МЕТОД ЭМПИРИЧЕСКИХ ВЕРОЯТНОСТЕЙ: АВТОМАТИЧЕСКАЯ СИСТЕМА ДЛЯ РЕКОМЕНДАЦИИ СЛЕДУЮЩИХ ДЕСЯТИ ЛЕКЦИЙ КУРСА ПОСЛЕ ПРОСМОТРА ТРЕХ ДАННЫХ ЛЕКЦИЙ

В.Н. Никулин, *кандидат физико-математических наук, доцент кафедры математических методов в экономике Вятского государственного университета*

С.А. Палешева, *студентка кафедры математических методов в экономике Вятского государственного университета*

Д.С. Зубарева, *студентка кафедры математических методов в экономике Вятского государственного университета*

E-mail: vnikulin.uq@gmail.com, s.palesheva@gmail.com, zubarevadasha@yandex.ru
Адрес: г. Киров, ул. Московская, д. 36

В статье представлен алгоритм, который был награжден призом за третий лучший результат, продемонстрированный в ходе международного соревнования по анализу данных VideoLectures.Net ECML/PKDD 2011 (Track 2). Мы предлагаем использовать две лекции (взятые из тройки данных лекций), для того чтобы определить направление прогноза. Соответствие всего предсказанного набора вычисляется согласно оставшейся третьей лекции.

Ключевые слова: рекомендательные системы, прогнозирование, оценки популярности, ансамбли и элементарные классификаторы, бэггинг, анализ данных.

1. Введение

VideoLectures.Net[1] – это открытый и общедоступный мультимедийный ресурс видео-лекций, в основном исследовательского и образовательного характера. Лекции предлагаются выдающимися учеными и исследователями в рамках самых значимых и известных событий, таких как: конференции, летние школы, семинары, а также научно-популярные мероприятия в различных отраслях науки. Целями интернет-ресурса являются продвижение научных идей, стимулирование обмена знаниями, которые достигаются посредством предоставления высококачественных учебных материалов не только научной общественности, но и более широкой аудитории. Все лекции, включая документы, информацию и ссылки, систематизированы и сгруппированы редакторами с учетом комментариев пользователей.

[1] http://videolectures.net/

Задача международного соревнования VideoLectures.Net в рамках Центральной европейской конференции по анализу данных ECML/PKDD 2011 состояла в подготовке списка рекомендованных лекций ресурса VideoLectures.Net на основе исторических данных с этого сайта. Описание методов, которые использовал победитель соревнования VideoLectures.Net даны в [1]. Наш метод [2] был награжден третьим призом (Track 2).

Согласно [3], открытые социально-образовательные системы предоставляют новые возможности для миллионов заинтересованных студентов, для того чтобы последние могли пользоваться высококачественными дидактическими материалами в режиме реального времени. В соответствии с известными оценками, более 100 миллионов студентов по всему миру имеют вполне достаточный уровень образования для поступления в университет в течение следующих десяти лет. Университеты откликаются на сложившиеся потребности посредством создания открытых образовательных ресурсов: тысячи общедоступных высококачественных online-курсов, подготовленных сотнями преподавателей университетов, используются миллионами людей по всему миру. К сожалению, учебные материалы, соответствующие курсу в режиме реального времени, не дают достаточный опыт для эффективного изучения, что является необходимым условием для поддержания заинтересованности студентов.

Однако в настоящее время студенты обеспокоены качеством своего образования. С целью стимулирования и облегчения процесса обмена опытом для этих студентов необходимо решить два важных вопроса: 1) создание и накопление библиотеки учебных материалов online; 2) стимулирование обмена опытом в режиме реального времени (общение).

Центральная проблема заложена как раз во втором вопросе: каким образом задать подходящее направление для студентов-слушателей, которые «живут» в Интернете, принимая во внимание существование многообразия доступных исследовательских / образовательных ресурсов.

Функциями рекомендательных систем является профилирование пользователей по определенным критериям предпочтения и моделирование соотношений между пользователями и предметами потребления. Задача подобной системы заключается в формировании рекомендаций для того, чтобы максимально удовлетворить вкусы пользователей и облегчить последним выбор из огромного разнообразия предлагаемых услуг и товаров [4]. Рекомендательные системы имеют огромное значение в таких сферах деятельности, как: электронная торговля, подписка на базовые службы, отбор информации и др. Рекомендательные системы, формирующие персонализированные предложения, значительно повышают вероятность осуществления покупки клиентом. Индивидуальные рекомендации особенно важны на рынке, где выбор достаточно велик, вкусы потребителей играют значительно большее значение в сравнении с ценами, которые ограничены. Типичными сферами применения подобных систем являются: искусство (книги, фильмы, музыка), мода и одежда, еда и рестораны, игры и юмор.

Большинство методов, представленных в [4], были мотивированы известным соревнованием по анализу данных NetflixCup. Отметим, что методы, основанные на разложении матриц, не могут быть применены в нашем случае напрямую, поскольку данные имеют иную структуру. В нашем случае мы имеем дело не с конкретными, а с абстрактными пользователями, которые ранее ознакомились с содержанием трех лекций из предложенного набора лекций, причем их точная последовательность неизвестна.

Использование традиционных методов анализа данных (ассоциативные правила) позволило получить хорошие результаты на ранних стадиях разработки рекомендательных систем [5]. Наиболее часто используемые наборы лекций (или иных товаров/предметов потребления), определенные методом ассоциативных правил, представляют собой тип направляющих или ориентирующих образцов, поскольку они сконцентрированы на факте наличия лекций, нежели на их порядке, в котором происходит процесс рассмотрения или обучения [6]. Частотные методы (или методы, основанные на эмпирических вероятностях) являются основным инструментом в следующих 3.1 – 3.6 разделах. Заметим также, что модель Марковских цепей для принятия решений позволяет улучшить качество принятия решений для рекомендательных систем в случае, если последовательность состояний известна. Согласно теории Марковских цепей, мы имеем дело с пространством, в котором число состояний ограничено, и, используя оценку максимального правдоподобия (опирающуюся на исторические данные), мы формулируем прогноз или предсказание.

Метод бэггинг используется для вычисления множества элементарных предсказаний с целью формирования суммарного (совокупного) предиктора. Этот предиктор представляет собой усреднение относительно элементарных предикторов и дает прогноз в соответствии с большинством голосов. В разделе 3.6 мы рассмотрим метод случайных повторных выбо-

рок: предполагается, что, используя сотни предикторов (элементарных классификаторов), опирающихся на подмножества всей тренировочной выборки, мы сможем уменьшить эффект случайных факторов. Согласно принципам, на которых базируются однородные ансамбли, финальный предиктор представляет собой среднее элементарных предикторов. Отметим, что параллельно вычислению однородного ансамбля мы можем вычислить CV-паспорт [7] (cross-validation passport) для оценки качества решения. Заметим, что популярная модель случайных деревьев [8] являются хорошо известным примером удачного однородного ансамбля. Однако структура случайных деревьев основана на другом методе, который опирается прежде всего на признаки, но не на подмножества.

2. Данные и некоторые определения

Тренировочная база данных состоит из двух подмножеств: 1) пары лекций P; 2) тройки лекций T, каждая из которых включает две части (левую и правую), где левая часть содержит входные тройки лекций и соответствующие количества их просмотров, правая часть содержит выходные лекции и соответствующие количества их просмотров.

2.1. Пары лекций

Обозначим через I_p множество индексов, соответствующих парам данных. Любой элемент из I_p представляет неупорядоченный набор из двух индексов $I=\{a, b\}$, где $I \in I_p$, а индексы a и b однозначно определяют соответствующие лекции. Под выражением $P_I = P_{ab}$ мы будем понимать число тех случаев, когда лекции с индексами a и b были просмотрены вместе.

2.2. Тройки лекций

Обозначим через I_T множество индексов, соответствующих тройкам данных: $\tau_I = \{a, b, c\}$ – это тройка или набор из трех лекций a, b и c. Элемент с индексом $I \in T$ имеет два значения (одно для левой и одно для правой частей). Под выражением T_I мы будем понимать число случаев, когда соответствующие три лекции были просмотрены вместе; L_I – набор отдельных лекций, просмотренных после τ_I. А также обозначим через $T_I(\ell), \ell \in L_I$, количество случаев когда лекция была просмотрена после тройки τ_I.

2.3. Графические иллюстрации

Рис. 1 иллюстрирует гистограмму эмпирических вероятностей или частот:

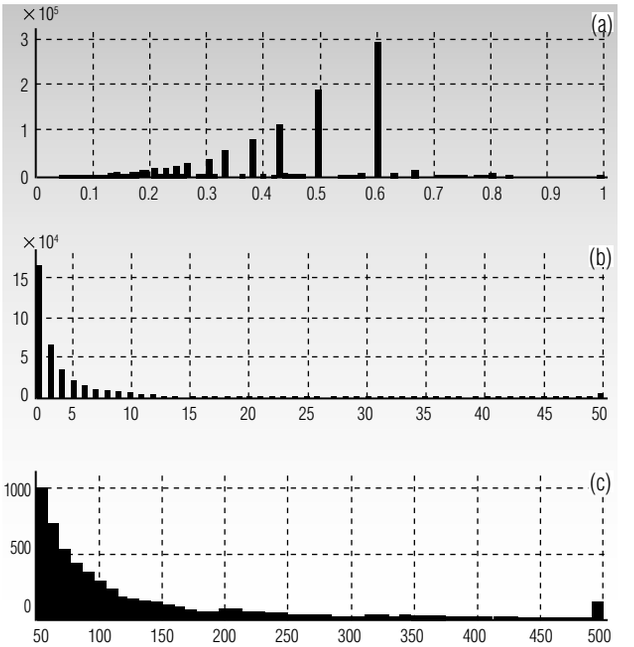

Рис. 1. Гистограммы частот, соответствующих (a) тройкам; (b) парам до 50; (c) парам от 50 до 500, см. раздел 2.3

$$r_\tau(\ell) = \frac{T_I(\ell)}{T_I},\ \ell \in L_I, I \in I_T, \qquad (1)$$

где мы заменили I на τ в левой части (1), поскольку имеет место взаимно однозначное соответствие между параметрами I и τ.

Рис. 1(*b-c*) иллюстрирует гистограмму частот: P_I, $I \in I_p$, где все значения на *рис. 1*(*b*) сокращены до уровня 50 (если значения превышают 50). Дополнительно *рис. 1*(*c*) демонстрирует гистограмму значений P_I от 50, где значения, превышающие 500, сокращены до 500.

3. Методы

3.1. Прогнозы при использовании двоек лекций

Задача соревнования заключалась в построении прогноза (предсказания) согласно тестовой базе данных V, которая имеет такую же структуру, как и T, (левая часть). Точнее говоря, было необходимо предложить рекомендацию десяти наиболее подходящих лекций после просмотра данных трех.

Замечание 1. В качестве особенного и наиболее важного фактора данного соревнования отметим отсутствие одинаковых троек в тренировочном T (левая часть) и тестовом V множествах. В то же время мы обнаружили значительное количество одинаковых пар в обоих множествах: тренировочном и тестовом.

В целом мы нашли $n_c = 34756$ двоек (которые были найдены в левой части T) с количеством повторов, соответствующих отдельной двойке, от 1 до $m_c = 4020$.

Отметим, что каждая тройка может быть рассмотрена как набор из трех двоек. Общее количество троек в тестовой выборке V составило $N_V = 60274$, и 1) мы не обнаружили соответствующих им двоек в тренировочной выборке только в 116 случаях, 2) мы нашли только одну двойку в 829 случаях, 3) мы нашли две двойки в 4705 случаях; 4) все три двойки были обнаружены в абсолютном большинстве (54624 случаев).

Замечание 2. В оригинальной базе данных каждая лекция была идентифицирована при помощи индекса, где наибольший индекс $n_L = 13251$. Однако не все n_L лекции были использованы. Мы предположили, что предсказанные лекции следует искать в правой части T, где определены только $n_s = 5209$ различных лекций.

3.2. Некоторые предварительные определения и обозначения

Мы объясним, как работает система в условиях n_s вторичных индексов, так как преобразование к первоначальным n_L индексам является тривиальной задачей. Наша база данных была организована следующим образом: квадратная матрица A размерности $n_s \times n_s$ содержит n_c различных адресов матрицы B размерности $n_c \times m_c$.

Сначала мы найдем три пары $\alpha_{ij}, j = 1, 2, 3$, для всех троек τ_i, $i = 1, ..., N_V$, содержащихся в тестовой базе данных V. Затем для каждой пары α_{ij} мы найдем (в соответствии с матрицей A) соответствующий адрес $\beta(\alpha_{ij})$(номер строки в матрице B) и количество имеющихся записей $n(\alpha_{ij})$, где $1 \le \beta \le n_c$, $1 \le n \le m_c$.

Под элементом матрицы B мы будем понимать предсказанную/рекомендованную лекцию ℓ и соответствующую частоту:

$$\{\ell, r_\tau(\ell)\}, \quad (2)$$

где отношение r_τ определено в (1).

Процесс обновления

Здесь мы опишем наиболее важный шаг вычислительного процесса. Каждая конкретная тройка τ тренировочной базы данных рассматривается идентичным образом, поэтому мы опускаем индекс i с целью упрощения обозначений.

Предположим, что первоначально все рейтинги равны нулю $s(\ell)=0$, $\ell = 1, ..., n_s$, где $s(\ell)$ – это рейтинги, соответствующие лекции ℓ, которые будут использованы для конечного ранжирования лекций в качестве результата модели. Далее представлена наиболее важная формула для обновления:

$$s(B_{\beta k}(1)) = s(B_{\beta k}(1)) + B_{\beta k}(2), k = 1, ..., n(\alpha_j), j = 1, 2, 3, \quad (3)$$

где $B_{\beta k}(1)$ – индекс лекции, $B_{\beta k}(2)$ – соответствующая частота, которая была определена в (2).

После вычисления вектора s в соответствии с (3) мы отсортируем соответствующие элементы в порядке убывания, и индексы лекций, соответствующие десяти наибольшим значениям s, могут быть представлены в качестве решения.

Замечание 3. В случае если число положительных значений вектора s меньше, чем 10, мы генерировали оставшиеся индексы случайным образом, предполагая, что они отличны от 1) индексов входящей тройки τ_i, а также отличны от 2) тех индексов, что уже выбраны.

При помощи метода, описанного в этом разделе, был получен результат 0,49568 в терминах критерия качества, который использовался организаторами соревнования[2] для сравнения различных решений:

$$MAP_p = \frac{1}{|V|}\sum_{\ell \in V} AvgRp(\ell), \text{ где}$$

$$AvgRp(\ell) = \frac{1}{|Z|}\sum_{z \in Z} Rp@z(\ell),$$

$$Rp@z(\ell) = \frac{|relevant \cap retrieved|_z}{|relevant|_z},$$

где $Z = \{5, 10\}$ – первые 5 или все 10 лекций (имеется в виду, что рекомендованные 10 лекций отсортированы в порядке убывания в соответствии с рейтингом).

Замечание 4. Главное преимущество описанного выше метода, который основан на информационной базе данных, включающей матрицы A и B, состоит в его скорости. Согласно проведённым экспериментам, изложенный в этом разделе алгоритм прошёл через всю тренировочную выборку V и вычислил требуемое решение в течение пятидесяти одной секунды. Использовались 1) многопроцессорная рабочая станция с операционной системой Linux 3.2 GHz 16 GB RAM и 2) специально разработанная программа написанная на языке программирования C (все временные замеры осуществлялись в автоматическом режиме).

3.3. Прогнозы при использовании отдельных лекций

Отметим, что прогнозы при использовании отдельно взятых лекций работают схожим образом, что и прогнозы с парными предсказаниями. Однако имеются некоторые различия, которые могут быть рассмотрены как упрощения. Мы выяснили, что максимальное число записей, соответствующих отдельной лекции, – $m_s = 77798$. Соответственно

[2] http://tunedit.org/challenge/VLNetChallenge

матрица $\hat{B}$ (замена матрицы B в предыдущем разделе 3.1) имеет размерность $n_s \times m_s$.

Модель работает следующим образом: по определению, любая тройка представляет собой множество (набор из трех лекций), состоящее из трех лекций ℓ_1, ℓ_2, ℓ_3. Мы найдем количество записей для каждой отдельной лекции $1 \le n(\ell) \le m_s$, где $1 \le \ell \le n_s$.

Процесс обновления

Как и ранее, первоначально все рейтинги равны нулю: $s(\ell) = 0$, $\ell = 1, ..., n_s$. Ниже представлена основная формула для обновления:

$$s\left(\hat{B}_{\ell_j k}(1)\right) = s\left(\hat{B}_{\ell_j k}(1)\right) + \hat{B}_{\ell_j k}(2),\ k = 1,...,n(\ell_j),\ j = 1,2,3. \quad (4)$$

После вычисления вектора s в соответствии с (4) сортируем значения в порядке убывания, и индексы лекций, соответствующие десяти наибольшим значениям s, представимы в качестве решения.

Метод, который описан в этом разделе, позволил нам получить результат 0,33278.

3.4. Прогнозирование при использовании пар лекций

Определение 1. Будем называть лекции a и b *P-связанными*, если $P_{ab} \ge 1$. В соответствии с симметричной матрицей P определим множество лекций $H(a)$, которые P-связаны с данной лекцией a.

Процесс обновления

Как и ранее, первоначально все рейтинги равны нулю: $s(\ell) = 0$, $\ell = 1, ..., n_s$.

Далее мы представим формулу для обновления:

$$s(d) = s(d) + P(\ell_j, d), d \in H(\ell_j),\ j = 1,2,3, \quad (5)$$

где определение лекций ℓ_j является таким же, что и в (4).

После вычисления вектора s в соответствии с (5) мы сортируем значения в порядке убывания, и индексы лекций, соответствующие десяти наибольшим значениям s, представимы в качестве решения.

При использовании метода, представленного в этом разделе, был получен результат 0,12677.

Замечание 5. Решение, описанное в данном разделе, было рекомендовано организаторами форума как «simple pairs solution».

Заметим, что в ходе наших числовых экспериментов мы сделали довольно интересное наблюдение.

Замечание 6. Оценки, определенные в (3-5), представляют суммы частот. Очень интересно отметить, что результаты будут значительно слабее, если мы используем среднее в качестве альтернативы сумме.

3.5. Прогнозы при использовании двоек с весовыми коэффициентами

В соответствии с предыдущими тремя разделами, прогнозы при использовании двоек позволили нам получить лучшие результаты. Мы решили продолжить работу в этом направлении и принять во внимание оставшиеся третьи лекции φ и ψ в обеих тренировочной и тестовой выборках.

Основная идея метода: в случае если оставшиеся лекции φ и ψ подобны (имеют большое количество общих просмотров в соответствии с парными данными), направление прогноза, соответствующее двойке, приобретает больший вес.

Как было отмечено в замечании 1, лекции φ и ψ различны по определению. Иными словами, схожие тройки из тренировочной и тестовой баз данных могут быть представлены в следующем виде: $\alpha_j \cup \varphi_j, \alpha_j \cup \psi_j$, где $\varphi_j \ne \psi_j$, $j = 1,2,3$.

Процесс обновления

Как и ранее, первоначально все рейтинги равны нулю: $s(\ell) = 0$, $\ell = 1, ..., n_s$. Затем мы можем переписать (3) следующим образом:

$$s\left(B_{\beta k}(1)\right) = s\left(B_{\beta k}(1)\right) + w\left(P(\varphi_j, \psi_j)\right) B_{\beta k}(2), \quad (6)$$
$$k = 1,...,n(a_j),\ j = 1,2,3,$$

где w – возрастающая функция весовых коэффициентов. При вычислении нашего финального решения мы использовали простейшую линейную функцию: $w(x) = 0.01 \cdot x + 0.05$.

После вычисления вектора s в соответствии с (6) мы сортируем полученные значения в порядке убывания, и индексы лекций, соответствующие десяти наибольшим значениям s, представимы в качестве решения.

При использовании метода с взвешенными двойками, который был описан нами в этом разделе, было достигнуто значительное улучшение: 0,58145.

3.6. Прогнозы при использовании отдельных лекций с весовыми коэффициентами

Данный раздел может быть рассмотрен как дополнение к разделу 3.2. В некотором смысле прогнозы с взвешенными отдельно взятыми лекциями схожи с прогнозами с взвешенными двойками (раздел 3.4). Однако есть некоторые отличия. В случае с отдельными лекциями мы определяем направление прогноза согласно одиночным лекциям. Соответственно мы имеем две другие (дополнительные) лекции, которые необходимо сравнить с двумя лек-

циями из соответствующей тройки лекций из тренировочного множества.

Процесс обновления осуществляется следующим образом:

$$s\left(B_{\ell_j k}(1)\right) = s\left(B_{\ell_j k}(1)\right) + w(\varphi_{1j},\varphi_{2j};\psi_{1j},\psi_{2j})\cdot B_{\ell_j k}(2),$$
$$k = 1,\ldots,n(\ell_j),\ j = 1,2,3,\ \text{где} \qquad (7)$$
$$w(\varphi_{1j},\varphi_{2j};\psi_{1j},\psi_{2j}) =$$
$$= 0,0005\cdot\begin{pmatrix} P(\varphi_{1j};\psi_{1j})P(\varphi_{2j};\psi_{2j}) + \\ + P(\varphi_{1j};\psi_{2j})P(\varphi_{2j};\psi_{1j}) \end{pmatrix} + 0,01.$$

Идея данной формулы проста: мы должны быть уверены, что каждая дополнительная лекция из тестовой тройки лекций близка по крайней мере к одной дополнительной лекции из тренировочной тройки.

После вычисления вектора *s* согласно (7) мы сортируем полученные результаты в порядке убывания, и индексы лекций, соответствующие десяти наибольшим значениям *s*, представимы в качестве решения.

При использовании данного метода был получен результат 0,4529.

3.7. Метод случайных повторных выборок (финальная рекомендательная система)

Вычисление отдельного упорядоченного вектора *s* в данном разделе основано на 75% случайно выбранных выборках. В абсолютном большинстве всех 60274 тестовых образцов число положительных компонентов вектора *s*, определенных в (6), больше 100. Поэтому мы будем рассматривать только этот случай.

Таблица 1.

Различия между пятью решениями (в терминах дистанции (8)), представленными в разделах 3.1 - 3.7

N	Метод/ Раздел	Результат	1	2	3	4	5
1	3.1	0,49568	0	0,2605	0,2137	0,6394	0,6517
2	3.3	0,33278	0,2605	0	0,5832	0,4269	0.4327
3	3.4	0,12677	0,2137	0,5832	0	0.1565	0.1664
4	3.5	0,58145	0,6394	0.4269	0.1565	0	0.91
5	3.7	0,58727	0,6517	0.4327	0.1664	0.91	0

Обозначим вектор вторичных рейтингов как *z*, представляющий собой множество нулей в начале всего процесса повторных выборок. Мы исследовали 200 случайных выборок (глобальные итерации). После каждой глобальной итерации только 100 лекций (компоненты вектора *z*) получили приращение в диапазоне от 1 до 100 голосов (чем больше, тем лучше). Метод, который мы использовали внутри каждой глобальной итерации (элементарный классификатор), описан в разделе 3.4.

После завершения всех 200 глобальных итераций мы отсортировали вектор *z* в порядке убывания, и индексы лекций, соответствующие десяти наибольшим значениям *z*, представим в качестве решения.

Таким образом, при использовании метода повторных выборок, представленного в этом разделе, был получен результат **0,58727**. Этот результат был использован в качестве итогового результата.

3.8. Статистическое сравнение различных решений

Отметим, что любое решение представляет собой целочисленную матрицу $N_V \times 10$, $T = 10N_V$ целочисленных индексов в целом. Путем сравнения двух матриц мы найдем число общих индексов (пересечение) в каждом ряду. Общее число пересечений даст нам значение числителя *R*, и требуемое расстояние будет представлено в виде отношения:

$$D = \frac{R}{T} \qquad (8)$$

3.9. Время вычисления

Для вычислений была использована многопроцессорная рабочая станция с операционной системой Linux 3.2 GHz 16 GB RAM. Вычисления были произведены на основе специально разработанного кода (алгоритма) в C. Для получения финального решения, описанного в разделе 3.6, потребовалось около 12 часов.

4. Заключительные замечания

Мы согласны с утверждением [9], что превосходство новых алгоритмов следует демонстрировать на независимых данных. В этом смысле важность соревнований по анализу данных является неоспоримой. Стремительно растущая популярность подобных соревнований свидетельствует о том, что они является одним из самых эффективных способов оценки различных моделей и систем.

В целом, мы удовлетворены нашими результатами, продемонстрированными в ходе соревнования PKDD2011. В качестве одного из направлений для дальнейшего развития было бы интересно найти эффективный способ построения неоднородных ансамблей. Например, при использовании методов отдельных лекций со взвешенными коэффициентами (см. раздел 3.6) и двоек лекций (см. раздел 3.1) результаты имеют значительные расхождения. Тем

не менее, оба результата являются достаточно хорошими и заслуживают внимание с тем, чтобы найти, каким образом интерпретировать и использовать различия между исходными/базовыми решениями для построения решения более высокого уровня.

Кроме того, мы полагаем, что метод градиентной факторизации [10-11] вполне применим в этой задаче и может привести к принципиально новому высококачественному решению.

Отметим, что предложенный метод эмпирических вероятностей был мотивирован структурой данных Международного соревнования PKDD 2011 и имеет тесную связь с популярным методом ассоциативных правил, который нашел широкое применение в ряде областей, см. [12 - 14]. В качестве альтернативного примера и иллюстрации применения метода эмпирических вероятностей мы можем рассмотреть задачу определения влияния принятия лекарств на последующие состояния пациентов [15]. Путем сравнения реальных событий (интенсивность которых измеряется путем метода эмпирических вероятностей) и аналитически ожидаемых событий мы можем выявить интересные закономерности и явления. Этой задаче было посвящено Международное соревнование по анализу данных, организованное американской компанией OMOP (Observational Medical Outcomes Partnership). Кубок OMOP 2010[3] включал две секции, где описание метода победителя в первой секции опубликовано в статье [16]. Наш метод был официально признан лучшим в секции №2.

Авторы благодарны анонимному рецензенту за ряд полезных замечаний. ■

Литература

1. Дьяконов А.Г. Алгоритмы для рекомендательной системы: технология LENKOR // Бизнес-Информатика. – 2012. – Т. 1, № 19. – С. 32-39.
2. Nikulin V. OpenStudy: recommendations of the following ten lectures after viewing a set of three given lectures // ECML/PKDD workshop and conference proceedings, discovery challenge. Editors: Tomislav Smuc, Nino Antulov-Fantulin, Mikołaj Morzy. – Athens, Greece, 2011. – C. 59-70.
3. Ram A., Ai H., Ram P., Sahay S. Open social learning communities // In International conference on web intelligence, mining and semantics. – Sogndal, Norway, 2011.
4. Takacs G., Pilaszy I., Nemeth B., Tikk D. Scalable collaborative filtering approaches for large recommender systems // Journal of machine learning research. – 2009. - No. 10. – P. 623-656.
5. Agrawal R., Srikant R. Fast algorithms for mining association rules // Proceedings of 20th Int. conf. very large data bases. 1994. – 32 p.
6. Mobasher B., Dai H., Luo T., Nakagawa M. Using sequential and non-sequential patterns in predictive web usage mining tasks // ICDM. – 2002.
7. Ефимов Д.А., Никулин В.Н. Предсказание биологического состояния молекул исходя из их химических свойств // Advanced Science, Вятский Государственный Университет. – 2013. – Т. 2, № 2. – С. 107-123.
8. Breiman L. Random Forests // Machine Learning. – 2001. – Vol. 45, No.1. – P. 5-32.
9. Jelizarow M., Guillemot V., Tenenhaus A., Strimmer K., Boulesteix A.-L. Over-optimism in bioinformatics: an illustration // Bioinformatics. – 2010. – Т. 26, № 16. С. 1990-1998.
10. Nikulin V., Huang T.-H., Ng S.-K., Rathnayake S., McLachlan G. A very fast algorithm for matrix factorization // Statistics and probability letters. – 2011. – Vol. 81. – P. 773-782.
11. Rendle S. Factorization machines with libFM // ACM Transactions on Intelligent Systems and Technology (TIST). – 2012. – Vol. 3, No. 3. – P. 22.
12. Papender K., Deepak D., Nidhi P. Diagnosis of tuberculosis using association rule method // Journal of Information and Operations Management. – 2012. – Vol. 3, No. 1. – P. 133-135.
13. Martinez-Ballesteros M., Troncoso A., Martinez Alvarez F., Riquelme J. Mining quantitative association rules based on evolutionary computation and its application to atmospheric pollution // Integrated Computer-Aided Engineering. – 2010. – Vol. 17. – P. 227-242.
14. Gautam P., Pardasani K. Efficient method for multiple-level association rules in large databases // Journal of Emerging Trends in Computing and Information Sciences. – 2011. – Vol. 2, No. 12. – P. 722-732.
15. Norén G., Hopstadius J., Bate A., Star K., Edwards I. Temporal pattern discovery in longitudinal electronic patient records // Data Mining and Knowledge Discovery. – 2009.
16. Schuemie M. Methods for drug safety signal detection in longitudinal observational databases: LGPS and LEOPARD // Pharmacoepidmiology and Drug Safety. – 2010.

[3] http://omop.fnih.org/omopcup