.1 Экспериментальные оценки точности и вычислительной эффективности разработанных алгоритмов выработки рекомендаций методом фильтрации контента

Для оценки алгоритма выработки рекомендаций методом фильтрации контента с помощью скользящего контроля было случайным образом выбрано 1000 пользователей, которые имеют минимум 15 отзывов на различные продукты. В ходе экспериментов исследовались две метрики оценки причинной связи (используются на этапе обучения профиля пользователя, подраздел 4.2.3) – *коэффициент регрессии* и *мера Клозгена*.

По результатам тестирования для каждой метрики была построена сводная матрица неточностей для пяти классов (рисунки 4.12а, 4.13а). На основе построенной матрицы было вычислено значение RMSE для пяти классов. Наконец, на основе матрицы неточностей для пяти классов была построена сводная матрица неточностей для двух классов (рисунки 4.12б, 4.13б), по которым посчитаны значения RMSE для двух классов, а также остальные показатели эффективности классификатора (рисунки 4.12в, 4.13в). Основные характеристик и результаты экспериментов приведены в таблице 4.1.

Исследована также точность алгоритма выработки рекомендаций методом фильтрации контента без обогащения информации о продуктах пользователя и построения онтологии. Результаты такого эксперимента представлены на рисунке 4.14.

В таблице 4.2 приведена зависимость времени построения профиля пользователя в зависимости от количества оцененных продуктов и их атрибутов.

Таблица 4.1 - Характеристики и результаты экспериментов по оценке точности алгоритмов выработки рекомендаций методом фильтрации контента (*коэффициент регрессии*)

Название характеристики	**Значение**
Всего пользователей	1000
Всего продуктов, по которым имеются оценки	80486
Коэффициент регрессии с построением онтологии данных	
Всего продуктов, для которых не удалось построить рекомендации	1609
Среднее значение RMSE для пяти классов	0.96
Среднее знач ение RMSE для двух классов	0.35
Коэффициент регрессии без построения онтологии данных	
Всего продуктов, для которых не удалось построить рекомендации	2806
Среднее значение RMSE для пяти классов	0.99
Среднее значение RMSE для двух классов	0.36
Мера Клозгена	
Всего продуктов, для которых не удалось построить рекомендации	11268
Среднее значение RMSE для пяти классов	1.04
Среднее значение RMSE для двух классов	0.41

Таблица 4.2 - Зависимость времени построения профиля пользователя от количества оцененных продуктов и их атрибутов

Количество атрибутов	**Количество экземпляров (продуктов)**	**Время обработки, мсек**	**Ln времени обработки**
1	**3**	**3**	**4**
79	16	93	4,53
194	22	54	3,99
179	27	60	4,09
140	28	58	4,06
135	30	21	3,04
192	31	53	3,97
59	32	13	2,56
602	41	1202	7,09
547	43	661	6,49
187	43	100	4,61
140	47	30	3,40
494	58	788	6,67
1	2	3	4
1067	63	4913	8,50
759	63	1507	7,32
509	75	1470	7,29
581	80	1334	7,20
235	120	238	5,47
1207	451	27233	10,21

а)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	3313	375	1013	713	187
2	321	3575	1500	763	779
3	191	211	9348	1321	663
4	638	689	3163	15575	950
5	1250	1288	3113	7855	20083

б)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	47978	8244
1,2,3	3678	18977

в)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	0.86	0.14
1,2,3	0.14	0.84

Рисунок 4.12 - Сводные матрицы неточностей для 5 (а) и 2 (б) классов и матрица показателей эффективности для двух классов (в) (*коэффициент регрессии*)

Анализ результатов экспериментов позволил сделать следующие выводы:

– более выраженная зависимость существует между временем обучения профиля пользователя и количеством атрибутов продуктов;

– использование при обучении профиля пользователя коэффициента регрессии, позволяет получить более низкое значение RMSE для пяти и для двух классов, по сравнению с мерой Клозгена. Значение RMSE для коэффициента регрессии, равное 0.96, означает, что алгоритм выработки рекомендаций в среднем ошибается на 0.96, то есть, если истинный класс продукта – «4», алгоритм может классифицировать его также и как класс «3», и как класс «5». Таким образом, классификатор ошибается не более, чем на один «соседний» класс;

– алгоритм обучения с использованием коэффициента регрессии имеет лучшие значения для показателя FPR, который играет важную роль в оценке точности рекомендательных систем;

– алгоритм выработки рекомендаций методом фильтрации контента с использованием коэффициента регрессии позволяет классифицировать более 98%

продуктов от их общего количества в тестовых выборках для 1000 пользователей, в то время как использование меры Клозгена снижает процент классифицированных продуктов до 86;

– на основании вышесказанного, следует выбрать коэффициент регрессии как основную меру оценки силы причинных связей при обучения профиля пользователя на наборе данных Amazon;

– без использования обогащения данных и построения онтологии с помощью семантического анализа понятий значение RMSE и основных показателей точности падает несильно, однако количество неклассифицированных продуктов увеличивается практически в два раза по сравнению с алгоритмом выработки рекомендаций методом фильтрации контента с использованием коэффициента регрессии и построением онтологии данных;

– продемонстрированная точность является хорошей для выборки такого размер (от 15 экземпляров).

Во всех последующих экспериментах будет использован вариант алгоритма с построением онтологии данных и коэффициентом регрессии в качестве метрики оценки причинной связи.

4.5.2 Экспериментальные оценки разработанных алгоритмов выработки рекомендаций методом гибридной коллаборативной фильтрации

Для оценки алгоритма выработки рекомендаций методом гибридной коллаборативной фильтрации было случайным образом выбрано 4000 пользователей, которые имеют минимум 15 отзывов на различные продукты. Проведена предварительная кластеризация пользователей. Тестирование проводилось для всех трех вариантов подсчета коллаборативного рейтинга, представленных формулами (4.16), (4.17), (4.18). Обучение профиля каждого пользователя проводилось на трех четвертях его продуктов, тестирование - на оставшейся одной четверти.

а)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	1493	504	688	791	298
2	317	3072	708	1138	297
3	99	151	6833	2732	527
4	185	444	1839	14011	7770
5	119	526	2811	10868	10997

б)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	42503	5857
1,2,3	5717	15141

в)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	0.89	0.11
1,2,3	0.28	0.72

Рисунок 4.13 – Сводные матрицы неточностей для пяти (а) и двух (б) классов и матрица показателей эффективности для двух классов (в) (*мера Клозгена*)

а)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	3642	391	869	594	286
2	282	3520	1354	772	598
3	151	216	8382	1497	799
4	583	693	3471	13840	1056
5	1321	1331	3392	8235	20405

б)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	43806	10058
1,2,3	4359	19447

в)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	0.81	0.19
1,2,3	0.18	0.82

Рисунок 4.14 – Сводные матрицы неточностей для 5 (а) и для 2 (б) классов и матрица показателей эффективности для 2 классов (в) (*без использования онтологии*)

Рисунок 4.15 – Зависимость времени построения профиля пользователя от количества оцененных продуктов

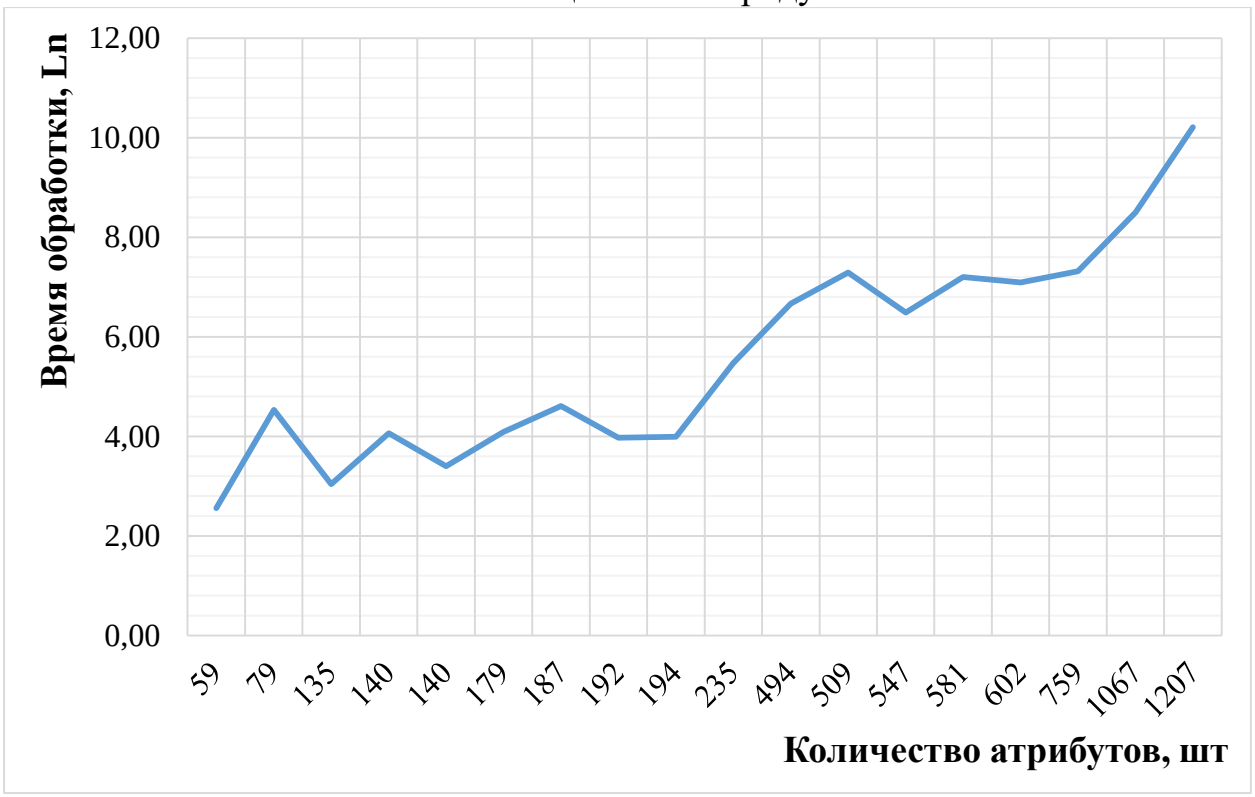


Рисунок 4.16 – Зависимость времени построения профиля пользователя от количества оцененных продуктов

Выходные значения формул (4.16), (4.17), (4.18) не являются целочисленными, то есть не соответствуют в точности меткам класса. Поэтому сперва для каждого варианта вычисления рейтинга посчитано RMSE для вещественных значений класса, то есть рейтинг, предсказанный формулами (4.16), (4.17), (4.18), учитывался без округления. Затем были построены сводные матрицы неточностей для пяти классов (рисунок 4.17). При этом предсказанный рейтинг ω, представленный в вещественный шкале, относился к некоторому классу по следующим правилам:

- если $\omega < 1.7$, то предсказанный класс «1»;
- если $1.7 \leq \omega < 2.7$, то предсказанный класс «2»;
- если $2.7 \leq \omega < 3.3$, то предсказанный класс «3»;
- если $3.3 \leq \omega < 4.3$, то предсказанный класс «4»;
- если $4.4 \leq \omega$, то предсказанный класс «5».

На основе построенных матриц были вычислены значения RMSE для пяти классов. Наконец, на основе матриц для пяти классов построены сводные матрицы неточностей для двух классов (рисунок 4.18), по которым посчитаны значения RMSE для двух классов, а также остальные показатели эффективности классификатора (рисунок 4 .19). Основные характеристики и результаты эксперимента приведены в таблице 4.3.

Анализ результатов экспериментов позволил сделать следующие выводы:

– использование при вычислении рейтинга формулы (4.18) позволяет добиться наибольшей точности по сравнению с формулами (4.16) и (4.17), однако, формула (4.18) позволяет вычислить рейтинг меньшего количество продуктов;

– несмотря на это, следует выбрать в качестве основной формулы вычисления рейтинга именно формулу (4.18), так как с увеличением размера выборки пользователей, количество продуктов, которым удастся вычислить рейтинг, будет возрастать;

– более низкая точность алгоритма выработки рекомендаций на основе коллаборативной фильтрации, по сравнению с алгоритмом на основе метода фильтрации контента, может быть связана с недостаточным количеством пользователей в выборке. При увеличении количества пользователей в выборке будет расти вероятность обнаружить больше пользователей со схожими интересами и, соответственно, использовать больше оценок таких пользователи при выработке рекомендаций.

Таблица 4.3 - Характеристики и результаты эксперимента по оценке алгоритмов выработки рекомендаций методом гибридной коллаборативной фильтрации

Название характеристики	**Значение**
Всего пользователей	4000
Удалось кластеризовать пользователей	3640
Всего продуктов оценили	293954
Всего продуктов в тестовых выборках	71882
Подсчет рейтинга по формуле (5.16)	
Удалось классифицировать	44506
Среднее значение RMSE для вещественных значений	1.00
Среднее значение RMSE для 5 классов	1.02
Среднее значение RMSE для 2 классов	0.43
Подсчет рейтинга по формуле (5.17)	
Удалось классифицировать	40499
Среднее значение RMSE для вещественных значений	0.99
Среднее значение RMSE для 5 классов	1.00
Среднее значение RMSE для 2 классов	0.42
Подсчет рейтинга по формуле (5.18)	
Удалось классифицировать	39904
Среднее значение RMSE для вещественных значений	0.96
Среднее значение RMSE для 5 классов	0.97
Среднее значение RMSE для 2 классов	0.39

а)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	1245	937	282	780	87
2	217	2740	449	451	405
3	233	943	2485	572	837
4	328	871	1732	7939	2475
5	113	370	1043	3742	13230

б)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	1458	998	267	636	101
2	199	2805	475	674	350
3	187	854	3121	473	735
4	154	1199	1434	8495	1869
5	119	314	958	1725	10899

в)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	1323	1027	274	630	87
2	213	2882	468	480	305
3	173	690	3032	571	537
4	142	868	1028	8228	2069
5	124	346	976	2501	10930

Рисунок 4.17 – Сводные матрицы неточностей для 5 классов: а) формула (4.16); б) формула (4.17); в) формула (4.18)

а)

Подлинный класс	Предсказанный класс:	**4,5**	**1,2,3**
	4,5	26701	5142
	1,2,3	3132	9531

б)

Подлинный класс	Предсказанный класс:	**4,5**	**1,2,3**
	4,5	22988	4178
	1,2,3	2969	10364

в)

Подлинный класс	Предсказанный класс:	**4,5**	**1,2,3**
	4,5	23728	3484
	1,2,3	2610	10082

Рисунок 4.18 – Сводные матрицы неточностей для 2 классов: а) формула (4.18); б) формула (4.19); в) формула (4.20)

4.6.2 Экспериментальные оценки разработанных алгоритмов кросс-доменных рекомендаций

Для оценки алгоритма выработки кросс-доменных рекомендаций методом гибридной коллаборативной фильтрации случайным образом выбрано 4000 пользователей, которые имеют более 15 отзывов на различные продукты минимум из двух доменов. Обучение профиля целевого пользователя проводилось только для продуктов из одного домена, а тестирование - на продуктах другого домена. Для подсчета рейтинга использовалась только формула (4.18). Рейтинг для продукта вычислялся только тогда, когда в выборке имеется минимум три пользователя для вычисления коллаборативного рейтинга. Порядок вычислений характеристик аналогичен описанному в предыдущем подразделе. Основные характеристики и результаты экспериментов приведены в таблице 4.4.

а)

Подлинный класс \ Предсказанный класс		**4,5**	**1,2,3**
	4,5	0.84	0.16
	1,2,3	0.25	0.75

б)

Подлинный класс \ Предсказанный класс		**4,5**	**1,2,3**
	4,5	0.85	0.15
	1,2,3	0.22	0.78

в)

Подлинный класс \ Предсказанный класс		**4,5**	**1,2,3**
	4,5	0.87	0.13
	1,2,3	0.20	0.80

Рисунок 4.19 – Сводные матрицы показателей эффективности для 2 классов: а) формула (4.16); б) формула (4.17); в) формула (4.18)

На рисунках 4.20 и 4.21 приведены сводные матрицы неточностей для 5 и 2 классов, а также матрица показателей эффективности классификаторов, для вариантов экспериментов с предварительной кластеризацией пользователей и без неё.

Таблица 4.4 - Характеристики и результаты эксперимента по оценке алгоритмов выработки кросс-доменных рекомендаций методом гибридной коллаборативной фильтрации

Название характеристики	**Значение**
Всего пользователей	4000
Всего продуктов в тестовых выборках	145588
Удалось классифицировать	45863
Среднее значение RMSE для вещественных значений	1.02
Среднее значение RMSE для 5 классов	1.01
Среднее значение RMSE для 2 классов	0.44

а)

Подлинный класс \ Предсказанный класс	1	2	3	4	5
1	1395	1044	234	605	167
2	406	2923	801	755	461
3	178	911	4002	798	879
4	167	1443	2400	9434	1951
5	135	365	858	2204	11347

б)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	24936	5368
1,2,3	3665	11894

в)

Подлинный класс \ Предсказанный класс	4,5	1,2,3
4,5	0.82	0.18
1,2,3	0.24	0.76

Рисунок 4.20 – Сводные матрицы неточностей для 5 (а) и 2 (б) классов и матрица показателей эффективности для 2 классов (в)

Анализ результатов эксперимента позволил сделать следующий вывод: разработанный алгоритм выработки кросс-доменных рекомендаций на основе коллаборативной фильтрации позволяет с необходимой точностью предсказывать рейтинги, которые некоторый пользователь присвоит продуктам из тех доменов, в которых он ранее не выставлял рейтингов. При этом количество продуктов, для которых удается выработать рекомендацию будет увеличиваться с увеличением размера выборки пользователей.

4.6 Выводы

В данной главе решены следующие задачи диссертационного исследования:

1. Предложен математически корректный, масштабируемый и эффективный алгоритм поиска причинных зависимостей между переменными, описывающими данные, а также выразительный способ совместного представления причинных связей.

Для поиска причинных зависимостей в данных сначала выполняется агрегирование данных. В ходе агрегирования атрибуты данных преобразуются к виду утверждений в форме унарных предикатов, т.е. к единой шкале измерения.

В общем случае эта процедура позволяет преодолеть трудности, связанные с большой размерностью задачи и гетерогенностью признаков. Построенные предикаты могут рассматриваться как компоненты нового редуцированного пространства признаков, а сформированные правила выступать в качестве простых правил классификации. При этом типы исходных атрибутов, значения которых подвергаются агрегированию, могут быть любыми: включать в себя тексты на естественном языке, номинальные, числовые или булевы данные, а также могут быть представлены понятиями онтологии, построенной в ходе семантического анализа понятий, тогда как все признаки редуцированного пространства описания данных будут булевыми утверждениями, представленными в форме одноместных предикатов. Построенное множество правил отражает ассоциативные связи в данных.

На следующем этапе из множества правил, построенного на этапе агрегирования, выделяются те правила, которые отражают причинные зависимости в данных. Причинная фильтрация правил выполняется с помощью метрики причинной связи, выбор которой обоснован в главе 2. Фильтрация, сохраняющая только самые «сильные» правила (с причинной точки зрения), позволяет значительно сократить размерность модели данных, улучшив тем самым вычислительную эффективность последующих процедур обучения и принятия решений без заметной потери точности. Полученные причинные правила могут быть представлены в виде бинарного дерева принятия решений, что облегчает их применение в задачах принятия решений.

2. Разработан масштабируемый алгоритм выбора множества атрибутов для решения задач классификации на основе модели причинных связей и семантики данных, представленной в терминах понятий онтологии данных.

С целью оптимизации множества признаков, которое далее используется для решения задач классификации, выполняется кластеризация причинных правил

классификации таким образом, чтобы в общие кластеры объединялись сильно коррелированные правила, а в разные кластеры попадали те, которые коррелированы слабо или вообще не коррелированы. В итоговом оптимизированном множестве правил каждый кластер представлен одним своим «самым сильным» (согласно сформулированным критериям) представителем.

Кластеризация правил позволяет удалить из итоговой модели принятия решений бесполезные правила – элементарные классификаторы, то есть те правила, использование которых не ведет к повышению точности алгоритма. Кроме того, их удаление позволяет повысить вычислительную эффективность работы системы в целом.

Решение описанных задач работы и демонстрация разработанных подходов и алгоритмов выполняется на примере построения семантического профиля пользователя рекомендательной системы третьего поколения. При этом выбранные причинные правила классификации, по сути, представляют интересы пользователя, которые представлены в форме, хорошо применимой на практике: механизм принятия решений (а именно, бинарное дерево решений), по сути, встроен в представление знаний о профиле пользователя.

3. Разработан эффективный механизм ассоциативной классификации.

Работа предложенного алгоритма ассоциативной классификации демонстрируется на примере выработки рекомендаций для некоторого пользователя рекомендательной системы на основе его семантического профиля интересов с помощью метода фильтрации контента.

В каждом узле бинарного дерева принятия решений, которое «встроено» в семантической профиль пользователя, классификация рекомендованных пользователю продуктов выполняется с помощью процедуры взвешенного голосования. При этом в качестве веса выступает значение выбранной метрики причинной связи.

Кроме решения перечисленных выше задач, получены также следующие дополнительные результаты, которые могут быть использованы в области рекомендательных систем третьего поколения:

1. Разработан алгоритм гибридной коллаборативной фильтрации, который использует семантическое сходство интересов пользователей, в отличие от стандартных подходов, использующих только статистическую информацию.

2. Для повышения эффективности процедуры выработки рекомендаций методом коллаборативной фильтрации предложен алгоритм предварительной кластеризации пользователей на основе сходства их интересов, который использует информацию из семантического профиля пользователя. Кластеризация пользователей является важным шагом в условиях больших данных, так как она помогает снизить вычислительную сложность задачи выработки рекомендаций.

3. Предложена метрика семантического сходства пользователей, которая позволяет оценить степень сходства интересов пользователи на основе их семантических профилей. Такая метрика может, например, быть использована в процессе выработки рекомендаций методом коллаборативной фильтрации.

4. Предложен алгоритм выработки кросс-доменных рекомендаций на основе семантического профиля пользователя и гибридной коллаборативной фильтрации. Алгоритм позволяет с необходимой точностью предсказывать рейтинги, которые некоторый пользователь присвоит продуктам из тех доменов, в которых он ранее не выставлял рейтингов.

Оценка качества и вычислительной эффективности всех разработанных алгоритмов экспериментально оценена с помощью набора данных Amazon.

Разработанные алгоритмы продемонстрировали удовлетворительную и хорошую точность на выбранном тестовом наборе, если принимать во внимание малый размер выборки (от 15 экземпляров для пользователя). Точность алгоритмов может быть улучшена за счёт увеличения обучающей выборки и привлечения дополнительной информации из внешних источников данных, например, DBpedia.

ЗАКЛЮЧЕНИЕ

В диссертационной работе решена актуальная научная задача - разработка алгоритмов обработки больших данных для построения модели ассоциативно-причинной классификации и её реализация в форме программного прототипа, а также выполнено экспериментальное исследование этих алгоритмов для конкретного приложения – рекомендательной системы третьего поколения, - в том числе, получены следующие результаты:

1. Теоретически и экспериментально обоснован выбор семантически корректной и вычислительно эффективной меры оценки «силы» причинной связи атрибутов данных. На основании анализа экспериментальных результатов сделан вывод о том, что наиболее перспективными с точки зрения способности выявлять правила, представляющие причинные связи в данных, являются коэффициент регрессии и мера Клозгена. Однако, дальнейшие исследования показали, что алгоритмы классификации, использующие коэффициент регрессии, демонстрируют большую точность при обучении на выбранном тестовом наборе данных.

2. Разработан масштабируемый алгоритм автоматического построения семантической модели данных в задаче принятия решений. В его основу положена методика семантического анализа понятий. Новизна алгоритма и методики состоит в совместном использовании (в рамках одной структуры данных) спецификации семантики данных с помощью иерархии понятий онтологии данных, которые извлекаются с помощью средств DBpedia и формулируются в терминах понятий естественного языка, и структуры формальных понятий модели данных, рассматриваемой в АФП.

3. Предложена новая модель больших данных, названная семантической моделью данных, основу которой составляет иерархия понятий онтологии данных и двойственная ей иерархия формальных понятий. Для её представления использована единая структура, которая наряду с семантической компонентой (иерархией понятий онтологии) и синтаксической компонентой (иерархией формальных понятий модели данных) содержит также метаинформацию, которая используется для формирования причинной модели данных без дополнительного прохода по

ним. Структура также обеспечивает быстрый доступ к данным и эффективность вычислений.

4. Разработан алгоритм поиска причинных связей в больших данных, например, атрибутов данных и целевой переменной (в случае рекомендательной системы - причинных связей интересов пользователя и дискретных значений рейтинга некоторого продукта). Алгоритм включает в себя агрегирование данных, в ходе которого данные преобразуются к виду утверждений о свойствах атрибутов в форме предикатов, т.е. к бинарной шкале измерения. Такие предикаты рассматриваются как посылки правил классификации. Далее все полученные правила подвергаются пороговой фильтрации с использованием значения меры «силы» причинной связи.

5. Разработан алгоритм минимизации пространства атрибутов для решения задач ассоциативно-причинной классификации. Механизм основан на методах кластерного анализа. Он позволяет устранять избыточные правила, упрощая алгоритм принятия решений.

6. Выполнено экспериментальное исследование всех разработанных алгоритмов. Алгоритмы продемонстрировали точность, удовлетворяющую требованиям, выдвинутым в рамках одного из проектов, на выбранном тестовом наборе. Точность алгоритмов может быть улучшена за счёт увеличения обучающей выборки и привлечения дополнительной информации, например, из DBpedia, либо путем добавления правил в семантический профиль пользователя, т.е. путем повышения размерности итоговой модели принятия решений.

Кроме того, получены также следующие дополнительные результаты, которые относятся к области обучения и принятия решений для РС третьего поколения:

1. Разработан алгоритм гибридной коллаборативной фильтрации, который использует семантическое сходство интересов пользователей, в то время как стандартные подходы используют только статистическую информацию.

2. Разработан алгоритм гибридной коллаборативной фильтрации, который использует семантическое сходство интересов пользователей, в отличие от стандартных подходов, использующих только статистическую информацию.

3. Предложена метрика семантического сходства пользователей, которая позволяет оценить степень сходства интересов пользователи на основе их семантических профилей.

4. Предложен алгоритм выработки кросс-доменных рекомендаций на основе семантического профиля пользователя и гибридной коллаборативной фильтрации. Алгоритм позволяет с необходимой точностью предсказывать рейтинги, которые некоторый пользователь присвоит продуктам из тех доменов, в которых он ранее не выставлял рейтингов.

Основные результаты работы были представлены и получили положительную оценку на следующих конференциях:

– международная конференция «The Tenth International Workshop on Agents and Data Mining Interaction» (г. Париж, 2014 г.);

– Всероссийская научно-практическая конференция «Перспективные системы и задачи управления» (п. Домбай, 2015);

– международная конференция «Creativity in intelligent technologies and data science» (г. Волгоград, 2015);

– международная конференция «The 2015 IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA'2015)» (г. Париж, 2015);

– объединенная международная конференция «The 2015 IEEE/WIC/ACM International Conference on Web Intelligence (WI’15) and the 2015 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT’15)» (г. Сингапур, 2015).

Основные результаты диссертационной работы использованы в проектах «Контекстно-управляемый ассоциативный и причинный анализ данных для принятия решений» ПФИ ОНИТ РАН «Интеллектуальные информационные технологии, системный анализ и автоматизация», (2013-2015 гг.), «Алгоритм автомати-

ческого инкрементного обучения для улучшения распознавания табличных данных» с «EMC International Company» (2015 г.), а также при выполнении работ по контракту «Многоагентные алгоритмы для кросс-доменных рекомендательных систем» с Московским подразделением Samsung Electronics – Samsung Research Center (2014 г.), что подтверждено соответствующими актами внедрения.

Полученные результаты соответствуют п. 4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации», п. 5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации», паспорта специальности 05.13.01 - «Системный анализ, управление и обработка информации (технические системы)».

ЛИТЕРАТУРА

1. Wikipedia.org: the free encyclopedia // URL: https://en.wikipedia.org/wiki/Big_data (дата обращения 1.06.2016 г.).

2. NESSI White Paper, December 2012: Big Data A New World of Opportunities // URL: http://www.nessi-europe.eu/Files/Private/NESSI_WhitePaper_BigData.pdf (дата обращения 1.06.2016 г.).

3. Bickel P. Discussion on the paper «Sure independence screening for ultrahigh dimensional feature space» by Fan and Lv // Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2008. No. 70 (5). pp. 883–884.

4. Городецкий В.И. Состояние и перспективы интеллектуального анализа больших данных // Труды всероссийской конференции «Интеллектуальные технологии в управлении», Санкт-Петербург, 7- 9 октября 2014 г. С. 61–73.

5. KDnuggets - Data Mining, Analytics, Big Data, and Data Science // URL: http://www.kdnuggets.com/2013/11/ninesigma-rfp-numerical-data-retrieval-algorithm-using-natural-language.html (дата обращения 1.06.2016 г.).

6. Gorodetsky V. Big Data: Opportunities, Challenges and Solutions. Computer, Communication and Information Technologies. vol. 469. Ermolaev V., Mayr H.C., Nikitchenko M., et al (Eds.). Springer. 2014. pp.1–20.

7. Gorodetsky V., Samoylov V., Serebryakov *S.* Ontology–based Context–dependent Personalization Technology // Proc. of the WI/IAT 2010. Toronto. 2010. pp. 278–283.

8. The Apache Software Foundation // URL: http://www.apache.org/ (дата обращения 1.06.2016 г.).

9. The Apache Pig // URL: https://pig.apache.org/ (дата обращения 1.06.2016 г.).

10. The Apache Hive // URL: https://hive.apache.org/ (дата обращения 1.06.2016 г.).

11. The Apache HBase// URL: https://hbase.apache.org/ (дата обращения 1.06.2016 г.).

12. The Apache Spark // URL: http://spark.apache.org/ (дата обращения 1.06.2016 г.).

13. Big Data: Introducing BigInsights, IBM's Hadoop-based analytical platform // URL: http://www.slideshare.net/CynthiaSaracco/introducing-ibms-info?cm_mc_uid= 12300433611014363662019&cm_mc_sid_50200000=1446479842 (дата обращения 1.06.2016 г.).

14. The value of IBM InfoSphere BigInsights // URL: http://www.raidinc.com/assets/documents/ibm-biginsights.pdf (дата обращения 1.06.2016 г.).

15. IBM InfoSphere Streams Enabling complex analytics with ultra-low latencies on data in motion // URL: http://www.monash.com/uploads/IBM-InfoSphere-Streams-White-Paper.pdf (дата обращения 1.06.2016 г.).

16. InfoSphere BigInsights Enterprise Edition // URL: http://www-03.ibm.com/software/ products/ru/infobigienteedit (дата обращения 1.06.2016 г.).

17. InfoSphere Streams Technical Overview - Use Cases Big Data // URL: http://www.slideshare.net/ IBMInfoSphereUGFR/infosphere -streams-technical-overview-use-cases-big-data-jerome-chailloux (дата обращения 1.06.2016 г.).

18. IBM Big Data Success Stories // URL: http://public.dhe.ibm.com/software/data/sw-library/big-data/ibm-big-data-success.pdf (дата обращения 1.06.2016 г.).

19. Fan J., Han F., and Liu H. Challenges of Big Data Analysis // Princeton University, Johns Hopkins University, August 7, 2013. URL: http://arxiv.org/pdf/1308.1479.pdf (дата обращения 1.06.2016 г.).

20. Cand`es E., Li X., Ma Y., and Wright J. Robust principal component analysis // Journal of the ACM. 2011. No. 58 (3). pp. 1-39.

21. Loh P.-L., Wainwright M. High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity // The Annals of Statistics. 2012. No. 40 (3). pp. 1637–1664.

22. Liu H., Han F., Yuan M., Lafferty J., Wasserman L. High-dimensional semi-para- metric Gaussian copula graphical models // The Annals of Statistics. 2012. No. 40 (4). pp. 2293–2326.

23. Xue L., Zou H. Regularized rank-based estimation of high-dimensional nonparanormal graphical models // The Annals of Statistics. 2012. No. 40 (5). pp. 2541–2571.

24. Fan J., Guo S., Hao N. Variance estimation using refitted cross-validation in ultrahigh dimensional regression // Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2012. No. 74 (1). pp. 37–65.

25. Lam C., Yao Q. Factor modeling for high dimensional time series: Inference for the number of factors // The Annals of Statistics. 2012. No. 40 (2). pp. 694 - 726.

26. Han F., Liu H. Transition matrix estimation in high dimensional time series // Proceedings of the 30th International Conference on Machine Learning. USA. 2013. Vol. 28. pp. 172–180.

27. Huang J., Sun T., Ying Z., Yu Y., Zhang C.-H. (2013). Oracle inequalities for the lasso in the Cox model // The Annals of Statistics. 2013. No. 41(3). pp. 1142–1165.

28. Aliferis C.F., Statnikov A., Tsamardinos I., .Xenofon S.M., Koutsoukos D. Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification. Part I: Algorithms and Empirical Evaluation. // Journal of Machine Learning Research. 2010. No. 11. pp. 171-234.

29. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook. In Ricci F., Rokach L, Shapira B,·Kantor P (Eds.). Recommender Systems Handbook. Springer. 2011. pp. 1–35.

30. Tuzhilin A. Keynote presentation at International Conference on Data Mining (ICDM 2013) Dallas, Texas, December, 2012.

31. Segaran T. Programming Collective Intelligence. O'Reilly. 2006 (Русский перевод: Тоби Сегаран. Программируем коллективный разум. Издательство Символ +, 2008. – 368 С.)

32. Adomavicius G., Sankaranarayanan R., Sen S., Tuzhilin A. Incorporating contextual information in recommender systems using a multidimensional approach. // ACM Transactions on Information Systems (TOIS). 2005. No. 23(1). pp. 103–145.

33. Adomavicius G., Mobasher B., Ricci F., Tuzhilin A. Context-Aware Recommender Systems. // AI Magazine. 2011. pp. 67–80.

34. Adomavicius G., Tuzhilin A. Context-Aware Recommender Systems. In Ricci F., Rokach L, Shapira B,·Kantor P (Eds.). Recommender Systems Handbook. Springer. 2011. pp. 217–256.

35. Linked Data // URL: http://linkeddata.org (дата обращения 1.06.2016 г.).

36. Adamo J.-M. Data Mining for Association Rules and Sequential Patterns. Springer. 2000.

37. Han J., Kamber M. Data Mining: Concepts and Techniques, 2nd ed. J. Gray (Ed.). The Morgan Kaufmann Series in Data Management Systems. 2006.

38. Agrawal R., Sricant R. Fast Algorithm for Mining Association rules // Proc. of the 20th Intern. Conference on Very Large Databases. Santiago, Chile. 1994. pp. 68-77.

39. Aliferis C.F., Statnikov A., et al. Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part II: Analysis and Extensions // Journal of Machine Learning Research. 2010. No. 11. pp. 235 – 299.

40. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Representation and Reasoning Series (2nd printing ed.). San Francisco. California: Morgan Kaufmann. 1988.

41. Witten I.H., Frank E., Hall M.A. Data Mining: Practical machine learning tools and techniques (3rd Edition). San Francisco. California. Morgan Kaufmann. 2011.

42. Городецкий В.И., Самойлов В.В. Ассоциативный и причинный анализ и ассоциативные байесовские сети // Труды СПИИРАН. 2009. №9. С. 13-65.

43. Brin S., Motwani R., Silverstein C. Beyond market baskets: generalizing association rules to correlations // Proceedings of the ACM SIGMOD Intern. Conf. on Management of Data. 1997. pp. 255–264.

44. Han J., Pei J., Yin Y. Mining frequent patterns without candidate generation // Proceedings of the ACM SIGMOD Intern. Conf. on Management of Data. 2000. pp. 1–12.

45. Piatetsky–Shapiro G. Discover, analysis, and presentation of strong rules // Knowledge discovery from Databases. G. Piatetsky–Shapiro and W.Frawley (Eds.). AAAI Press/MIT Press. 1991. pp. 229-248.

46. Liu B., Hsu W., Ma Y. Integrating classification and association rule mining// Proceedings of the KDD'98, New York, NY, Aug. 1998. pp. 80–86.

47. Schapire R.E. The Boosting Approach to Machine Learning. An Overview Nonlinear Estimation and Classification. Springer. 2003. Lecture Notes in Statistics. Vol. 171. Denison D.D., Hansen M.H, Holmes C.C., Mallick B., Yu B. (Eds). pp. 149–172.

48. Blake C.L., Murphy P.M. UCI Repository of machine learning database / University of California, Department of Information and Computer Science. Irvine, CA. 1998. URL: http://www.cs.uci.edu/ mleam/mlrepository.html (дата обращения 20.06.2014 г.).

49. Li W., Han J., Pei J. CMAR: Accurate and efficient classification based on multiple class-association rules // Proceedings of the ICDM'01, San Jose, CA, Nov. 2001. pp. 369–376.

50. Yin X., Han J. CPAR: Classification Based on Predictive Association Rule // Proceedings of the SDM'03. 2003. pp. 369–376.

51. Quinlan J.R. C4.5: Programs for Machine Learning. Morgan Kaufmann. 1993.

52. Quinlan J.R., Cameron-Jones R.M. FOIL: A midterm report // Proceedings of the European Conference on Machine Learning. Vienna, Austria. 1993. pp. 3 – 20.

53. Ibrahim S., Chandran K.R. Compact Weighted Class Association Rule Mining using Information Gain // International Journal of Data Mining & Knowledge Management Process (IJDKP). 2011. Vol.1, No .6. pp. 1–13.

54. Cohen W. Fast effective rule induction // Proceedings of the ICML'95. Tahoe City, CA. 1995. pp. 115–123.

55. Wedyan S. Review and Comparison of Associative Classification Data Mining Approaches // International Journal of Computer, Information, Systems and Control Engineering. 2014. Vol. 8. No.1. pp. 34–45.

56. Huang Z., Zhou Z., He T., Wang X. ACAC: Associative Classification based on All-Confidence // Proceedings of IEEE International Conference on Granular Computing (GrC). 2011. pp. 289-293.

57. Dong G., Li J. Efficient Mining of Emerging Patterns: Discovering Trends and Differences// Proc. of the KDD'99. 1999. pp. 43–52.

58. Dong G., Zhang X., Wong L., Li J. CAEP: Classification by Aggregating Emerging Patterns// Proc. of the DS'99. 1999. pp. 30–42.

59. Fan H., Ramamohanarao K. Fast Discovery and the Generalization of Strong Jumping Emerging Patterns for Building Compact and Accurate Classifiers // IEEE Trans. Knowl. Data Eng. 2006. Vol. 18(6). pp. 721-737.

60. Li J., Dong G., Ramamohanarao K. Making use of the most expressive jumping emerging patterns for classification // Proc. of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Kyoto, Japan. 2000. pp. 220-232.

61. Condorcet N.C. Essaisur l'application de l’analyse a la probabilite des decisions rendues a la pluralité des voix. Paris: Imprimerie Royale. 1785.

62. Condorcet's jury theorem// Wikipedia.org: the free encyclopedia. URL: http://en.wikipedia.org/ wiki/Condorcet's_jury_theorem (дата обращения 1.06.2016 г.).

63. Michalski R.S. On the Quasi-Minimal Solution of the General Covering Problem // Proc. of the V International Symposium on Information Processing (FCIP-69), Bled, Yugoslavia, October 8-11, 1969. Vol. A3. pp. 125-128.

64. Michalski R.S. A Theory and Methodology of Inductive Learning // Machine Learning. Carbone J.G., Michalski R.S., Mitchel T.M. Tigoda. (Eds.). 1983. vol. 1. Palo Alto. pp. 83–134.

65. Gorodetsky V., Karsaev O., Samoilov V. Direct Mining of Rules from Data with Missing Values// Studies in Computational Intelligence, Volume 6. Chapter in book T.Y.Lin, S.Ohsuga, C.J. Liau, X.T.Hu, S.Tsumoto (Eds.). Foundation of Data Mining and Knowledge Discovery. Springer. 2005. pp. 233-264.

66. Милль Дж.Ст. Система логики силлогистической и индуктивной: Изложение принципов доказательства в связи с методами научного исследования. Пер. с англ. Изд. 5, испр. и доп. М.: ЛЕНАНД, 2011. ISBN 978-5-9710-0181-2.

67. Пять канонов Джона Милля // Vikent.ru: информ.-справочный портал. URL: http://vikent.ru/enc/834/ (дата обращения 1.06 2016 г.).

68. Kobyliński L., Walczak K. Efficient Mining of Jumping Emerging Patterns with Occurrence Counts for Classification // Transactions on Rough Sets XIII. LNCS. 2011. vol. 6499. pp . 73–88.

69. Sherhod R., Judson P.N., et al. Emerging Pattern Mining To Aid Toxicological Knowledge Discovery // Journal of Chemical Information Modeling. 2014. No. 54 (7). pp 1864–1879.

70. Silverstein C., Brin S., Motwani R. Scalable Techniques for Mining Causal Structures // Journal of Data Mining and Knowledge Discovery. 2000. Vol. 4 (2-3). Springer. pp 163-192.

71. Cooper G.F., Herskovits E. A Bayesian Method for the Induction of Probabilistic Networks from Data // Machine Learning. 1992. Vol. 9. pp. 309-347.

72. Spirtes P., Glymour C., Scheines R. Causation, Prediction, and Search (Second ed.). The MIT press. 2000.

73. Yu K., Wu X., Ding W., Wang H., Yao H. Causal Associative Classification // Proceedings of the 2011 IEEE 11th International Conference on Data Mining. IEEE Computer Society Washington, DC, USA. 2011. pp. 914-923.

74. Городецкий В.И., Серебряков С.В.. Методы и алгоритмы коллективного распознавания: обзор // Труды СПИИРАН. 2006. Вып. 3. Том 1. С. 139-171.

75. Yafi E., Alam M.A., Biswas R. Development of subjective measures of interestingness: From unexpectedness to shocking // Proceedings of World Academy of Science, Engineering and Tech. No. 26. 2007. pp. 368-370

76. Фёрстер Э., Рёнц Б. Методы корреляционного и регрессионного анализа. М.: Финансы и статистика, 1981. 302 с.

77. Mosteller F. Association and estimation in contingency tables // Journal of American Statistical Association. 1968. No. 63 (321). pp. 1-26

78. Lenca P., Vaillant B., Meyer P., Lallich S. Association Rule Interestingness Measures // Experimental and Theoretical Studies. Quality Measures in Data Mining. 2007. Vol. 43. pp. 51-76.

79. Yule G.U. On the methods of measuring association between two attributes // J. R. Stat. Soc. 75. 1912. pp. 579-642.

80. Tan P.N., Kumar V., Srivastava J. Selecting the right objective measure for association analysis // Joirnal of Information Systems - KDD. 2004. No. 4. pp. 293-313.

81. Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large databases // Proceedings of ACM SIGMOD International Conf. on Management of Data. In P. Buneman, & S. Jajodia (eds.). 1993. pp. 207-216.

82. Clark P., Boswell R. Rule induction with cn2: some recent improvements // Proceedings of the European Working Session on Learning EWSL-91, Porto, Portugal. 1991. pp. 151-163.

83. Clark P, Brin S., Motwani R., Ullman J., Tsur S. Dynamic itemset counting and implication rules for market basket data // Proceedings of ACM-SIGMOD International Conference on Management of Data, Montreal, Canada. 1997. pp. 255-264.

84. Tan P., Kumar V. Interestingness measures for association patterns: A perspective. Technical Report TR00-036 // Proceeedings of Workshop on Postprocessing in Machine Learning and Data, Mining University of Minnesota, Department of Computer Science. 2000.

85. Sahar S., Mansour Y. An empirical evaluation of objective interestingness criteria // Proceeedings of SPIE Conference on Data Mining and Knowledge Discovery, Orlando, FL. 1999. pp. 63-74.

86. Wikipedia.org: the free encyclopedia. Jaccard index // URL: http://en.wikipedia.org/wiki/: http://en.wikipedia.org/wiki/Jaccard_index (accessed 01.06.2016 г.)

87. Sebag M., Schoenauer M. Generation of rules with certainty and confidence factors from incomplete and incoherent learning bases // Proc. of the European Knowledge Acquisition Workshop EKAW'88. 1988. pp. 28.1-28.20.

88. Иоффе А.Я., Марков В.И., Петухов Г.Б. и др. Вероятностные методы в прикладной кибернетике: Учебное пособие. Под ред. Р.М. Юсупова. Л. 1976. 424 с.

89. Городецкий В.И., Самойлов В.В. Ассоциативный и причинный анализ и ассоциативные байесовские сети // Труды СПИИРАН. 2009. № 9. С. 13-65.

90. Li J., Le T.D, Liu L., Liu J., Jin Z., Sun B. Mining causal association rules // Proc. of The First IEEE ICDM Workshop on Causal Discovery (CD 2013). 2013. pp. 114-123.

91. UCI Machine Learning Repository // URL: http://archive.ics.uci.edu/ml/ (accessed 01.06.2016 г.)

92. Cooper G.F. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships // Journal of Data Mining and Knowledge Discovery. 1997. No. 1. pp. 203–224/

93. Wikipedia.org: свободная энциклопедия. ROC-кривая // URL: https://ru.wikipedia.org/wiki/ROC-кривая (дата обращения 01.06.2016 г.)

94. Wikipedia.org: the free encyclopedia. Precision and recall // URL: http://en.wikipedia.org/wiki/Precision_and_recall (дата обращения 01.06.2016 г.)

95. Wikipedia.org: свободная энциклопедия. Скользящий контроль // URL: http://machinelearning.ru/wiki/index.php?title=Скользящий_контроль (дата обращения 01.06.2016 г.)

96. Weka 3: Data Mining Software. URL: http://www.cs.waikato.ac.nz/ml/weka (дата обращения 01.06.2016 г.)

97. Causality Challenge №1: Causation and Prediction // URL: http://www.causality.inf.ethz.ch/ challenge.php (дата обращения 01.06.2016 г.)

98. Causality Challenge №1: Causation and Prediction. Результаты соревнования // URL: http://www.causality.inf.ethz.ch/challenge.php?page=results&ds=cina0 (дата обращения 01.06.2016 г.)

99. Ganter, B., and Wille, R. Formal Concept Analysis: Mathematical Foundations. Springer-Verlag, Berlin. 1998.

100. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2001. – 384с.

101. Gavrilova T., Gladkova M. Big Data Structuring: The Role of Visual Models and Ontologies // Procedia Computer Science, Elseiver (2nd International Conference on Information Technology and Quantitative Management, ITQM 2014). – 2014. – Т. 31. – с. 336 - 343.

102. Costa A.C., Guizzardi R.S.S., Filho J.G.P. COReS: Context-aware, ontology based recommender system for service recommendation // Proc. of 19-th International Conference on Advanced Information Systems Engineering (CAISE07). 2007.

103. Middleton S.E., Shadbolt N.R., De Roure, D.C. Ontological user profiling in recommender systems // ACM Transaction on Information Systems. 2004. vol. 22(1). pp. 54 – 88.

104. Trajkova J., Gauch S. Improving Ontology-Based User Profile //Proc. of RIAO. 2004. pp. 380-390.

105. Kolchin M., Klimov N., Andreev A., Shilin I., Garayzuev D., Mouromtsev D., Zakoldaev D. Ontologies for Web of Things: a Pragmatic Review // Communications in Computer and Information Science - 2015, Vol. 518, pp. 102-116.

106. Замула Д.А., Муромцев Д.И. Автоматизированное проектирование мобильных приложений на основе онтологий и семантических данных // Известия высших учебных заведений. Приборостроение - 2015. - Т. 58. - № 11. - С. 939-944.

107. Муромцев Д.И., Горовой А.А., Злобин А.Н., Катков Ю.В., Починок И.Н. Архитектура системы управления знаниями на основе Wiki-технологии с интегрированными онтологическими моделями // Известия высших учебных заведений. Приборостроение - 2011. - Т. 54. - № 1. - С. 5-12.

108. WordNet – a lexical database for English // URL: https://wordnet.princeton.edu.

109. DBpedia - Towards a Public Data Infrastructure for a Large, Multilingual, Semantic Knowledge Graph // URL: http://wiki.dbpedia.org.

110. Городецкий В.И., Тушканова О.Н. Онтологии и персонификация профиля пользователя в рекомендующих системах третьего поколения // Онтология проектирования. – 2014. – № 3 (13). – С.7-31.

111. Pena P., del Hoyo R., Vea-Murguia J., Gonzalez C. Mayo S. Collective Knowledge Ontology User Profiling for Twitter // Proc. of 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT). 2013. vol. 1. pp. 439-444.

112. OpenDNS: Cloud-Delivered Security Enforcement //URL: https://www.opendns.com/.

113. Garcia-Silva A., Garcia-Castro L. J., Garcia Castro A., Corcho O. Social Tags and Linked Data for Ontology Development: A Case Study in the Financial Domain // Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), ACM New York, NY. USA. 2014. pp. 1-10.

114. Wikipedia.org: the free encyclopedia. Folksonomy // URL: https://en.wikipedia.org/wiki/ Folksonomy (дата обращения 1.06.2016 г.).

115. Delicious Dataset // URL: http://grouplens.org/datasets/hetrec-2011.

116. OpenCyc // URL: http://www.opencyc.org.

117. UMBEL- upper mapping and binding exchange level // URL: http://www.umbel.org.

118. Obitko M., Snasel V., Smid J. Ontology Design with Formal Concept Analysis. Technical University of Ostrava, Dept. of Computer Science. 2004. pp. 111–119.

119. Gu T. Using Formal Concept Analysis for Ontology Structuring and Building. ICIS, Nanyang Technological University. 2003.

120. Cimiano P., Hotho A., Stumme G., Tane J. Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies. Lecture Notes in Computer Science. vol. 2961. 2004. pp 189-207.

121. Godin R., Missaoui R., Alaoui H. Incremental Concept Formation Algorithms Based on Galois Lattices // Computation Intelligence. 1995. vol. 11(2). pp. 246-267.

122. Кузнецов С.О. Теория решеток для интеллектуального анализа данных // URL: https://www.hse.ru/data/143/315/1234/ltida3_1409.pdf (дата обращения 1.06.2016 г.).

123. Биркгоф Г. Теория структур. М.: ИЛ, 1952.

124. Яглом И.М. Булева структура и ее модели. Москва, Сов. радио, 1980, 192 С.

125. Биркгофф Г. Теория решеток. М.: Наука, Главная редакция физико-математической литературы, 1984. — 568 с.

126. Amazon Data Set // URL: https://snap.stanford.edu/data/amazon-meta.html (дата обращения 1.06.2016 г.).

127. Wikipedia.org: the free encyclopedia. Cluster analysis. https://en.wikipedia.org/wiki/Cluster_analysis (дата обращения 1.06.2016 г.).

128. Результаты экспериментов по построению онтологий Amazon // URL: https://drive.google.com/open?id=0ByiklOTaH_zZbi12YTFwVFY5NG8 (дата обращения 1.06.2016 г.).

129. Www.MachineLearning.ru – профессиональный вики-ресурс, посвященный машинному обучению и интеллектуальному анализу данных // URL: www.MachineLearning.ru (дата обращения 1.06.2016 г.).

130. Drechsler R., Becker B. Binary Decision Diagrams: Theory and Implementation. Springer. 1998.

131. Городецкий В.И., Самойлов В.В. Контекстно–зависимое обучение для принятия решений // Труды 2-й Международной конференции «Автоматизация управления и интеллектуальные системы и среды» (АУИСС-2011), Красная поляна, 15–22 декабря 2011 г.

132. Самойлов В.В. Агрегирование многомерных объектных данных в задачах анализа ассоциаций // Искусственный интеллект и принятие решений. 2009. №3. 15 – 23.

133. Gorodetsky V., Samoylov V. Feature Extraction for Machine Learning: Logic-Probabilistic Approach // International Journal for Machine Learning Research. Workshop and Conference Proceedings: The Fourth Workshop on Feature Selection in Data Mining. 2010. Vol. 10. pp. 55-65.

134. Мандель И.Д. Кластерный анализ. М.: Финансы и статистика, 1988. – 176 с.

135. Терентьев П.В. Метод корреляционных плеяд // Вестник ЛГУ. – 1959. - №9. С. 137 – 141.

136. Lops P., De Gemmis M., Semeraro G. Content-based recommender systems: state of the art and trends. In: Ricci F., Rokach L., Shapira B. (eds.) Recommender systems handbook, pp. 73-105. Springer, Hedelberg. 2011. pp.73-106.

137. Desrosiers C., Karypis G. A comprehensive survey of neighborhood-based recommendation methods. In: Ricci F., Rokach L., Shapira B. (eds.) Recommender systems handbook. Springer, Hedelberg. 2011. pp. 107-144.

138. Su Z., Yan J., Ling H., Che H. Research on personalized pecommendation algorithm based on ontological user interest model// Journal of Computational Information Systems. 2012. Vol. 8(1). pp. 169–181.

139. Fernández-Tobíias I., Cantador I., Kaminskas M., Ricci, F. Cross-domain Recommender Systems: A Survey of the State of the Art // Proc. of the 2nd Spanish Conference on Information Retrieval. 2012. pp. 187-198.

140. Shani G., Gunawardana A. Evaluating Recommendation Systems. In Ricci F., Rokach L, Shapira B,·Kantor P (Eds.). Recommender Systems Handbook. Springer. 2011. pp. 1–35.

ПРИЛОЖЕНИЕ А РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ ПО ИССЛЕДОВАНИЮ ЧИСЛЕННЫХ МЕР ОЦЕНКИ АССОЦИАТИВНЫХ И ПРИЧИННЫХ СВЯЗЕЙ ДЛЯ АНАЛИЗА БОЛЬШИХ ДАННЫХ

А.1 Описание модифицированного набора данных Adult

Модифицированный набор данных *Adult* (*MAdult*) сформирован из открытого набора данных *Adult*, который представляет собой фрагмент базы данных переписи населения, проведенной бюро переписи населения США в 1994 году (таблица А.1).

На основании данных о некотором индивиде необходимо предсказать, получает ли он более $50,000 в год. Модифицированный набор данных *Adult* содержит 30160 примеров без пропущенных значений, каждый из который описан 99 бинарными признаками.

Таблица А.1 - Характеристики модифицированного набора данных Adult

Характеристики атрибутов:	Бинарные	Количество экземпляров:	30160
Пропущенные значения:	Нет	Количество атрибутов:	99

А.2 Результаты эксперимента с использованием способа 1

Проведенный эксперимент включает нахождение причинных связей в данных с помощью мер *уверенность* (*conf*), *мера Лапласа* (L), *J-мера* (J), *убеждение* (V), *добавочное значение* (AV), *индекс Джини* (G), *мера Клозгена* (K), *мера Сибега и Шонауера* (SEB), *коэффициент регрессии* (R) и *фактор определенности* (F) двумя способами и сравнение с результатами работы [90].

Найденные причинные связи представлены в виде однолитерных правил, в заключении которых находится метка класса:

Если <атрибут>=true, то класс = <метка класса>,

причем метка класса принимает значение 1, если индивид зарабатывает более $50,000, и значение 0 –в противном случае.

Полученный список правил был сравнен с правилам, полученными авторами работы [90] с помощью методик CAR [90], CCC [92] и CCU [70].

Приведем шаги формирования правил с помощью вышеперечисленных метрик для набора данных MAdult.

1. Для каждого бинарного атрибута набора данных были сформированы два правила следующего вида: (1) *если <атрибут>=true, то класс = 0;* (2) *если <атрибут>=true, то класс = 1.*

2. Для каждого правила в полученном списке была вычислена соответствующая численная мера *F*. Затем, из двух правил (1 и 2) правило, имеющее меньшее значение численной меры *F*, отбрасывалось.

3. В итоговом списке были оставлены правила, для которых мера *F* принимает значение, удовлетворяющее некоторому порогу θ[9].

В первом варианте эксперимента использовались только первые два шага алгоритма, во втором – все три. Таким образом, в списках, полученных в ходе первого варианта эксперимента, правил оказывалось больше.

В таблицах А.2 и А.3 приведены результаты сравнения списков правил, полученных в работе [90], и правил, сформированных с помощью соответствующих метрик в ходе первого и второго варианта эксперимента.

В столбце «CAR» таблицы А.2 плюсами отмечены правила, полученные авторами [90] с помощью предложенной ими методики; в столбцах «CCC» и «CCU» плюсами отмечены правила полученные авторами [90] с помощью методик построения локальных причинных структур CCC [92] и CCU [70] соответственно.

В последующих столбцах плюсами отмечены те правила, из представленных в первом столбце, которые попали в список 30 «лучших» для соответствующей меры.

В таблице А.3 в столбцах «R» - «F» «плюсами» отмечены те правила, которые попали в список правил для соответствующих мер в ходе второго варианта эксперимента.

---

[9] Выбор значения порога θ зависит от конкретной задачи и набора данных и в работе не обсуждается

А.3 Результаты эксперимента с использованием метода 2

На основе наборов правил, полученных с помощью мер *уверенность* (*conf*), *мера Лапласа* (L), *J-мера* (J), *убеждение* (V), *добавочное значение* (AV), *индекс Джини* (G), *мера Клосгена* (K), *мера Сибега и Шонауера* (SEB), *коэффициент регрессии* (R) и *фактор определенности* (F), и набора правил из работы [90] (методика CAR) были построены простые классификаторы для набора данных MAdult следующим образом:

4. каждое правило из сформированного для каждой меры списка представляет собой отдельный классификатор, который может голосовать только за тот класс, который представлен в заключении соответствующего правила;

5. объединение решений этих отдельных классификаторов происходит по схеме простого голосования.

Для оценки качества исследуемых классификаторов были подсчитаны следующие характеристики:

6. матрица неточностей (*confusion matrix*);

7. среднеквадратическое отклонение;

8. чувствительность классификатора (*true positive rate*) для каждого класса;

9. коэффициент ложной тревоги классификатора (*false positive rate*) для каждого класса;

10. точность классификатора (*precision*) для каждого класса;

11. полнота классификатора (*recall*) для каждого класса;

12. F-мера классификатора (*F-measure*) для каждого класса;

13. площадь под ROC-кривой для классификатора.

Тестирование классификаторов выполнялось с помощью процедуры скользящего контроля с 10 блоками (*10-fold cross-validation*).

Таблица А.2 – Правила, сформированные с помощью мер, по сравнению с правилами из работы [90], вариант1

Правило	CAR	CCC	CCU	R	G	J	conf	L	V	AV	K	SEB	F
1	2	3	4	5	6	7	8	9	10	11	12	13	14
Education=1-4th → ≤ 50K	-	+	+	+	-	+	+	+	+	+	+	+	+
Education=10th → ≤ 50K	+	+	+	+	+	+	+	+	+	+	+	+	
Education=11th → ≤ 50K	+	+	+	+	+	+	+	+	+	+	+	+	
Education=12th → ≤ 50K	+	+	+	+	+	+	+	+	+	+	+	+	
Education=5-6th → ≤ 50K	+	+	+	+	+	+	+	+	+	+	+	+	
Education=7-8th → ≤ 50K	+	+	+	+	-	+	+	+	+	+	+	+	
Education=9th → ≤ 50K	+	+	+	+	+	+	+	+	+	+	+	+	
Education=bachelors → > 50K	+	+	+	+	-	+	-	-	+	+	+	-	+
Education=doctorate → > 50K	+	+	+	+	+	+	+	+	+	+	+	+	+
Education=HS-grad → ≤ 50K	+	+	+	+	+	+	+	+	+	+	+	+	+
Education=masters → > 50K	+	+	+	+	-	+	+	+	+	+	+	+	+
Education=preschool → ≤ 50K	-	+	-	+	-	-	+	+	+	+	+	+	+
Education=prof-School → > 50K	+	+	+	+	-	+	+	+	+	+	+	+	+
Education=some-college → ≤ 50K	+	+	+	+	-	+	+	+	+	+	+	+	+
Occupation=adm-clerical → ≤ 50K	+	-	-	+	+	+	+	+	+	+	+	+	+
Occupation=craft-repair → ≤ 50K	+	-	-	+	-	-	+	+	+	+	+	+	+
Occupation=exec-managerial → > 50K	+	-	-	+	-	+	-	-	+	+	+	-	+
Occupation=farming-fishing → ≤ 50K	+	-	-	+	+	+	+	+	+	+	+	+	+
Occupation=handlers-cleaners → ≤ 50K	+	-	-	+	+	+	+	+	+	+	+	+	+
Occupation=machine-op-inspct → ≤ 50K	+	-	-	+	+	+	+	+	+	+	+	+	+

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Occupation=other-service → ≤ 50K	+	-	-	+	+	+	+	+	+	+	+	+	+
Occupation=prof-specialty → > 50K	+	-	-	+	+	+	-	-	+	+	+	-	+
Occupation=protective serv → > 50K	+	-	-	+	-	-	-	-	+	+	+	-	+
Occupation=sales → > 50K	+	-	-	+	-	-	-	-	+	+	+	-	+
Occupation=tech-support → > 50K	+	+	+	+	-	-	-	-	+	+	+	-	+
Occupation=transport-moving → ≤ 50K	+	-	-	+	-	-	+	+	+	+	+	+	+
Workclass=federal-gov → > 50K	+	+	+	+	-	+	-	-	+	+	+	-	+
Workclass=local-gov → > 50K	+	+	+	+	-	-	-	-	+	+	+	-	+
Workclass=private → ≤ 50K	+	+	+	+	+	+	+	+	+	+	+	+	+
Workclass=sel-emp-inc → > 50K	+	+	-	+	-	+	+	+	+	+	+	+	+
Workclass=sel-emp-not-inc → > 50K	+	+	+	+	-	-	-	-	+	+	+	-	+
Workclass=state-gov → > 50K	+	-	-	+	-	-	-	-	+	+	+	-	+
Общие с CAR	**-**	**19**	**18**	**30**	**14**	**22**	**20**	**20**	**30**	**30**	**30**	**20**	**30**
Общие с CCC	**18**	**-**	**18**	**20**	**8**	**16**	**15**	**15**	**20**	**20**	**20**	**15**	**20**
Общие с CCU	**18**	**18**	**-**	**18**	**8**	**15**	**13**	**13**	**18**	**18**	**18**	**13**	**18**

Таблица А.3 – Правила, сформированные с помощью мер, по сравнению с правилами из работы [90], вариант 2

Правило	CAR	CCC	CCU	R	G	J	conf	L	V	AV	K	SEB	F
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Education=1-4th → ≤ 50K	-	+	+	+	-	-	+	+	+	+	-	+	+
Education=10th → ≤ 50K	+	+	+	-	+	-	+	+	+	-	+	+	+
Education=11th → ≤ 50K	+	+	+	+	+	-	+	+	+	+	+	+	+

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Education=12th → ≤ 50K	+	+	+	-	-	-	+	+	+	-	-	+	+
Education=5-6th → ≤ 50K	+	+	+	+	-	-	+	+	+	+	-	+	+
Education=7-8th → ≤ 50K	+	+	+	-	-	-	+	+	+	+	+	+	+
Education=9th → ≤ 50K	+	+	+	+	-	+	+	+	+	+	+	+	+
Education=bachelors → > 50K	+	+	+	+	-	-	-	-	-	-	+	-	-
Education=doctorate → > 50K	+	+	+	+	+	-	-	-	+	+	+	-	+
Education=HS-grad → ≤ 50K	+	+	+	-	+	-	-	-	-	-	+	-	
Education=masters → > 50K	+	+	+	+	-	+	-	-	-	+	+	-	
Education=preschool → ≤ 50K	-	+	-	+	-	-	+	+	+	+	-	+	+
Education=prof-School → > 50K	+	+	+	+	-	+	-	-	+	+	+	-	+
Education=some-college → ≤ 50K	+	+	+	-	+	-	-	-	-	-	-	-	
Occupation=adm-clerical → ≤ 50K	+	-	-	-	-	-	-	-	-	-	+	-	
Occupation=craft-repair → ≤ 50K	+	-	-	-	-	-	-	-	-	-	-	-	
Occupation=exec-managerial → > 50K	+	-	-	+	-	-	-	-	-	+	+	-	
Occupation=farming-fishing → ≤ 50K	+	-	-	-	-	+	-	-	-	-	+	-	
Occupation=handlers-cleaners → ≤ 50K	+	-	-	+	+	-	+	+	+	+	+	+	+
Occupation=machine-op-inspct → ≤ 50K	+	-	-	-	+	+	-	-	-	-	+	-	
Occupation=other-service → ≤ 50K	+	-	-	+	+	+	+	+	+	+	+	+	+
Occupation=prof-specialty → > 50K	+	-	-	+	+	-	-	-	-	+	+	-	
Occupation=protective serv → > 50K	+	-	-	-	-	-	-	-	-	-	-	-	
Occupation=sales → > 50K	+	-	-	-	-	-	-	-	-	-	-	-	
Occupation=tech-support → > 50K	+	+	+	-	-	-	-	-	-	-	-	-	

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Occupation=transport-moving → ≤ 50K	+	-	-	-	-	-	-	-	-	-	-	-	
Workclass=federal-gov → > 50K	+	+	+	-	-	-	-	-	-	-	-	-	
Workclass=local-gov → > 50K	+	+	+	-	-	-	-	-	-	-	-	-	
Workclass=private → ≤ 50K	+	+	+	-	+	+	-	-	-	-	+	-	
Workclass=sel-emp-inc → > 50K	+	+	-	+	-	+	-	-	-	+	+	-	
Workclass=sel-emp-not-inc → > 50K	+	+	+	-	-	-	-	-	-	-	-	-	
Workclass=state-gov → > 50K	+	-	-	-	-	-	-	-	-	-	-	-	
Общие с CAR	**-**	**18**	**17**	**12**	**10**	**8**	**8**	**8**	**10**	**12**	**18**	**8**	**10**
Общие с CCC	**18**	**-**	**18**	**10**	**6**	**5**	**8**	**8**	**10**	**10**	**11**	**8**	**10**
Общие с CCU	**18**	**18**	**-**	**8**	**6**	**4**	**7**	**7**	**9**	**8**	**10**	**7**	**9**

Дополнительно для сравнения использовался алгоритм классификации BayesNet [41], реализованный в системе Weka.

В ходе построения классификаторов количество используемых правил варьировалось таким образом, чтобы получить лучшие характеристики для каждого классификатора, но при этом количество неклассифицированных экземпляров не превысило 10% от общего количества экземпляров.

В таблице А.4 представлены характеристики классификаторов, полученных для наборов правил метрик и набора правил из работы [90]. А также, характеристики классификатора построенного, в Weka с помощью алгоритма BayesNet [41] для набора данных MAdult.

Таблица А.4 - Результаты оценки классификаторов

J-мера (J); количество правил: 69; **всего не классифицировано: 2772; RMSE = 0.7845; AUC =0,527**

0	1	предсказ.
4860	15574 |	0
1281	5673 |	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.238	0.184	0.791	0.238	0.366	0
	0.816	0.762	0.267	0.816	0.402	1
Взвеш. ср..	0.385	0.331	0.658	0.385	0.375	

Индекс Джини (G); количество правил: 61; всего не классифицировано: 1056; RMSE = 0.7215; AUC = 0,502

0	1	предсказ.
10016	11878 |	0
3273	3937 |	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.457	0.454	0.754	0.457	0.569	0
	0.546	0.543	0.249	0.546	0.342	1
Взвеш. ср.	0.479	0.476	0.629	0.479	0.513	

Уверенность (conf); количество правил: 96; всего не классифицировано: 0; RMSE = 0.4989; AUC = 0.5

0	1	предсказ.
22652	0 |	0
7508	0 |	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	1	1	0.751	1	0.858	0
	0	0	0	0	0	1
Взвеш. ср.	0.751	0.751	0.564	0.751	0.644	

Мера Лапласа (L); количество правил: 98; всего не классифицировано: 0; RMSE = 0.4989; AUC = 0.5

0	1	предсказ.
22652	0 |	0
7508	0 |	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	1	1	0.751	1	0.858	0
	0	0	0	0	0	1
Взвеш. ср.	0.751	0.751	0.564	0.751	0.644	

Мера Сибега и Шонауера (SEB); количество правил: 92; всего не классифицировано: 0; RMSE = 0.4989; AUC = 0.5

0	1	предсказ.
22652	0 |	0
7508	0 |	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	1	1	0.751	1	0.858	0
	0	0	0	0	0	1
Взвеш. ср.	0.751	0.751	0.564	0.751	0.644	

Продолжение Таблицы А.4

Убеждение (V); количество правил: 61; всего не классифицировано: 2984; RMSE = 0.4785; AUC = 0.799

0	1	предсказ.
15214	5205	0
1018	5739	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.745	0.151	0.937	0.745	0.83	0
	0.849	0.255	0.524	0.849	0.648	1
Взвеш. ср.	0.771	0.177	0.835	0.771	0.785	

Добавочное значение (AV); количество правил: 74; всего не классифицировано: 2988; RMSE = 0.4819; AUC =0.799

0	1	предсказ.
15014	5374	0
937	5847	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.736	0.138	0.941	0.736	0.826	0
	0.862	0.264	0.521	0.862	0.649	1
Взвеш. ср.	0.768	0.169	0.836	0.768	0.782	

Мера Клосгена (K); количество правил: 30; всего не классифицировано: 3098; RMSE = 0.4738; AUC = 0.792

0	1	предсказ.
15487	4892	0
1184	5499	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.76	0.177	0.929	0.76	0.836	0
	0.823	0.24	0.529	0.823	0.644	1
Взвеш. ср.	0.775	0.193	0.83	0.775	0.789	

Фактор оперделенности (F); количество правил: 64; всего не классифицировано: 2951; RMSE = 0.479; AUC = 0.797

0	1	предсказ.
15209	5232	0
1012	5756	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.744	0.15	0.938	0.744	0.83	0
	0.85	0.256	0.524	0.85	0.648	1
Взвеш. ср.	0.771	0.176	0.835	0.771	0.785	

Коэффициент регрессии (R); количество правил: 50; всего не классифицировано: 2928; RMSE = 0.4791; AUC = 0.798

0	1	предсказ.
15209	5252	0
1000	5771	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.743	0.148	0.938	0.743	0.83	0
	0.852	0.257	0.524	0.852	0.649	1
Взвеш. ср.	0.77	0.175	0.835	0.77	0.784	

CAR - правила из работы [24]; количество правил: 30; всего не классифицировано: 1100; RMSE = 0.5067; AUC = 0.6

0	1	предсказ.
17421	4444	0
3017	4178	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.797	0.419	0.852	0.797	0.824	0
	0.581	0.203	0.485	0.581	0.528	1
Взвеш. ср.	0.743	0.366	0.761	0.743	0.751	

Weka (BayesNet); всего не классифицировано: 0; RMSE = 0.3659; AUC = 0.745

0	1	предсказ.
19931	2721	0
2957	4551	1

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.88	0.394	0.871	0.88	0.875	0
	0.606	0.12	0.626	0.606	0.616	1
Взвеш. ср.	0.812	0.326	0.81	0.812	0.811	

ПРИЛОЖЕНИЕ Б РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ ПО ПОСТРОЕНИЮ СЕМАНТИЧЕСКОЙ МОДЕЛИ ДАННЫХ

Таблица Б.1 – Продукты пользователя «ANIHXYXKPZL2H»

Продукт	Категория
The Big Clock	Video
It's a Mad, Mad, Mad, Mad World	DVD
Red Rock West	DVD
My Favorite Brunette	DVD
Deathtrap	DVD
Strangers on a Train	DVD
The Woman in the Window	Video
Funny Girl	Video
The Caine Mutiny	Video
Sunset Boulevard (Special Collector's Edition)	DVD
My Favorite Brunette	DVD
North by Northwest	DVD
Casablanca	Video
Road to Utopia	DVD
Funny Girl	DVD
A Big Hand for the Little Lady	Video
It's a Mad, Mad, Mad, Mad World	Video
Casablanca	Video
It's a Mad, Mad, Mad, Mad World	Video
Jolson Sings Again	Video
Strangers on a Train (British Version)	Video
The Caine Mutiny	DVD
The 39 Steps - Criterion Collection	DVD
Dial M for Murder	Video
North By Northwest - Limited Edition Collector's Set	DVD
Bad Day at Black Rock	Video
Lady Vanishes	Video
Deathtrap	Video
The Jolson Story	Video
North by Northwest - Special Edition	Video

Casablanca (includes CD of Soundtrack)	Video
The Lady Vanishes - Criterion Collection	DVD
My Cousin Vinny	DVD
My Cousin Vinny	Video
The Music Man (Widescreen Edition)	Video
Rear Window	Video
The Lady Vanishes	DVD
The Music Man (Special Edition)	DVD
12 Angry Men	DVD
Backstage at the Dean Martin Show	Books
My Favorite Brunette	Video
My Cousin Vinny	Video
Casablanca	Video
The Lady Vanishes	DVD
The 39 Steps	DVD
39 Steps (1935)	Video
Lady Vanishes (1938)	DVD
Funny Girl	Video
The Music Man	Video
Alfred Hitchcock and the Making of Psycho	Books
The Treasure of the Sierra Madre	Video
Casablanca	DVD
The List of Adrian Messenger	Video
Rear Window (Collector's Edition)	DVD
My Favorite Brunette	Video

Таблица Б.2 – Продукты пользователя «A3DMZKTBIJURE1»

Название	**Категория**
Eyes of the Dragon: A Story	Book
On Writing	Book
Carrie	Book
The Gunslinger (The Dark Tower, Book 1)	Book
Ring	Book
The Drawing of the Three (The Dark Tower, Book 2)	Book
Hearts In Atlantis	Book

To Kill a Mockingbird	Book
Carrie (Los Jet De Plaza & Janes. Biblioteca De Stephen King. 102, 8)	Book
The Exorcist	Book
Harry Potter and the Order of the Phoenix (Book 5 Audio CD)	Book
The Gunslinger (The Dark Tower, Book 1)	Book
On Writing	Book
Harry Potter and the Order of the Phoenix (Book 5, Audio)	Book
Carrie	Book
Hearts In Atlantis (MTI)	Book
The Gunslinger (The Dark Tower, Book 1)	Book
Carrie	Book
Imajica : Featuring New Illustrations and an Appendix	Book
Imajica	Book
Interview with the Vampire : Anniversary edition (The vampire chronicles)	Book
To Kill a Mockingbird	Book
Max Notes - To Kill a Mockingbird (MAXnotes)	Book
To Kill a Mockingbird	Book
The Regulators	Book
Harry Potter and the Order of the Phoenix (Book 5)	Book
Nightmares and Dreamscapes : Unabridged edition (Vol 3)	Book
Hearts In Atlantis : New Fiction	Book
The Gunslinger (The Dark Tower, Book 1)	Book
On Writing : A Memoir Of The Craft	Book
Watchers	Book
To Kill a Mockingbird	Book
The Drawing of the Three (The Dark Tower, Book 2)	Book
The Drawing of the Three (The Dark Tower, Book 2)	Book
Interview with the Vampire	Book
Hearts In Atlantis MTI CD	Book
Hearts in Atlantis	Book
Carrie	Book
On Writing	Book
Harry Potter and the Order of the Phoenix (Book 5, Deluxe Edition)	Book

False Memory	Book
To Kill a Mockingbird : The 40th Anniversary Edition of the Pulitzer Prize-Winning Novel	Book
To Kill a Mockingbird (Cliffs Notes)	Book
Wolves of the Calla (The Dark Tower, Book 5)	Book
The Drawing of the Three (The Dark Tower, Book 2)	Book
On Writing	Book
The Drawing of the Three (The Dark Tower, Book 2)	Book
The Drawing of the Three (The Dark Tower, Book 2)	Book
False Memory	Book
To Kill a Mockingbird	Book
The Gunslinger (The Dark Tower, Book 1)	Book
On Writing : A Memoir Of The Craft	Book
The Gunslinger (The Dark Tower, Book 1)	Book
Darkness, Tell Us	Book
Interview with the Vampire	Book
Watchers	Book
Interview with the Vampire	Book
Nightmares & Dreamscapes	Book
Interview With the Vampire	Book
Kill a Mockingbird, To	Book
The Eyes of the Dragon	Book

Таблица Б.3 – Продукты пользователя «A3IMNZSYDOTTU6»

Название	**Категория**
Danzig 4 (Reis)	Music
Danzig	Music
God Lives Underwater	Music
Colour & Shape	Music
Covenant	Music
The Thing	Video
Short Bus	Music
Star Trek III - The Search for Spock (Special Edition)	DVD
Degradation Trip	Music

Superunknown [Single]	Music
Empty (Reis)	Music
Naveed	Music
Superunknown	Music
Visual Audio Sensory Theater	Music
Star Trek - The Motion Picture (The Director's Edition)	DVD
Music for People	Music
Star Trek II - The Wrath of Khan	Video
Creepshow	DVD
Star Trek VI - The Undiscovered Country	Video
Ministry - Tapes of Wrath	DVD
In Case You Didn't Feel Like Showing Up	Music
True Carnage	Music
Villains	Music
God Says No	Music
Star Trek V - The Final Frontier	DVD
Frailty	DVD
Tab	Music
Star Trek IV - The Voyage Home	DVD
Danzig 5: Blackacidevil	Music
The Mind Is a Terrible Thing to Taste	Music
The Fragile	Music
The Fog (Special Edition)	DVD
A Christmas Story	Video
The Thing - Collector's Edition	DVD
There Is Nothing Left to Lose	Music
A Christmas Story (Full Screen Edition)	DVD
The Amalgamut	Music
Parachutes	Music
Louder Than Love	Music
Powertrip	Music
A Christmas Story	Video
Alice, Sweet Alice	DVD
Star Trek Generations	Video

Spine of God	Music
No Name Face	Music
Ministry Keianh	Music
Star Trek Generations	Video
No Name Face	Music
Broken	Music
Monstruos, Inc. (Monsters, Inc.)	Video
A-Sides	Music
Creepshow	Video
Star Trek II - The Wrath of Khan	DVD
Live on the Black Hand Side	Music
Sacrifice	Music
Sphinctour	Music
Greatest Fits	Music
Star Trek Generations	DVD
Downward Spiral	Music
Gravity	Music
The Thing (Widescreen Edition)	Video
Pretty Hate Machine	Music
Maximum Violence	Music
Creepshow	Video
Star Trek II - The Wrath of Khan	DVD
The Fog	Video
Star Trek - The Motion Picture	Video
Silver Side Up	Music
Creepshow 2	DVD
Star Trek III - The Search for Spock	DVD
Star Trek IV - The Voyage Home (Special Edition)	DVD
Star Trek III - The Search for Spock	DVD
Star Trek III - The Search for Spock	Video
Monsters, Inc. (Collector's Edition)	DVD
Monsters, Inc.	Video
Creepshow 2	Video
Star Trek IV - The Voyage Home	DVD

Star Trek - The Motion Picture (The Director's Edition) (Widescreen)	Video
Superjudge	Music
Foo Fighters	Music
Clumsy [Single]	Music
Star Trek II - The Wrath of Khan (Director's Edition)	DVD
Star Trek IV - The Voyage Home	Video
Monstruos, Inc. (Monsters, Inc. - Spanish Edition)	DVD
Basket Case (20th Anniversary Special Edition)	DVD
Degradation Trip Volumes 1 & 2	Music
Danzig 3: How The Gods Kill	Music
Star Trek VI - The Undiscovered Country	DVD
Stabbing Westward	Music
I Luciferi	Music
Danzig 2: Lucifuge	Music
Animositisomina	Music

Таблица Б.4 – Результаты экспериментов по построению онтологий пользователей набора данных Amazon с помощью семантического анализа понятий

Характеристики	ANIHXYXKPZL2H		A3DMZKTBIJURE1		A3IMNZSYDOTTU6	
	Без правил фильтрации	**С правилами фильтрации**	**Без правил фильтрации**	**С правилами фильтрации**	**Без правил фильтрации**	**С правилами фильтрации**
Время обработки	4568 сек	85 сек	3592 сек	70 сек	6891 сек	207 сек
Уровень базовых понятий (первый уровень)						
Общее количество базовых понятий	362	362	222	222	640	640
Среднее кол-во понятий на один текст	20.57	20.57	17.49	17.49	19.65	19.65
Максимальное кол-во понятий на один текст	28	28	24	24	31	31
Минимальное кол-во понятий на один текст	8	8	5	5	2	2
Среднее кол-во экземпляров текстов на одно понятие	2.6	2.6	3.45	3.45	2.29	2.29
Максимальное кол-во экземпляров текстов на одно понятие	30	30	34	34	43	43
Минимальное кол-во экземпляров текстов на одно понятие	1	1	1	1	1	1
Второй уровень понятий						
Общее число понятий	2273	53	1085	20	3418	91
Среднее кол-во экземпляров на понятие	3.44	3.64	4.37	3.8	2.94	5.01
Максимальное кол-во экземпляров на понятие	35	13	40	22	46	22
Минимальное кол-во экземпляров на понятие	1	1	1	1	1	1
Третий уровень понятий						
Общее число понятий	3929	4	2009	1	5291	7
Среднее кол-во экземпляров на понятие	4	2.25	5.08	1	3.65	7.71
Максимальное кол-во экземпляров на понятие	41	4	43	1	47	1

Минимальное кол-во экземпляров на понятие	1	1	1	1	1	2
Четвертый уровень понятий						
Общее число понятий	5347	-	2932	-	6424	1
Общее число уровней	3	3	25	3	21	4

ПРИЛОЖЕНИЕ В ПСЕВДОКОД АЛГОРИТМОВ ПОСТРОЕНИЯ СЕМАНТИЧЕСКОГО ПРОФИЛЯ ПОЛЬЗОВАТЕЛЯ И ВЫРАБОТКИ РЕКОМЕНДАЦИЙ

UserProfileModeling (U_t, $\delta_{\omega_a,\omega_{\bar{a}}}$, DT **)** :

Вход: данные о пользователе U_t;

константа $\delta_{\omega_a,\omega_{\bar{a}}}$;

узлы бинарного дерева решений DT;

Выход: профиль пользователя Pr_{U_t};

Начало

$Pr_{U_t} = \varnothing$;

I_{U_t} = *ExtractProducts (*U_t*)*; *// извлечение информации о продуктах, оцененных пользователем*

O_{U_t} = *SemanticConceptAnalisys (*I_{U_t}*)*; *// построение онтологии интересов пользователя*

Pr_{U_t} = *BuildProfileRules (*$O_{U_t}, I_{U_t}, \delta_{\omega_a,\omega_{\bar{a}}}, DT$*)*; *// агрегирование признаков, построение предикатов и правил*

CausalRulesFiltering (Pr_{U_t}*)*; *// причинная фильтрация правил класса*

RulesClustering (Pr_{U_t}*)*; *// кластеризация правил в каждом узле дерева*

вернуть Pr_{U_t};

Конец.

Рисунок В.1 - Псевдокод алгоритма, реализующего технологию построения семантического профиля пользователя

CausalRulesFiltering *(*$Pr_{U_t}, \delta^{\mu}_{\min}$*)* **:**

Вход: промежуточный профиль пользователя Pr_{U_t};

Выход: причинный профиль пользователя Pr_{U_t};

Начало

для всех правил $[\, P_k^i\left(\widetilde{X}_k^i\right) \rightarrow \omega_k \,] \in Pr_{U_t}$:

$\mu_k^i = \mu\left(P_k^i, \omega_k\right)$; *// вычислить значение метрики причинной связи для правила*

если $\left|\mu_k^i\right| < \delta^{\mu}_{\min}$

то удалить правило $[\, P_k^i\left(\widetilde{X}_k^i\right) \rightarrow \omega_k \,]$ из профиля Pr_{U_t};

вернуть Pr_{U_t};

Конец.

Рисунок В.2 - Псевдокод алгоритма, причинной фильтрации правил профиля пользователя

BuildProfileRule ($I_{U_t}, O_{U_t}, \delta_{\omega_a,\omega_{\bar{a}}}, \boldsymbol{DT}$) **:**

Вход: продукты, оцененные пользователем I_{U_t};
онтология интересов пользователя O_{U_t};
константа $\delta_{\omega_a,\omega_{\bar{a}}}$;
узлы бинарного дерева решений $\boldsymbol{DT}$.

Выход: промежуточный профиль пользователя $\boldsymbol{Pr}_{U_t}$

Начало

$\boldsymbol{Pr}_{U_t}\ Pr_{U_t} = \varnothing;$

$\boldsymbol{X} = FormAttributeList(I_{U_t}, O_{U_t})$ *// сформировать список всех атрибутов продуктов, включая понятия онтологии; подсчет статистических значений (полный проход по данным)*

для всех атрибутов $X^i \in \boldsymbol{X}$:

для всех узлов $(\omega_a, \omega_{\bar{a}})$ дерева решений $\boldsymbol{DT}$:

$\widetilde{\boldsymbol{X}}_a^i = \varnothing; \widetilde{\boldsymbol{X}}_{\bar{a}}^i = \varnothing;$ *// агрегированные множества значений атрибутов*

для всех значений $x_j^i \in \widetilde{\boldsymbol{X}}^i$ *// $\widetilde{\boldsymbol{X}}^i$ - множество всех значений атрибута X^i*

$\delta_{\omega_a,\omega_{\bar{a}}}^{x_j^i} = p(\omega_a|x_j^i) - p(\omega_a|x_j^i);$ *// $p(\omega_a|x_j^i), p(\omega_a|x_j^i)$ - вероятности*

если $\delta_{\omega_a,\omega_{\bar{a}}}^{x_j^i} \geq \delta_{\omega_a,\omega_{\bar{a}}}$

то $\widetilde{\boldsymbol{X}}_a^i = \widetilde{\boldsymbol{X}}_a^i + x_j^i;$

если $\delta_{\omega_a,\omega_{\bar{a}}}^{x_j^i} \leq -\delta_{\omega_a,\omega_{\bar{a}}}$

то $\widetilde{\boldsymbol{X}}_{\bar{a}}^i = \widetilde{\boldsymbol{X}}_{\bar{a}}^i + x_j^i;$

если $\widetilde{\boldsymbol{X}}_a^i \neq \varnothing$

то $\boldsymbol{P}_a^i(\widetilde{\boldsymbol{X}}_a^i) \rightarrow \omega_a;$

$\boldsymbol{Pr}_{U_t} = \boldsymbol{Pr}_{U_t} + [\boldsymbol{P}_a^i(\widetilde{\boldsymbol{X}}_a^i) \rightarrow \omega_a];$

если $\widetilde{\boldsymbol{X}}_{\bar{a}}^i \neq \varnothing$

то $\boldsymbol{P}_{\bar{a}}^i(\widetilde{\boldsymbol{X}}_{\bar{a}}^i) \rightarrow \omega_{\bar{a}};$

$\boldsymbol{Pr}_{U_t} = \boldsymbol{Pr}_{U_t} + [\boldsymbol{P}_{\bar{a}}^i(\widetilde{\boldsymbol{X}}_{\bar{a}}^i) \rightarrow \omega_{\bar{a}}];$

вернуть $\boldsymbol{Pr}_{U_t}$;

Конец.

Рисунок В.3 - Псевдокод алгоритма, реализующего технологию агрегирования значений атрибутов данных

RulesClustering $(\boldsymbol{Pr}_{U_t})$:

Вход: причинный профиль пользователя $\boldsymbol{Pr}_{U_t}$;

Выход: минимизированный профиль пользователя $\boldsymbol{Pr}_{U_t}^{\min}$;

Начало

$\boldsymbol{Pr}_{U_t}^{\min} = \varnothing$;

для всех узлов $(\omega_a, \omega_{\bar{a}})$ дерева решений $\boldsymbol{DT}$ профиля $\boldsymbol{Pr}_{U_t}$:

для $\omega_k \in \{\omega_a, \omega_{\bar{a}}\}$:

$\mathbf{Cor}_k = \left\| Cor_k^{i,j} \right\|_{i,j}$; *// матрица корреляции правил класса в узле дерева решений*

для всех пар правил ($\boldsymbol{P}_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right) \to \omega_k$, $\boldsymbol{P}_k^j\left(\widetilde{\boldsymbol{X}}_k^j\right) \to \omega_k$) в текущем узле:

$\mathbf{Cor}_k[i][j] = Cor\left(P_k^i, P_k^j\right)$ *// коэффициент корреляции между правилами*

$\boldsymbol{T_k}$ = *SelectThreshold (*$\mathbf{Cor}_k$*)* *// выбор порога разделения кластеров*

для всех $\mathbf{Cor}_k[i][j]$

если 1-$|\mathbf{Cor}_k[i][j]| < \boldsymbol{T_k}$

то $\mathbf{Cor}_k[i][j] = 0$;

$\boldsymbol{\Omega_k}$ = *DepthFirstSearch(*$\mathbf{Cor}_k$*)*; *// выделение связных компонент графа с помощью поиска в глубину; возвращает множество кластеров*

для всех кластеров $\Omega_k^j \in \boldsymbol{\Omega_k}$:

Sort(Ω_k^j*)*; *// отсортировать правила в кластере*

$\boldsymbol{Pr}_{U_t}^{\min} = \boldsymbol{Pr}_{U_t}^{\min} + \Omega_k^l$; *// добавить в мин. профиль первое по порядку правило кластера*

вернуть $\boldsymbol{Pr}_{U_t}^{\min}$;

Конец.

Рисунок В.4 - Псевдокод алгоритма, реализующего технологию кластеризацию причинных правил и минимизацию профиля пользователя

ContentFilteringRecommender(Pr_{U_t}, I_t); **:**

Вход: профиль целевого пользователя Pr_{U_t} ;

целевой продукт I_t ;

Выход: рейтинг

Начало

$\boldsymbol{X'} = FormProductAttributeList(I_t)$ *// сформировать список всех атрибутов целевого продукта*

$(\omega_a, \omega_{\bar{a}}) =$ ***DT(0)***; *// **DT(0)** - корневой узел дерева решений **DT***

выполнять

$\mu_a = 0;\ \mu_{\bar{a}} = 0;$

для всех правил $\boldsymbol{P}_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right) \to \omega_k \in$ узлу $(\omega_a, \omega_{\bar{a}})$

для всех атрибутов $X_t^j \in \boldsymbol{X'}$:

если *SameAs* (X_t^j, X^i) *// соответствует ли X_t^j атрибуту из профиля пользователя, для которого построен предикат $\boldsymbol{P}_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right)$*

то если $x_t^j \in \widetilde{\boldsymbol{X}}_k^i$

то если $\omega_k == \omega_a$

то $\mu_a = \mu_a + \mu_k^i$; *// - значение метрики причинной связи для правила $\boldsymbol{P}_k^i\left(\widetilde{\boldsymbol{X}}_k^i\right) \to \omega_k$*

иначе $\mu_{\bar{a}} = \mu_{\bar{a}} + \mu_k^i$;

если $\mu_a > \mu_{\bar{a}}$

то идти по ветке ω_a

если *IsNode(* $\boldsymbol{DT(\omega_a)}$ *)* *// IsNode(* $\boldsymbol{DT(\omega_a)}$ *) проверяет есть ли далее ветвления дерева или далее - один из листов **DT***

то $(\omega_a, \omega_{\bar{a}}) =$ $\boldsymbol{DT(\omega_a)}$; *// переходим в следующему узлу по ветке ω_a*

иначе $\omega_{U_t, I_t} = \omega_a$;

вернуть ω_{U_t, I_t} ;

иначе идти по ветке $\omega_{\bar{a}}$

если *IsNode (* $\boldsymbol{DT(\omega_{\bar{a}})}$ *)*

то $(\omega_a, \omega_{\bar{a}}) =$ $\boldsymbol{DT(\omega_{\bar{a}})}$

иначе $\omega_{U_t, I_t} = \omega_{\bar{a}}$;

вернуть ω_{U_t, I_t} ;

Конец.

Рисунок В.5 - Псевдокод алгоритма, реализующего технологию выработки рекомендаций методов фильтрации контента

UserClustering ($\boldsymbol{U}$, $\boldsymbol{\Xi}$, $\Delta_{\max}$, $\Delta_{\min}$, M_{min}, k); **:**

Вход: множество пользователей $\boldsymbol{U}$;

множество профилей пользователей $\boldsymbol{\Xi} = \boldsymbol{Pr}_{U_1}, \ldots, \boldsymbol{Pr}_{U_e}$;

пороги отсечения для построения кластеров $\Delta_{\max}$, $\Delta_{\min}$;

минимально допустимая мощность кластеров M_{min};

коэффициент нормировки профиля *k*;

Выход: множество кластеров $\mathfrak{I}$.

Начало

$\mathfrak{I} = \varnothing$;

$\boldsymbol{\Xi}'$ =*Norm*($\boldsymbol{\Xi}$, *k*); *// нормирование профилей*

$\mathbf{Sim}' = \left\| Sim'(U_k, U_l) \right\|_{k,l}$ *// матрица сходства пользователей*

для всех пар профилей $\boldsymbol{Pr}'_{U_k}$, $\boldsymbol{Pr}'_{U_l} \in \boldsymbol{\Xi}'$:

$\mathbf{Sim}'[k][l] = Sim'(\boldsymbol{Pr}'_{U_k}, \boldsymbol{Pr}'_{U_l})$;

$\boldsymbol{U}^c = \boldsymbol{U}$;

пока $\left|\boldsymbol{U}^c\right| > M_{min}$ **выполнять:**

$U_c = Random(\boldsymbol{U}^c)$; *// выбрать случайным образом пользователя из* $\boldsymbol{U}^c$

$\boldsymbol{UC}^c = \varnothing$;

$\boldsymbol{UC}^c = \boldsymbol{UC}^c + U_c$;

для всех $\mathbf{Sim}'[c][i]$:

если $\mathbf{Sim}'[c][i] \geq \Delta_{\min}$

то $\boldsymbol{UC}^c = \boldsymbol{UC}^c + U_i$;

$\mathbf{Sim}'_{UC} = \mathbf{Sim}'(\boldsymbol{UC}^c)$; *// извлечь из матрицы* $\mathbf{Sim}'$ *значения для всех пар пользователей из* $\boldsymbol{UC}^c$

$\mathfrak{I}_c = DepthFirstSearch(\mathbf{Sim}'_{UC})$; *// выделение связных компонент графа с помощью поиска в глубину; возвращает множество кластеров*

для всех кластеров $\boldsymbol{UC}^c_w$ из $\mathfrak{I}_c$:

если $\left|\boldsymbol{UC}^c_w\right| > M_{min}$

то $\mathfrak{I} = \mathfrak{I} + \boldsymbol{UC}^c_w$;

для всех пар U_i , $U_j \in \boldsymbol{UC}^c_w$:

если $\mathbf{Sim}'[i][j] \geq \Delta_{\max}$

удалить U_i , U_j из $\boldsymbol{U}^c$;

вернуть $\mathfrak{I}$;

Конец.

Рисунок В.5 - Псевдокод алгоритма предварительной кластеризации ользователей

Norm ($\boldsymbol{\Xi}$, *k*) **:**

Вход: множество профилей пользователей $\boldsymbol{\Xi} = \boldsymbol{Pr}_{U_1},..., \boldsymbol{Pr}_{U_e}$;

коэффициент нормировки профиля *k*;

Выход: множество нормированных профилей пользователей $\boldsymbol{\Xi}' = \boldsymbol{Pr}'_{U_1},..., \boldsymbol{Pr}'_{U_e}$;

Начало

$\boldsymbol{\Xi}' = \varnothing$;

для всех профилей $\boldsymbol{Pr}_{U_i} \in \boldsymbol{\Xi}$:

$\boldsymbol{Pr}'_{U_i} = \varnothing$;

Sort ($\boldsymbol{Pr}_{U_i}$); *// упорядочить правила из узлов дерева решений нижнего уровня по убыванию значения метрики причинной связи* $\mu(P_k^j, \omega_k)$

$\boldsymbol{\Sigma}$ = *Sum*($\boldsymbol{Pr}_{U_i}$); *// вычисляется сумма всех значений метрики* $\mu(P_k^j, \omega_k)$ *для таких правил*

j = 0;

пока $\boldsymbol{\Sigma}_C < k \cdot \boldsymbol{\Sigma}$:

$\boldsymbol{\Sigma}_C = \boldsymbol{\Sigma}_C + \mu(P_k^j, \omega_k)$;

$\boldsymbol{Pr}'_{U_i} = \boldsymbol{Pr}'_{U_i} + P_k^j$;

j = j + 1;

$\boldsymbol{\Xi}' = \boldsymbol{\Xi}' + \boldsymbol{Pr}'_{U_i}$;

Конец.

Рисунок В.6 - Псевдокод алгоритма нормализации профилей пользователей

ПРИЛОЖЕНИЕ Г АКТЫ ВНЕДРЕНИЯ

Общество с ограниченной ответственностью
“Исследовательский Центр Самсунг”
127018, Российская Федерация, Москва,
ул. Двинцев, 12, корп.1, офис № 1500
Тел: +7 (495) 797-2500; Факс: +7 (495) 797-2501
ОКПО 37285150 ОГРН 1117746985146 КПП 771501001

АКТ
о внедрении научных результатов,
полученных в диссертации Тушкановой Ольги Николаевны

Комиссия в составе:

- Рычагов Михаил Николаевич, д.ф.-м.н., директор Управления высокопроизводительных алгоритмов ООО «Исследовательский Центр Самсунг»;
- Невидомский Алексей Юрьевич, к.ф.-м.н., начальник Отдела интеллектуальной обработки данных ООО «Исследовательский Центр Самсунг»;

составила настоящий акт о том, что научные результаты, полученные Тушкановой Ольгой Николаевной в ее диссертационной работе на тему «Семантические структуры и причинные модели больших данных для принятия решений с приложением к рекомендательным системам», а именно:

1. Алгоритм построения онтологии данных с помощью технологии семантического анализа понятий;
2. Модель больших данных, которая представляет мета-свойства данных, их синтаксис и семантику в рамках единой структуры;
3. Мера оценки силы причинной связи – коэффициент регрессии случайных событий, - которая положена в основу ассоциативно-причинного анализа; и
4. Алгоритм поиска причинных зависимостей между атрибутами данных

использованы в работе «Многоагентные алгоритмы для кросс-доменных рекомендательных систем», руководитель Городецкий В.И., выполненной по контракту СПИИРАН с ООО «Исследовательский Центр Самсунг» в 2014 г.
Комиссия подтверждает практическую значимость и новизну результатов, полученных в данной работе.

Директор Управления высокопроизводительных алгоритмов
ООО «Исследовательский Центр Самсунг»,
д.ф.-м.н.
Рычагов Михаил Николаевич

Начальник Отдела интеллектуальной обработки данных
ООО «Исследовательский Центр Самсунг»,
к.ф.-м.н.
Невидомский Алексей Юрьевич

Подписи заверяю,
руководитель администрации
Навасардян С.В.

28 июня 2016г.