УДК 004.9
doi:10.15217/issn1684-8853.2015.5.91

АРХИТЕКТУРА ДЕЦЕНТРАЛИЗОВАННОЙ РЕКОМЕНДУЮЩЕЙ СИСТЕМЫ, ОСНОВАННОЙ НА ПРИМЕНЕНИИ ЛОКАЛЬНО-ЧУВСТВИТЕЛЬНОГО ХЕШИРОВАНИЯ

А. В. Пономарев[a], *канд. техн. наук, старший научный сотрудник*
[a]*Санкт-Петербургский институт информатики и автоматизации РАН, Санкт-Петербург, РФ*

Постановка проблемы: *рекомендующие системы широко используются в современных системах электронной коммерции, помогая пользователям ориентироваться в многообразии предлагаемых товаров и услуг. Наибольшее распространение получили централизованные архитектуры построения таких систем. Однако централизация влечет за собой ряд недостатков, среди которых — необходимость передачи пользователем сведений о предпочтениях стороне, осуществляющей эксплуатацию такой системы, и наличие единой точки отказа.* ***Цель:*** *построение децентрализованной рекомендующей системы, в которой для формирования рекомендаций используется сходство предпочтений пользователей (коллаборативная фильтрация), но полные сведения о предпочтениях хранятся только на узле, контролируемом самим пользователем, и не передаются другим узлам.* ***Результаты:*** *предложена архитектура децентрализованной рекомендующей системы, включающая структурированную одноранговую сеть, в которой каждый узел соответствует одному пользователю и хранит профиль его предпочтений, и специальный узел для информационного согласования участников сети. В качестве механизма, обеспечивающего, с одной стороны, поиск пользователей со схожими предпочтениями, а с другой стороны, ограниченное раскрытие информации о предпочтениях, используется локально-чувствительное хеширование. Для повышения уровня приватности пользователей в одноранговой сети применяется схема анонимизации.* ***Практическая значимость:*** *предложенный подход является достаточно универсальным и может быть использован для построения систем коллаборативной фильтрации в различных прикладных областях.*

Ключевые слова *— локально-чувствительное хеширование, одноранговые сети, рекомендующие системы, коллаборативная фильтрация.*

Введение

Большинство широко распространенных подходов к построению рекомендующих систем предполагают централизованную архитектуру. Важным достоинством централизации является существование широкого спектра техник моделирования предпочтений пользователей, предполагающих доступ к профилям всех пользователей (большинство реализаций метода ближайших соседей, разложение матрицы предпочтений и др.). Кроме того, при централизованном хранении информации о предпочтениях сторона, осуществляющая эксплуатацию рекомендующей системы, может производить всевозможные исследования этой информации, в том числе и не связанные напрямую с формированием рекомендаций.

Однако централизованная архитектура не свободна и от недостатков. Во-первых, в централизованных рекомендующих системах естественным образом возникает неоднозначная ситуация, касающаяся прав на информацию о предпочтениях. Как правило, пользователь не знает, какая именно информация о его действиях собирается, и не может извлечь (или уничтожить) эту информацию из системы. Более того, в случае прекращения функционирования сервиса, включавшего такую рекомендующую систему, соответствующая информация может быть безвозвратно утеряна. Во-вторых, централизация влечет за собой определенное разделение профиля пользователя. Пользователь может взаимодействовать с несколькими рекомендующими системами, предоставляя каждой лишь некоторые аспекты своих предпочтений. В результате предпочтения оказываются распределены между этими системами, хотя их консолидация могла бы улучшить качество рекомендаций. Наконец, централизация приводит к уменьшению надежности системы в целом за счет появления единой точки отказа, хотя в современных компьютерных системах этот недостаток в значительной мере ослабляется многоуровневыми схемами дублирования и репликации.

Децентрализация рекомендующей системы позволяет добиться двух важных целей:

— распределения функции формирования рекомендаций между пользователями и, как следствие, снятия необходимости в дорогостоящем сервере и повышения масштабируемости системы;

— повышения уровня приватности пользователей, поскольку исчезает необходимость в передаче предпочтений центральному серверу.

Есть несколько подходов к децентрализации рекомендующих систем. В этой статье развивается подход, в соответствии с которым пользователь хранит все сведения о предпочтениях только

на своем компьютере. Это полностью снимает упомянутую выше неоднозначную ситуацию, касающуюся прав на информацию о предпочтениях. Это также снимает проблему разделения профиля пользователя, поскольку все предпочтения сосредоточиваются на одном устройстве, контролируемом пользователем. При необходимости получения рекомендаций устройство посылает запросы на предоставление рекомендаций устройствам других пользователей.

И хотя в данном подходе устраняются все перечисленные недостатки централизованных рекомендующих систем, возникает и ряд трудностей. Главная проблема — ее решению посвящена и эта статья — состоит в реализации рекомендующего алгоритма, не требующего от пользователя передачи профиля своих предпочтений третьим лицам (участникам распределенной сети рекомендаций). Здесь следует сделать небольшое уточнение. Существует два основных класса рекомендующих систем: контентные системы и системы коллаборативной фильтрации. В контентных системах для формирования рекомендаций используются свойства самих объектов — система рекомендует объекты, похожие (с точки зрения некоторого формального представления) на те, что были полезны пользователю в прошлом. В системах же коллаборативной фильтрации сами свойства объектов не анализируются, система рекомендует те объекты, которые были высоко оценены пользователями, демонстрирующими схожие предпочтения. Конечно, трудности, связанные с децентрализацией, преимущественно касаются систем коллаборативной фильтрации, в основе которых лежит анализ сходства между пользователями, сопоставление их предпочтений, которое и осложняется распределенной организацией системы. Речь далее пойдет именно о таких системах, и под рекомендующей системой будет, если не оговорено иное, пониматься частный случай — система коллаборативной фильтрации.

В данной статье предложена архитектура рекомендующей системы, включающая структурированную одноранговую (P2P) сеть, в которой каждый узел соответствует одному пользователю и хранит профиль его предпочтений, и специальный узел для информационного согласования участников сети. Такой подход может быть классифицирован как гибридная одноранговая сеть, в которой часть функций выполняется исключительно посредством взаимодействия между равноправными узлами, а часть функций требует наличия специального узла. Предлагаемая архитектура обеспечивает ограниченное раскрытие предпочтений — не существует способа одновременно получить оценки, которые пользователь присвоил объектам, и сетевой адрес пользователя без глобального контроля над сетью.

Сама по себе задача построения рекомендующих систем, основанных на одноранговых сетях, не является новой. Существует определенный пласт работ, в которых эта задача ставится и предлагаются различные подходы к ее решению.

В системе P2Prec [1, 2] для распространения запросов и рекомендаций используется комбинация так называемого «дружеского» подхода к построению структуры сети (*friend-of-a-friend*), когда связи устанавливаются только между знакомыми пользователями, и лавинных алгоритмов распространения запросов.

В ряде описанных методов происходит построение оверлейной структуры, соответствующей близости интересов пользователей, поверх одноранговой сети [3, 4]. Рекомендации формируются поиском по оверлейной структуре на определенную глубину. Одним из распространенных алгоритмов такого «выравнивания» структуры сети под отношения между узлами является T-Man [5]. В данной работе сами оценки пользователя не раскрываются узлом сети, поэтому напрямую использовать T-Man или какой-либо схожий алгоритм нельзя из-за невозможности определить сходство узлов.

Другой подход заключается в применении алгоритмов случайного блуждания для поиска схожих узлов [6]. Для получения рекомендаций достаточно сформировать случайную выборку узлов сети, а затем использовать ближайшие, в соответствии с заданной мерой сходства, узлы из этой выборки [7].

Есть также работы, в которых авторы исследуют возможность применения структурированных одноранговых сетей для построения рекомендующих систем. Например, в работах [8, 9] оценки, присваиваемые объектам пользователями, сохраняются в распределенной хеш-таблице (*Distributed Hash Table* — DHT). Отличие предлагаемого в данной статье подхода заключается в том, что в распределенную хеш-таблицу помещаются не оценки, а сами узлы, и механизм быстрого поиска по этой таблице используется для поиска узлов, соответствующих пользователям со схожими интересами.

Формирование рекомендаций с помощью локально-чувствительного хеширования

Локально-чувствительное хеширование (ЛЧХ) — это широко распространенный метод приближенного решения задачи поиска k ближайших соседей. Идея метода состоит в построении такой хеш-функции многомерных объектов, чтобы схожие объекты с высокой вероятностью получали одинаковое значение хеш-функции. Методы и алгоритмы поиска ближайших соседей находят широкое применение при построении

рекомендующих систем. Одним из основополагающих допущений коллаборативной фильтрации является представление о том, что пользователи, имевшие схожие предпочтения в прошлом, вероятно, имеют схожие предпочтения в настоящем, что может быть использовано при формировании рекомендаций. Если представить предпочтения пользователя в виде вектора и ввести соответствующую меру близости, то поиск пользователей со схожими интересами можно будет интерпретировать как поиск ближайших соседей.

В этом подразделе приводится формальное описание коллаборативной фильтрации, основанной на локально-чувствительном хешировании.

Идея ЛЧХ

Описание базовых идей ЛЧХ приводится в соответствии с работой [10]. Пусть $d_1 < d_2$ — два значения расстояния в соответствии с некоторой мерой d. Семейство функций F называется (d_1, d_2, p_1, p_2)-чувствительным, если для каждой функции f в F:

— если $d(a, b) \le d_1$, то вероятность того, что $f(a) = f(b)$, не меньше p_1;

— если $d(a, b) \ge d_2$, то вероятность того, что $f(a) = f(b)$, не больше p_2.

Важной идеей в теории ЛЧХ является *усиление*, в основе которого лежат понятия И-конструкции и ИЛИ-конструкции, определенные ниже.

Пусть задано (d_1, d_2, p_1, p_2)-чувствительное семейство функций F, новое семейство функций F' может быть получено посредством И-конструкции или ИЛИ-конструкции.

И-конструкция F' определяется следующим образом. Каждый член F' состоит из r членов F. Если f в F' и f получена из множества $\{f_1, f_2, ..., f_r\}$ членов F, то $f(x) = f(y)$ тогда и только тогда, когда $f_i(x) = f_i(y)$ для всех $i \in \{1, ..., r\}$. Поскольку члены F' независимо выбираются из F, F' является (d_1, d_2, p_1^r, p_2^r)-чувствительным семейством функций [10].

ИЛИ-конструкция F' определяется следующим образом. Каждый член F' состоит из b членов F. Если f в F' и f получена из множества $\{f_1, f_2, ..., f_b\}$ членов F, то $f(x) = f(y)$ тогда и только тогда, когда существует $i \in \{1, ..., b\}$ такой, что $f_i(x) = f_i(y)$. Аналогично F' является $(d_1, d_2, 1 - (1 - p_1)^b, 1 - (1 - p_2)^b)$-чувствительным семейством.

Как правило, желательно, чтобы p_1 было большим, насколько возможно, а p_2 маленьким, насколько возможно. Если $p_1 < 1$, то существует вероятность того, что схожие объекты будут иметь различные значения. С другой стороны, если $p_2 > 0$, есть вероятность, что значительно различающиеся объекты получат одинаковое значение хеш-функции. Следовательно, семейство F следует выбирать таким образом, чтобы p_1 было близко к 1, а p_2 близко к 0. Существует определенный набор хорошо изученных семейств локально-чувствительных функций, механизм усиления применяется в том случае, если только лишь средствами выбранного семейства не удается достичь желаемых вероятностей. Если семейство F^{Ar} получено И-конструкцией r функций из семейства F, а G затем получено ИЛИ-конструкцией b функций из семейства F^{Ar}, то G является $(d_1, d_2, 1 - (1 - p_1^r)^b, 1 - (1 - p_2^r)^b)$-чувствительным семейством. Неформально И-конструкция снижает изначально невысокую вероятность p_2, а последующая ИЛИ-конструкция повышает изначально высокую вероятность p_1.

Идея поиска ближайших соседей с помощью ЛЧХ описана, например, в работах [10, 11]. В первую очередь выбирается семейство F и создаются b обычных хеш-таблиц. Каждая таблица соответствует одной хеш-функции f_i^{Ar}, $i = 1, ..., b$, где f_i^{Ar} — И-конструкция r случайных функций из F. Каждый объект x помещается в каждую из b хеш-таблиц. Ключом является значение функции $f_i^{Ar}(x)$, а значением — идентификатор объекта x или сам объект, в зависимости от задачи. При поиске ближайших соседей объекта y вычисляются $f_i^{Ar}(y)$, $i = 1, ..., b$; все объекты, извлеченные из хеш-таблиц по полученным ключам, образуют множество кандидатов. Реальная близость оценивается уже с применением строгой меры, и происходит отсев ложно-положительных соседей из множества кандидатов.

Выбор семейства хеш-функций зависит от представления данных и функции расстояния d. Для хеммингова расстояния, например, часто применяется хеш-функция, осуществляющая выборку отдельных битов [12], для косинусной меры — метод случайных проекций [13].

В данной работе используется метод случайных проекций, т. е. функция f из семейства F соответствует одной случайной гиперплоскости; функция принимает значение 1, если хешируемая точка находится над гиперплоскостью, и 0 в противном случае.

Формирование рекомендаций

В системах коллаборативной фильтрации, основанных на сходстве пользователей (*user-based collaborative filtering*), рекомендации формируются с учетом того, в какой мере совпадают оценки пользователей, присвоенные одним и тем же объектам.

Формально, пусть r_{uj} — оценка, присвоенная объекту j пользователем u и выражающая степень того, насколько пользователю u интересен объект j или какова субъективная ценность объекта j для пользователя u. Пусть U — множество пользователей; I — множество объектов; I_u — множество объектов, оцененных пользователем u;

I_{uv} — множество объектов, оцененных как пользователем u, так и пользователем v. Методы, основанные на сходстве, используют меру близости между пользователями, определяемую посредством сопоставления оценок, присвоенных пользователями одним и тем же объектам: $(\text{sim}(u, v) = f_s(\{r_{uj}, j \in I_{uv}\}))$, и пытаются предсказать неизвестную оценку r^*_{uj} на основе известных оценок r_{vj} и сходства между пользователями $\text{sim}(u, v)$.

В данной статье используется косинусная мера сходства между пользователями:

$$\text{sim}(u,v) = \frac{\sum\limits_{I_{uv}} r_{uj} r_{vj}}{\sqrt{\sum\limits_{I_{uv}} r_{uj}^2} \sqrt{\sum\limits_{I_{uv}} r_{vj}^2}}.$$

Оценки пользователей нормализуются таким образом, что $r_{uj} = 1$ соответствует строго положительному отношению пользователя u к объекту j, а $r_{uj} = -1$ — строго отрицательному.

Рекомендующая система, использующая ЛЧХ, реализует поиск ближайших соседей. По известному набору значений хеш-функций для некоторого пользователя u система проверяет соответствующие хеш-таблицы и извлекает из них идентификаторы всех пользователей, чьи предпочтения вероятно похожи (в силу свойства хеш-функции) на предпочтения пользователя u. Затем может быть оценено точное сходство между пользователями, и объекты, высоко оцененные пользователями, похожими на u, будут рекомендованы u.

В предлагаемой системе точное значение сходства не вычисляется, поскольку это привело бы к раскрытию профиля пользователя. Вместо этого вводится приближенная мера сходства $s'(u, v)$, определяемая как количество локально-чувствительных хеш-функций, чьи значения совпали для пользователей u и v. Алгоритм рекомендации, во-первых, осуществляет поиск всех пользователей Q_u, которые могут быть соседями пользователя u, и вычисляет $s'(u, v)$ (где $v \in Q_u$). Каждому из кандидатов $v \in Q_u$ посылается запрос на список рекомендаций R_v. Предлагаемый алгоритм и система в целом предсказывают неизвестные оценки r^*_{uj}, вместо этого проводится ранжирование всех объектов, которые были рекомендованы кандидатами в соответствии с оценкой $\tilde{a}_{ui}$ объекта i для пользователя u, определяемой выражением

$$\tilde{a}_{ui} = \sum_{v \in Q_u} s'(u,v) P^R_{vi}.$$

Здесь P^R_{vi} — индикаторная функция, осуществляющая проверку того, есть ли объект i в множестве объектов, рекомендованных пользователем v:

$$P^R_{vi} = \begin{cases} 1, i \in R_v \\ 0, i \notin R_v \end{cases}.$$

Таким образом, в предлагаемой архитектуре профиль пользователя u представляется множеством пар (i, r_{ui}), где i — идентификаторы объектов. Каждая из b локально-чувствительных хеш-функций представлена r векторами размерности ($|I|$). После применения всех этих хеш-функций получается b r-мерных бинарных векторов. Полученные векторы сохраняются в хеш-таблице. Во время формирования рекомендаций производится b операций поиска в хеш-таблице, а затем каждому «кандидату», извлеченному из хеш-таблицы, посылается запрос на формирование рекомендаций. Список рекомендаций упорядочивается по значению $\tilde{a}_{ui}$.

Значения b и r являются параметрами алгоритма формирования рекомендаций. В разделе «Экспериментальное исследование» производится экспериментальная оценка того, как значения этих параметров влияют на качество рекомендаций.

Архитектура системы

Предлагаемая гибридная архитектура позволяет осуществлять обмен рекомендациями с ограниченным раскрытием предпочтений пользователя. В этом разделе описаны основные компоненты системы и сценарии их взаимодействия.

Сценарии

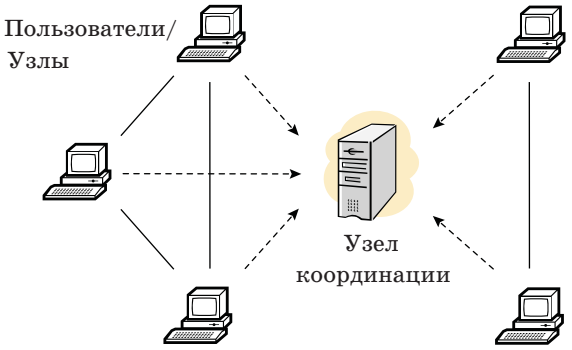
Предлагаемая система рассчитана на обеспечение двух вариантов использования: а) оценка потенциальной привлекательности объекта (или группы объектов) для данного пользователя; б) запрос рекомендаций.

Оценка потенциальной привлекательности объекта инициируется, когда необходимо проверить, может ли данный неизвестный объект быть интересен пользователю (с точки зрения логики, заложенной в систему). Пользователь передает системе идентификатор объекта, а рекомендующая система в ответ должна сообщить предполагаемую оценку привлекательности этого объекта для пользователя.

Запрос рекомендаций инициируется, когда необходимо сформировать набор новых, неизвестных пользователю объектов, которые могут оказаться ему интересны.

Компоненты

В соответствии с предлагаемым подходом рекомендующая система состоит из двух частей: одноранговой (P2P) сети рекомендаций и узла



■ ***Рис. 1.*** Связи между узлами в предлагаемой архитектуре

координации (рис. 1). Присутствие узла координации нарушает концептуальную чистоту одноранговой системы, превращая ее в гибридную, однако этот узел не играет важной роли в основных сценариях, перечисленных выше, его роль заключается в синхронизации вспомогательной информации между узлами сети.

Показаны два типа связей между узлами: связи между схожими узлами, образующими одноранговую сеть, отображены сплошными линиями; периодические связи узлов сети с узлом координации, устанавливаемые для обмена вспомогательной информацией, отображены пунктирными линиями.

1. *Одноранговая сеть рекомендаций.* В предлагаемом подходе каждый пользователь соответствует ровно одному узлу сети. На этом узле хранится вся информация о предпочтениях пользователя (в первую очередь, оценки объектов), причем узел не передает эту информацию другим узлам, он может передавать только значения локально-чувствительных хеш-функций, вычисленных от этой информации, для поиска схожих пользователей, к которым можно будет «обращаться» за получением рекомендаций.

Одноранговая сеть основана на использовании DHT [14] — распространенном подходе к построению так называемых структурированных одноранговых сетей. DHT — это класс систем, обеспечивающих хранение коллекции пар ключ — значение, распределенной по различным узлам сети, с учетом миграции фрагментов при выходе узла из состава сети.

Классические реализации подхода DHT обладают рядом уязвимостей. Для их преодоления разработано несколько анонимизированных реализаций DHT. Предлагаемая архитектура основывается на Octopus [15] — одной из таких анонимизированных реализаций. В основе таких реализаций, как правило, лежит идея построения цепочек анонимизации вместо непосредственного обращения к другому узлу сети, причем каждый узел, лежащий в такой цепочке, имеет информацию только о соседних узлах цепочки. Таким образом, становится значительно сложнее установить, от какого же именно узла исходил запрос.

В предлагаемой системе DHT используется для хранения хеш-таблиц, применяемых в целях поиска ближайших соседей, как описано в предыдущем разделе. Каждая пара ключ — значение, хранимая в DHT, содержит информацию об одном значении локально-чувствительной хеш-функции и список узлов, соответствующих этому значению. Как уже указывалось, для поиска ближайших соседей необходимо несколько (b) хеш-таблиц. Каждая из b таблиц использует свою локально-чувствительную хеш-функцию. В данной системе предлагается хранить все b хеш-таблиц в одной DHT. Для этого ключ должен включать в себя уникальный идентификатор локально-чувствительной хеш-функции и значение этой функции.

Перед тем как включить записи в DHT, узел создает анонимизированную цепочку и использует спецификатор окончания этой цепочки как свой адрес, передаваемый другим узлам. Эти анонимизированные пути создаются при каждом очередном подключении узла к сети заново, следовательно, во время каждой новой сессии у узла оказывается новый внешний идентификатор.

Поскольку предпочтения пользователя, выраженные в оценках, присвоенных этим пользователем различным объектам, достаточно статичны, предполагается хранение ссылок на внешние «публичные» идентификаторы узлов, соответствующих пользователям со схожими интересами. Таким образом, поверх одноранговой сети образуется оверлейная сеть, сформированная ссылками между узлами пользователей со схожими интересами. Следует иметь в виду, что ссылки между вершинами в этой оверлейной сети являются не идентификаторами узлов P2P-сети, а «входами» в анонимизированные пути, ведущие к ним.

2. *Узел координации.* Распределенный характер предлагаемой системы является причиной следующей технической сложности. Для корректного вычисления локально-чувствительных хеш-функций необходимо, чтобы сами хеш-функции (т. е. гиперплоскости, которыми они представляются) были одинаковы на всех узлах. Для согласования параметров этих функций все узлы сети должны использовать один и тот же порядок следования объектов, поскольку размерность гиперплоскостей совпадает с количеством объектов и с длиной вектора пользовательских оценок. Задача поддержания глобального состояния в одноранговой сети является нетривиальной [16–18]. В предлагаемой системе для ее решения используется подход, схожий с предложенным

в статье [19] и заключающийся в отказе от чисто однорангового устройства сети. Задачей узла координации является сбор всех объектов (о которых сообщают пользователи), поддержка отношения порядка между их идентификаторами и генерация локально-чувствительных функций. Таким образом, каждый узел должен соединиться с узлом координации для двух целей: во-первых, для регистрации нового, ранее неизвестного объекта; во-вторых, для получения нового набора локально-чувствительных хеш-функций. Следует заметить, что нет необходимости генерировать новый набор хеш-функций после обнаружения каждого нового объекта. При использовании «устаревшего» набора функций получение рекомендаций оказывается возможным, но их качество постепенно ухудшается с ростом расхождения между используемым и актуальным наборами. Таким образом, каждый узел накапливает новые объекты, посылает накопленный пакет объектов узлу координации, а в ответ получает обновленный набор хеш-функций.

Экспериментальное исследование

Экспериментальное исследование предлагаемого подхода было произведено с использованием набора данных MovieLens 100k, выложенного в открытый доступ исследовательской лабораторией GroupLens Research [20]. Этот набор содержит 100 000 оценок, присвоенных 943 пользователями 1682 фильмам.

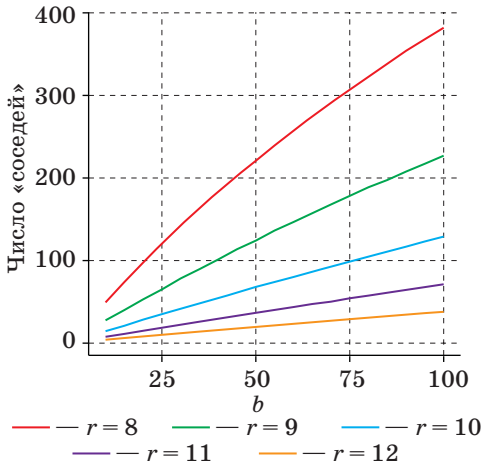
В ходе экспериментального исследования преследовались две цели. Во-первых, получить практическую информацию о количественных характеристиках подхода и оценить временную и пространственную сложность рекомендующих систем, основанных на ЛЧХ в DHT-сетях. Во-вторых, оценить качество рекомендаций по сравнению с широко распространенными альтернативными алгоритмами.

Временная и пространственная сложность

Как уже было отмечено, параметрами предлагаемого алгоритма формирования рекомендаций являются b (количество хеш-функций) и r (количество гиперплоскостей в каждой функции). Значения этих параметров оказывают существенное влияние как на время получения рекомендаций, так и на их качество.

Каждый узел помещает информацию о себе DHT b раз (по одному значению каждой из хеш-функций), следовательно, размер DHT равен Nb, где N — количество узлов, а это означает, что в среднем в узле размещено b записей DHT.

Поиск ближайших соседей требует b операций извлечения из хеш-таблицы, а значит, требует $O(b \log(N))$ взаимодействий между узлами.



■ ***Рис. 2.*** Зависимость среднего числа «соседей» от количества b хеш-функций и их размерности r

Наиболее важным параметром, хотя и не явным, а косвенным, является фактическое количество «соседей», которые обнаруживаются в результате извлечения b значений из хеш-таблиц, поскольку оно определяет количество сетевых запросов на формирование рекомендаций, и чем оно меньше (при определенном уровне качества), тем лучше.

На рис. 2 показана зависимость между параметрами b и r рекомендующей системы и средним количеством «соседей», найденных в результате поиска по хеш-таблице. Видно, что количество «соседей» увеличивается с ростом b, а темп увеличения существенно зависит от размерности хеш-функций. Это ожидаемое поведение, поскольку невысокая размерность хеш-функций и большое количество «альтернативных» хеш-функций делают процедуру поиска очень грубой. В данном исследовании предполагается, что количество хеш-функций должно быть менее 100, а количество «соседей» — менее 50.

Принимая во внимание изложенные соображения, для исследования качества рекомендаций были выбраны три конфигурации: (r = 12, b = 100), (r = 10, b = 35), (r = 8, b = 10), — так как в каждой из этих конфигураций количество «соседей» приблизительно равно 50.

Качество рекомендаций

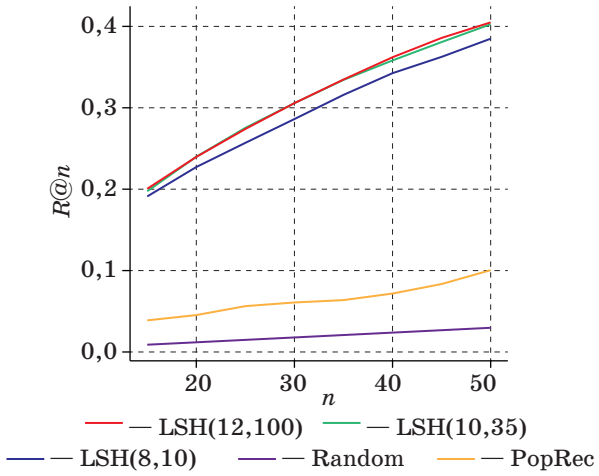
Поскольку предлагаемый алгоритм формирования рекомендаций не осуществляет предсказание самих оценок, традиционные методы оценки качества (например, среднеквадратическое отклонение между прогнозными и реальными оценками) оказываются неприменимы. Алгоритм оценивался по полноте (*recall*) — мере качества, широко распространенной при сравнении алгоритмов получения лучших n рекомендаций.

В частности, для оценки качества использовался подход, описанный в работе [21]. Набор данных с оценками был разделен на два: обучающую выборку и тестовую выборку в пропорции 80/20. Обучающая выборка использовалась для заполнения хеш-таблицы, затем для каждой высокой оценки (4 или 5) из тестовой выборки осуществлялась проверка — попадает ли соответствующий объект в список из n наиболее рекомендованных для соответствующего пользователя. Результатом такой проверки может быть либо 1 (если объект попадает в n рекомендованных), либо 0 (в противном случае). Сумма этих результатов по всему тестовому набору дает N_p. Полнота при n рекомендованных объектах (обозначаемая $R@n$) вычислялась по формуле

$$R@n = \frac{N_p}{N_H},$$

где N_H — количество высоких оценок в тестовой выборке. Другими словами, эта величина может интерпретироваться как вероятность того, что случайным образом выбранный объект с высокой оценкой действительно будет рекомендован данным алгоритмом (включен в n рекомендаций).

Полнота рекомендаций предложенной системы была сопоставлена с полнотой рекомендаций, полученных с помощью других (неперсонализированных) алгоритмов. Во-первых, случайного рекомендующего алгоритма (Random), предлагающего пользователю n случайных объектов, во-вторых, алгоритма рекомендации популярных объектов (PopRec), предлагающего объекты, получившие наибольшее количество оценок. Полнота каждого из названных рекомендующих алгоритмов при различных значениях n показана на рис. 3.



■ ***Рис. 3.*** Сравнение $R@n$ различных методов формирования рекомендаций

Все протестированные варианты формирования рекомендаций с помощью ЛЧХ дают примерно одинаковые результаты. Это можно объяснить тем, что во всех этих вариантах количество «соседей» оказывается примерно одинаковым (около 50, см. рис. 2).

Кроме того, можно увидеть, что предлагаемый алгоритм формирования рекомендаций существенно превосходит неперсонализированные рекомендующие алгоритмы по полноте.

Заключение

В статье предложена архитектура гибридной одноранговой рекомендующей системы, основанной на ЛЧХ профиля пользователей. В предлагаемой архитектуре приватность пользователей обеспечивается тем, что обмен сведениями о предпочтениях пользователей происходит только анонимизированным образом и только в виде значений хеш-функций. Было произведено экспериментальное исследование предлагаемого подхода с использованием одного из широко распространенных наборов данных и показано, что оценка полноты (*recall*) рекомендаций, формируемых с помощью предлагаемого подхода, существенно выше, чем для неперсонализированных алгоритмов формирования рекомендаций.

Однако следует отметить и некоторые ограничения предлагаемого подхода. Во-первых, из-за ограничений систем класса DHT он оказывается не подходящим для сетей с динамично изменяющейся структурой. Во-вторых, подобное устройство рекомендующей системы не подходит для областей, в которых часто появляются новые объекты (например, новостные сообщения), из-за необходимости распространения информации об объектах среди всех узлов сети.

В будущем планируется рассмотреть альтернативные решения по согласованию глобального состояния сети без нарушения одноранговой организации.

Работа выполнена при финансовой поддержке РФФИ (гранты № 13-07-00271, 13-07-00039, 14-07-00345), Президиума РАН (проект № 213) и отделения нанотехнологий и информационных технологий РАН (проект № 2.2).

Литература

1. **Draidi F., Pacitti E., Kemme B.** P2Prec: A P2P Recommendation System for Large-Scale Data Sharing // Journal of Transactions on Large-Scale Data and Knowledge-Centered Systems (TLDKS). 2011. Vol. 3. P. 87–116.
2. **Draidi F.,** et al. P2Prec: a Social-Based P2P Recommendation System //Proc. of the 20th ACM Intern.

Conf. on Information and Knowledge Management. 2011. P. 2593–2596.
3. **Kermarrec A.-M.,** et al. Application of Random Walks to Decentralized Recommendation Systems // Proc. of the 14th Intern. Conf. on Principles of Distributed Systems. 2010. P. 48–63.
4. **Pitsilis G., Marshall L.** A Trust-Enabled P2P Recommendation System // Proc. 15th IEEE Intern. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. 2006. P. 59–64.
5. **Jelasity M., Montresor A., Babaoglu O.** T-Man: Gossip-Based Fast Overlay Topology Construction // Computer Networks. Aug. 2009. Vol. 53. N 13. P. 2321–2339.
6. **Tveit A.** Peer-to-Peer Based Recommendations for Mobile Commerce // Proc. 1st Intern. Workshop on Mobile Commerce (WMC'01), ACM. 2001. P. 26–29.
7. **Bakker A., Ogston E., van Steen M.** Collaborative Filtering Using Random Neighbours in Peer-to-Peer Networks // Workshop on Complex Networks in Information & Knowledge Management. 2009. P. 67–74.
8. **Han P.,** et al. A Scalable P2P Recommendation System Based on Distributed Collaborative Filtering // Expert Systems with Applications. 2004. N 27(2). P. 203–210.
9. **Hecht F.,** et al. Radiommendation: P2P On-Line Radio with a Distributed Recommendation System // Proc. of the IEEE 12th Intern. Conf. on Peer-to-Peer Computing. 2012. P. 73–74.
10. **Rajaraman A., Ullman J.** Mining of Massive Datasets. — Cambridge University Press, 2012. — 326 p.
11. **Slanley M., Casey M.** Locality-Sensitive Hashing for Finding Nearest Neighbors // IEEE Signal Processing Magazine. Mar. 2008. Vol. 25. N 2. P. 128–131.
12. **Indyk P., Motwani R.** Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality // STOC'98: Proc. of the 30th Symp. on Theory of Computing. 1998. P. 604–613.
13. **Charikar M. S.** Similarity Estimation Techniques from Rounding Algorithms // STOC'02: Proc. of the 34th Annual ACM Symp. on Theory of Computing. 2002. P. 380–388.
14. **Korzun D., Gurtov A.** Structured Peer-to-Peer Systems. Fundamentals of Hierarchical Organization, Routing, Scaling and Security. — Springer, 2013. — 366 p.
15. **Wang Q., Borisov N.** Octopus: A Secure and Anonymous DHT Lookup // Proc. of the IEEE 32nd Intern. Conf. on Distributed Computing Systems. 2012. P. 325–334.
16. **Chen X.,** et al. SCOPE: Scalable Consistency Maintenance in Structured P2P Systems // Proc. of IEEE INFOCOM. 2005. P. 1502–1513.
17. **Hu Y., Bhuyan L. N., Feng M.** Maintaining Data Consistency in Structured P2P Systems // IEEE Transactions on Parallel and Distributed Systems. 2012. Vol. 23. Iss. 11. P. 2125–2137.
18. **Oster G.,** et al. Data Consistency for P2P Collaborative Editing // Proc. of the 20th Anniversary Conf. on Computer Supported Cooperative Work. 2006. P. 259–268.
19. **Mastroianni C., Pirro G., Talia D.** Data Consistency and Peer Synchronization in Cooperative P2P Environments. — Technical Report, 2008. — 16 p.
20. **MovieLens** dataset. http://grouplens.org/datasets/movielens/ (дата обращения: 17.05.2015).
21. **Cremonesi P., Koren Y., Turrin R.** Performance of Recommender Algorithms on Top-N Recommendation Tasks // Proc. of the Fourth ACM Conf. on Recommender Systems (RecSys '10). ACM, New York, NY, USA, 2010. P. 39–46.

UDC 004.9
doi:10.15217/issn1684-8853.2015.5.91

Decentralized Recommendation System Architecture Based on Locality-Sensitive Hashing

Ponomarev A. V.[a], PhD, Senior Researcher, ponomarev@iias.spb.su
[a]Saint-Petersburg Institute for Informatics and Automation of RAS, 39, 14 Line, V. O., 199178, Saint-Petersburg, Russian Federation

Purpose: Recommendation systems are widely used in modern e-commerce systems to help users make their ways in a vast variety of offered goods and services. Most of the modern recommendation system approaches are centralized. However, centralized recommendation have two primary disadvantages: the necessity for users to share their preferences and a single point of failure. The goal of this work is developing a decentralized recommendation system which employs user similarity (collaborative filtering) but holds all the user preferences only on the user's network node. **Results:**An architecture is proposed for a decentralized recommendation system. It includes a structured peer-to-peer network in which each node corresponds to one user and stores this user's preferences, and a special node used for the coordination of peer-to-peer nodes in some scenarios. To find users with similar interests and, in the same time, restrict the sharing of preferences, a locality-sensitive hashing is employed. For a higher level of privacy, the network uses an anonymization scheme. **Practical relevance:** The proposed approach is universal, as it relies only on ratings, and can be used to build collaborative filtering systems in various domains.
Keywords — Locality-Sensitive Hashing, Peer-To-Peer, Recommendation Systems, Collaborative Filtering.

References

1. Draidi F., Pacitti E., and Kemme B. P2Prec: A P2P Recommendation System for Large-Scale Data Sharing. *Journal of Transactions on Large-Scale Data and Knowledge-Centered Systems (TLDKS)*, 2011, vol. 3, pp. 87–116.

2. Draidi F., et al. P2Prec: A Social-Based P2P Recommendation System. *Proc. of the 20th ACM Intern. Conf. on Information and Knowledge Management*, 2011, pp. 2593–2596.
3. Kermarrec A.-M., et al. Application of Random Walks to Decentralized Recommendation Systems. *Proc. of the 14th Intern. Conf. on Principles of Distributed Systems*, 2010, pp. 48–63.
4. Pitsilis G., Marshall L. A Trust-Enabled P2P Recommendation System. *Proc. 15th IEEE Intern. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2006, pp. 59–64.
5. Jelasity M., Montresor A., Babaoglu O. T-Man: Gossip-Based Fast Overlay Topology Construction. *Computer Networks*, August 2009, vol. 53, no. 13, pp. 2321–2339.
6. Tveit A. Peer-to-Peer Based Recommendations for Mobile Commerce. *Proc. 1st Intern. Workshop on Mobile Commerce (WMC'01)*, ACM, 2001, pp. 26–29.
7. Bakker A., Ogston E., and van Steen M. Collaborative Filtering Using Random Neighbours in Peer-to-Peer Networks. *Workshop on Complex Networks in Information & Knowledge Management*, 2009, pp. 67–74.
8. Han P., et al. A Scalable P2P Recommendation System Based On Distributed Collaborative Filtering. *Expert Systems with Applications*, 2004, no. 27(2), pp. 203–210.
9. Hecht F., et al. Radiommendation: P2P On-Line Radio with a Distributed Recommendation System. *Proc. of the IEEE 12th Intern. Conf. on Peer-to-Peer Computing*, 2012, pp. 73–74.
10. Rajaraman A., Ullman J. *Mining of Massive Datasets*. Cambridge University Press, 2012. 326 p.
11. Slanley M., Casey M. Locality-Sensitive Hashing for Finding Nearest Neighbors. *IEEE Signal Processing Magazine*, March 2008, vol. 25, no. 2, pp. 128–131.
12. Indyk P., Motwani R. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. *STOC'98 Proc. of the 30th Symp. on Theory of Computing*, 1998, pp. 604–613.
13. Charikar M. S. Similarity Estimation Techniques from Rounding Algorithms. *STOC'02 Proc. of the 34th annual ACM Symp. on Theory of Computing*, 2002, pp. 380–388.
14. Korzun D., Gurtov A. *Structured Peer-to-Peer Systems. Fundamentals of Hierarchical Organization, Routing, Scaling and Security*. Springer, 2013. 366 p.
15. Wang Q., Borisov N. Octopus: A Secure and Anonymous DHT Lookup. *Proc. of the IEEE 32nd Intern. Conf. on Distributed Computing Systems*, 2012, pp. 325–334.
16. Chen X., et al. SCOPE: Scalable Consistency Maintenance in Structured P2P Systems. *Proc. of IEEE INFOCOM*, 2005, pp. 1502–1513.
17. Hu Y., Bhuyan L. N., and Feng M. Maintaining Data Consistency in Structured P2P Systems. *IEEE Transactions on Parallel and Distributed Systems*, 2012, vol. 23, iss. 11, pp. 2125–2137.
18. Oster G., et al. Data Consistency for P2P Collaborative Editing. *Proc. of the 20th Anniversary Conf. on Computer Supported Cooperative Work*, 2006, pp. 259–268.
19. Mastroianni C., Pirro G., and Talia D. *Data Consistency and Peer Synchronization in Cooperative P2P Environments*. Technical Report, 2008. 19 p.
20. MovieLens dataset. Available at: http://grouplens.org/datasets/movielens/ (accessed 17 May 2015).
21. Cremonesi P., Koren Y., Turrin R. Performance of Recommender Algorithms on Top-N Recommendation Tasks. *Proc. of the fourth ACM Conf. on Recommender Systems (RecSys '10)*, ACM, New York, NY, USA, 2010, pp. 39–46.

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (ius.spb@gmail.com).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию. Рукописи не возвращаются.

Редакция журнала напоминает, что ответственность за достоверность и точность рекламных материалов несут рекламодатели.