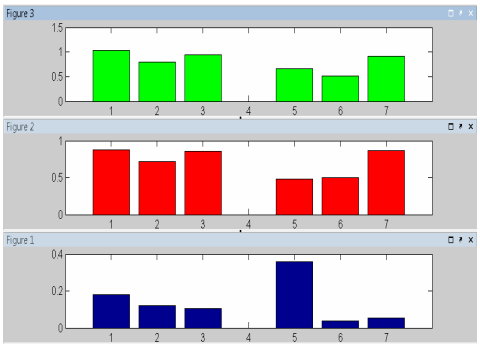
ВВЕДЕНИЕ В АНАЛИЗ ДАННЫХ

научно-популярная лекция,
подготовленная для просеминара
кафедры математических методов прогнозирования (12.03.2012)

доцент ВМК МГУ имени М.В. Ломоносова,
д.ф.-м.н. А.Г. Дьяконов



ВНИМАНИЕ! ЛЕКЦИЯ НАХОДИТСЯ В СТАДИИ ПРАВКИ

ОБО ВСЕХ НЕТОЧНОСТЯХ СООБЩАЙТЕ АВТОРУ (djakonov(a)mail(dot)ru)

Вступление

Прежде всего, давайте обсудим терминологию. Речь идёт об области, которая в западной литературе называется **Data Mining**, а на русский язык чаще переводится как «анализ данных[1]». Термин не совсем удачный, поскольку слово «анализ» в математике достаточно привычно, имеет устоявшееся значение и входит в название многих классических разделов: математический анализ, функциональный анализ, выпуклый анализ, нестандартный анализ, многомерный комплексный анализ, дискретный анализ, стохастический анализ, квантовый анализ[2] и т.д. Во всех перечисленных областях науки изучается математический аппарат, который базируется на некоторых фундаментальных результатах и позволяет решать задачи из этих областей. В анализе данных ситуация гораздо сложнее. Это, прежде всего, прикладная наука, в которой математического аппарата нет, в том смысле, что **нет конечного набора базовых фактов, из которых следует, как решать задачи**[3]. Многие задачи «индивидуальны», и сейчас появляются всё новые и новые классы задач, под которые необходимо разрабатывать математический аппарат. Тут ещё большую роль играет тот факт, что анализ данных относительно новое направление в науке.

Далее, надо пояснить, что такое «анализ данных». Я назвал это «областью», но областью чего? Здесь начинается самое интересное, поскольку это **не только область науки**. Настоящий аналитик решает, прежде всего, прикладные задачи и нацелен на практику. Кроме того, анализировать данные приходится в экономике, биологии, социологии, психологии и т.д. Решение новых задач, как я уже сказал, требует изобретения новых техник (это не всегда теории, но и приёмы, способы и т.п.), поэтому некоторые говорят, что анализ данных это также искусство и ремесло. И, как мы увидим дальше, анализ данных теперь становится даже спортом[4], поскольку многие компании выкладывают некоторые данные в открытом доступе, чтобы исследователи могли написать алгоритмы по их обработке, анализу и т.д. Лучшие алгоритмы покупаются, а процедура определения лучшего превращается в соревнование, со всеми спортивными изюминками: регламентом, подготовкой, тактикой, нечестными приёмами и т.д.

---

[1] Иногда даже говорят «интеллектуальный анализ данных», что, на мой взгляд, совсем «ужасно», поскольку не существует «неинтеллектуального» анализа данных.

[2] Полезно напомнить происхождение слова «анализ». Оно было использовано Лопиталем в названии учебника «*Анализ бесконечно малых*» *М.-Л.:ГТТИ, 1935*.

[3] Есть также мнение, что такого математического аппарата никогда и не будет. Впрочем, это не тема нашей лекции…

[4] Интересен лозунг на сайте компании [KAGGLE]: «Мы делаем анализ данных спортом». Компания как раз является посредником между бизнесом и специалистами по анализу данных.

Из того, что мы рассматриваем прикладную науку, следует и ещё одно важное замечание, которое существенно повлияет на изложение наших лекций. В прикладных областях **самое важное – это практика**! Невозможно представить себе хирурга, который не сделал ни одной операции. Собственно, это и не хирург вовсе. Также не может аналитик данных обойтись без решения реальных прикладных задач. Чем больше таких задач вы самостоятельно решите, тем более квалифицированными специалистами вы станете. И в этой лекции мы, прежде всего, **будем говорить о реальных прикладных задачах**. Возможно, во вводной лекции следовало бы рассказать, на какие подобласти разбивается анализ данных, дать постановки задач в общем виде и т.д. и т.п. Но мы отойдём от этой традиции и **на примерах покажем, с какими задачами приходится сталкиваться и как их удаётся решать**. Да, в этой лекции мы разберём, в том числе, решения задач, поскольку они не такие уж и сложные, это придаст вам уверенности, что анализом данных можно начать заниматься уже сейчас! Задач будет не очень много (поскольку у нас есть временные рамки), зато мы их более-менее подробно рассмотрим.

Кстати, такой взгляд на анализ данных определяет и тематику данной лекции (и всех последующих). Я считаю, что здесь преподаватель должен учить тому, в чём он действительно является высококлассным специалистом. Поэтому и о задачах я расскажу только тех, с которыми сам сталкивался и успешно решал, а не о тех, которые знаю по литературе и выступлениям своих коллег на конференциях.

Когда клиент заплатит деньги?

прогнозирование визитов покупателей супермаркетов

Рассмотрим первую задачу анализа данных. В 2011 году компания [dunnhumby] предложила исследователям такую задачу. Есть статистика посещения клиентами магазинов сети супермаркетов: кто[5] и когда приходил, сколько заплатил. Необходимо для каждого клиента предсказать, когда он в следующий раз посетит магазин и сколько при этом заплатит. Отметим, что алгоритмы оценивались очень строго: ответ считался правильным, если точно предсказана дата **первого** визита и в сумме покупки ошибка составляет не более 10$.

Замечание. В рассматриваемой задаче сложно сказать, какой экономический эффект ожидали представители компании от наличия алгоритма прогнозирования первых визитов и сумм покупок. Обычно супермаркеты заинтересованы в алгоритмах прогнозирования спроса: что в ближайшее время будут покупать. Это необходимо для закупки товаров и выставления их на витрины: закупается только нужное, выставляется только то, что раскупят до того, как товар пропадёт.

Хотя часто компании устраивают подобные соревнования для исследователей с целью

[5] Естественно, данные обезличены. Каждый клиент определяется идентификационным номером (например, номером скидочной карты).

не решить конкретную задачу, а найти высококлассных специалистов. Поэтому задача заведомо выбирается нестандартной (не освящённой в литературе). И именно поэтому **необходимо такие задачи уметь решать**.

На рис. 1 показана статистика покупок одного из клиентов. Она известна до 31 марта (30 марта он заплатил 60$, 28-го – 35$, 24-го – 5$ и т.д.). Для наглядности разобьём этот ряд на недели, см. рис.2–3.

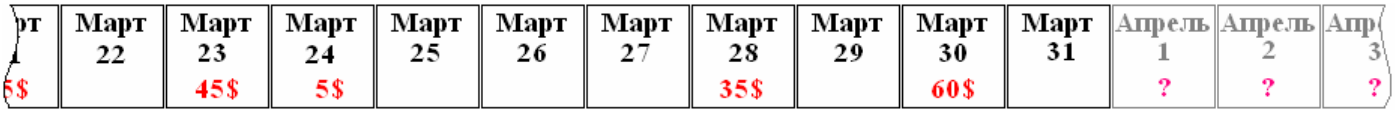


Рис. 1. Статистика покупок одного клиента.

Рис. 2. Разбиение на недели.

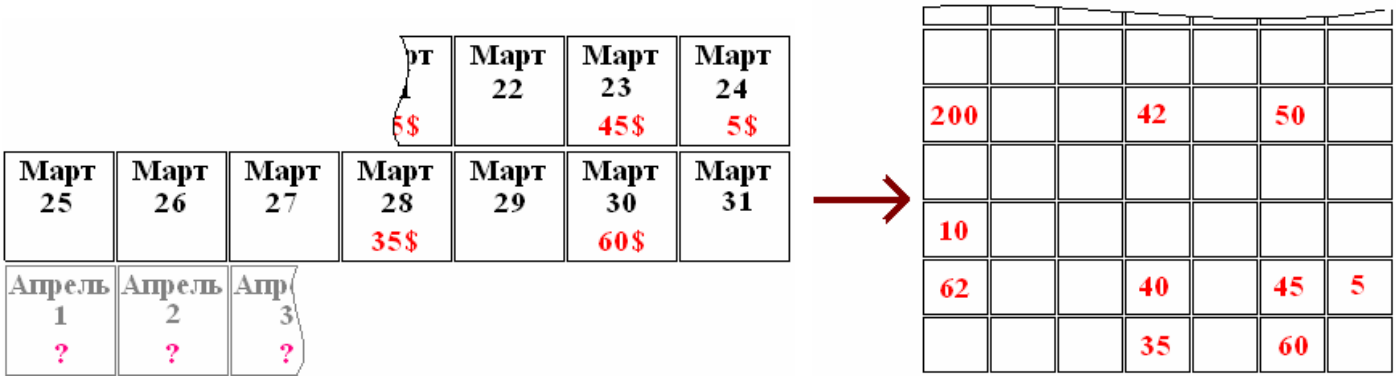


Рис. 3. Матрица недель.

На самом деле, такое разбиение делается не только для наглядности. Логично предположить, что у клиентов есть дни, в которые они чаще посещают магазин. Теперь по матрице недель (см. рис. 3–4) такие дни ясно видны. На рис. 4 удалены недели (точнее соответствующие строки матрицы), в которые данный клиент не ходил в магазин.

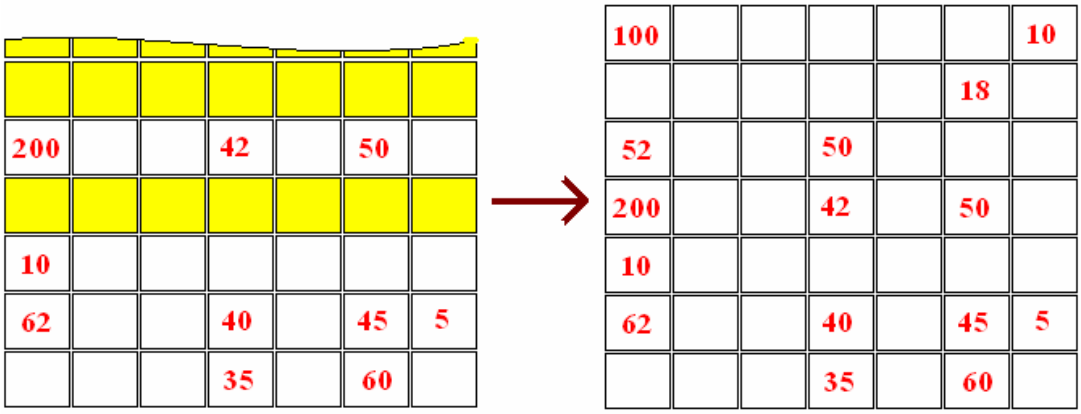


Рис. 4. Удаление недель без визитов.

Рассмотрим следующую математическую модель. Пусть клиент в первый день недели с вероятностью p_1 идёт в магазин (можно считать, что он подбрасывает монетку, которая с вероятностью p_1 падает орлом вверх, и в этом случае он идёт в магазин). Во второй день недели он посещает магазин с вероятностью p_2 и т.д., в седьмой – с вероятностью p_7. Из курса теории вероятностей следует, что вероятность того, что первый визит будет

в первый день недели, равна $\tilde{p}_1 = p_1$,

во второй день недели – $\tilde{p}_2 = (1-p_1)p_2$,

…

в седьмой день недели – $\tilde{p}_7 = \prod_{i=1}^{6}(1-p_i)p_7$.

Очевидно, что «ставить» надо на тот день, которому соответствует наибольшая вероятность. А как же оценить вероятности p_1, p_2, …, p_7 (по которым мы вычислим искомые $\tilde{p}_1,\ldots,\tilde{p}_7$)? По нашей матрице недель! Вероятность p_i можно оценить как число визитов в i-й день, делённое на число недель[6].

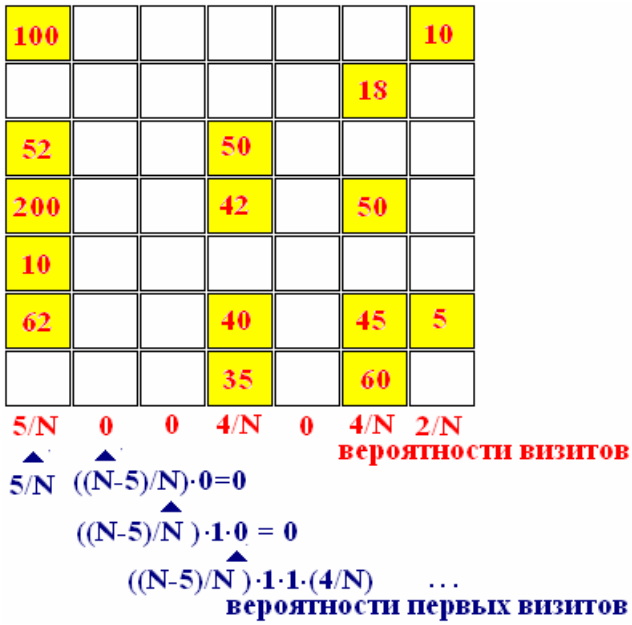


Рис. 5. Простейшее вычисление вероятностей визитов.

Замечание. Мы говорим об **оценке** вероятности, а не её вычислении, поскольку чтобы узнать её истинное значение необходимо иметь бесконечную выборку (в нашем случае – статистику поведения клиента). Да и существует ли она?! Ведь вероятности взялись из нашей модели, которая предполагает, что поведение клиента случайно, более того, может моделироваться ежедневным подбрасыванием монетки. Модель очень уж упрощает реальную ситуацию, но в данной задаче она «сработала».

Кстати, вероятности первых визитов можно было оценивать и по-другому: для *i*-го дня недели делить число недель, в которых первый визит был в *i*-й день, на число всех недель. На практике часто придумывают много способов оценки одной вероятности и

[6] Хотя делить надо на число всех недель, на практике лучше сработало деления на число недель, в которые были сделаны визиты.

пытаются их комбинировать. Здесь можно складывать полученные вероятности с неотрицательными коэффициентами, в сумме дающими 1. Сами коэффициенты определяются в результате решения задачи оптимизации качества алгоритма.

Между прочим, это очень важно! В теории можно получить какой-то оптимальный способ оценки вероятности, а на практике, на реальных данных, **часто лучше всего работает комбинация нескольких**, пусть даже и не оптимальных.

Замечание. Интересно, что многие при решении рассматриваемой задачи искали максимум среди вероятностей визитов, а не **первых** визитов, что, конечно же, неверно.

Пока в нашей модели никак не учитывается время (можно переставить строки в матрице недель, а вероятности не изменятся). Естественно, **«более свежие» данные о клиенте более важны, чем устаревшие.** Скажем, информация, что последний месяц он каждый понедельник ходил в магазин, более важна, чем информация о том, что он ходил каждый понедельник в магазин в прошлом году. Это легко учесть в модели, введя веса важности недель: более поздние имеют больший вес.

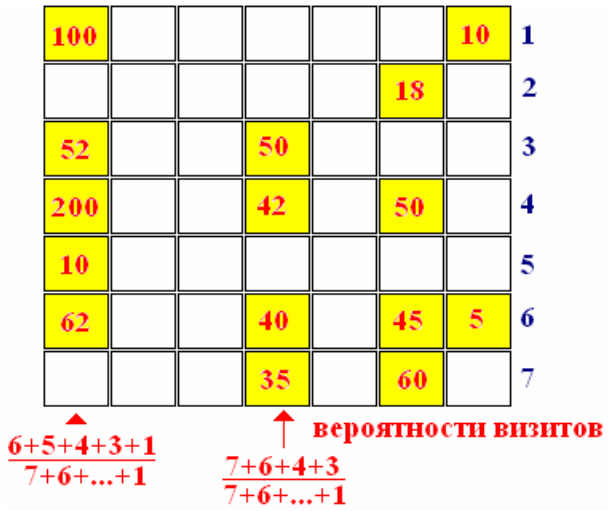


Рис. 6. Вычисление с весами недель

Замечание. На рис. 6 показана линейная схема весов: веса убывают линейно при «удалении от последней недели». На практике надо перебрать различные весовые схемы. В этой задачи лучшей была квадратичная: с весами 1, 4, 9 и т.д.

Теперь поговорим о том, как прогнозировать сумму покупки. В принципе, это задача прогнозирования временного ряда, методы её решения изучаются в курсе **эконометрики**. Но в нашем случае всё оказалось просто: лучше всех работал достаточно простой метод, который мы и рассмотрим. Предположим, что в день недели, в который наш клиент ожидается, раньше он платил ровно 50$. На какую сумму следует «делать ставку»? Правильно – 50$! А если он совершал покупки на суммы 50$ и 70$? В этом случае мы должны поставить на 60$! Ведь ответ нашего алгоритма считается правильным, если он ошибается не больше, чем на 10$. Поэтому ответ 60$ является, в некотором смысле, «компромиссным», он устраивает нас, если клиент будет вести себя как раньше: платить по 50$ и 70$. В общем случае, когда нам известны суммы покупок в

день недели, соответствующий дню прогноза, надо просто построить функцию на вещественной оси, равную сумме «ступенек». Каждая ступенька имеет центром соответствующую сумму покупки, а ширину – 20$. Точка, соответствующая аргументу, который максимизирует функцию, будет нашим ответом, см. рис. 7 – 8. Такое решение будет оптимальным, если клиент будет вести себя «как раньше».

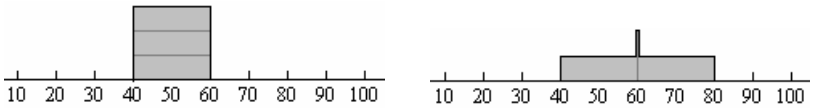


Рис. 7. «Суммы ступенек» при покупках 50, 50, 50 (слева) и 50, 70 (справа)

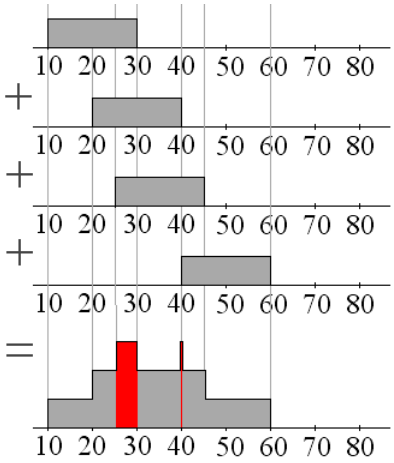


Рис. 8. «Суммы ступенек» при покупках 20, 30, 35, 50 – максимум достигается на отрезке [25, 30].

Замечание. На самом деле, мы только что описали идею непараметрического метода восстановления плотности. Подробнее о нём можно прочитать в [Дуда, Харт, 1976]. Различные вариации этого метода часто бывают полезными в задачах анализа данных.

Опять предложенный метод не учитывает временной фактор, и опять мы можем исправить это введением весов в нашу модель. Ступеньки, которые соответствуют поздним покупкам, должны быть выше, чем ступеньки, соответствующие ранним покупкам. Надо ещё отметить, что на практике при прогнозировании покупок учитываются не только покупки этого дня недели, но и последние покупки, покупки, сделанные ровно год назад, сделанные в похожие дни и т.д.

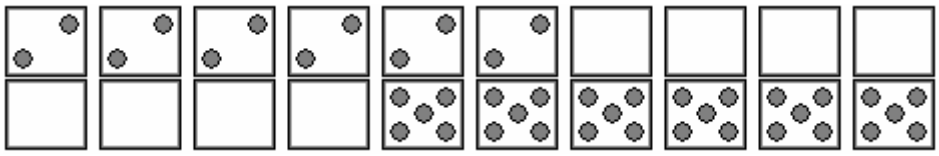


Рис. 9. Пример с костями домино.

В принципе, мы описали методы прогнозирования дня и суммы покупки. Можно ли использовать его для решения нашей задачи? В ней как раз требуется предсказать день и сумму! Оказывается, **«И» в условии не означает «И» в решении**, т.е. нельзя просто сначала предсказать день, а потом сумму. Чтобы пояснить это, рассмотрим простую задачу. Допустим, я с возвратом 10 раз выбрал кость домино из коробки

(не обязательно случайно). Теперь я вам говорю, что в первой позиции выбранных домино чаще всего встречалась двойка, а во второй – пятёрка. Ответ, что кость «2-5» была выбрана чаще остальных, неверный. На рис. 9 показан случай, когда она даже не на втором месте по частоте выбора, а кости «2-0» и «0-5» выбирались в два раза чаще!

Теперь пример для нашей задачи. Допустим, в понедельник первый визит клиента будет с вероятностью 0.9, а во вторник – с вероятностью 0.1[7]. По нашему методу мы ставим на понедельник. Но в понедельник его траты были:

10$, 50$, 220$, 100$, 310$, 5$, 250$, 75$, 500$ и т.д.,

а во вторник:

40$, 42$, 40$.

Видно, что во вторник, хоть он и ходит в магазин редко, но его поведение стабильно, можно сказать, что оно угадывается с вероятностью 1. В понедельник же его поведение нестабильно: суммы покупок сильно разбросаны. Допустим, мы как-то оценили вероятность угадывать суммы покупок по понедельникам как 0.1. Тогда, если мы ставим на понедельник, то вероятность успешности нашего прогноза равна произведению вероятности визита и вероятности угадывания:

$$0.9*0.1 = 0.09,$$

а если поставим на вторник –

$$0.1*1 = 0.1.$$

Поэтому в**ыгоднее ставить на вторник, а не на понедельник**!

Основной вопрос, который теперь возникает – как оценить вероятность угадывания суммы покупки в каждый день? На практике для такой оценки хорошо подходит высота того графика, который мы строили на рис. 7–8[8]. Если пересечение оснований всех ступенек непусто, то вероятность равна единице, в «противоположном» случае, когда они все попарно не пересекаются, вероятность равна

$$1 / (\text{число ступенек})$$

и близка к нулю.

Вопрос. Не являются ли приведённые примеры искусственными? Особенно пример с костями домино. Поскольку раньше мы говорили о случайном поведении пользователя, а выбор костей домино на рис. 9 явно не является случайным.

Ответ. Можно сказать «и да, и нет». Конечно, пример с домино искусственный, но в нашей задаче подобный эффект был. Этим и отличается анализ данных от «строгих наук». Можно придумать много разных моделей: простых и сложных, естественных и искусственных, но насколько они хороши, решает эксперимент! Надо только уметь придумывать.

В нашей задаче, правда, мы можем этот эффект объяснить. Допустим, человек каждую субботу закупает товары в магазине. Это продукты, иногда ещё что-то по хозяйству, иногда даже какая-то бытовая техника и т.д. Разброс сумм его покупок будет достаточно

[7] Числа условные – только чтобы пояснить идею.

[8] Если вспомнить, что эти графики являются графиками функций оценки плотности (после некоторой нормировки), то такая высота по смыслу и есть вероятность угадывания, точнее: площадь под графиком на каком-то отрезке равна вероятности принадлежности суммы покупок этому отрезку.

велик. Изредка, в будние дни вечером после работы он забегает в магазин за хлебом и какими-то продуктами. Ясно, что вряд ли он спонтанно будет покупать ещё мебель, одежду и технику. Поэтому в эти «редкие» дни суммы его покупок ведут себя «стабильно».

Вопрос. Можно ли улучшить качество алгоритма, анализируя его ошибки, т.е. дни, которые он неверно предсказывает, и суммы покупок?
Ответ. Это, кстати, типичная стратегия улучшения алгоритмов – посмотреть на их ошибки. Многие современные методы, например **бустинг**, используют её формализованную реализацию. Боюсь, здесь это практически невозможно. По крайней мере, с помощью просмотра статистики покупок в эти дни. Говорю это, учитывая опыт решения этой задачи участниками конкурса, а их было более 280 человек. Так вот, плохие алгоритмы давали примерно 11% верных ответов, хорошие – 16%, а самый лучший – 18%. Таким образом, чтобы хороший «превратить» в лучший надо добавить 2%, т.е. увидеть закономерности на 2 объектах из 84[9]! Причём мы не знаем, на каких объектах! Конечно, можно процесс поиска как-то автоматизировать, но такая автоматизация не является тривиальной задачей.

Вопрос. Почему мы не учитываем, что клиент на этой неделе вообще может не прийти в магазин? Ведь в его статистике посещений магазина есть недели без визитов.
Ответ. Как я ещё повторю в конце лекции, **в анализе данных всё решает эксперимент**. Мы пробовали учитывать, прироста качества решения это не дало. Кроме того, при вычислении вероятностей первых визитов максимум, как правило, был среди первых семи значений. Когда же он «выскакивал» за ближайшую неделю, он практически никогда не попадал на «правильный» день первого визита. Поэтому мы прекратили эксперименты в этом направлении и сосредоточились на отлаживании нашей простой модели (пусть она неявно и предполагает, что клиент приходит в ближайшие семь дней).

Выводы по первой задаче

Итак, мы видим, что решение нашей проблемы оказалось достаточно простым, необходимо лишь знать некоторые разделы теории вероятностей. Тем не менее, именно такое решение оказалось лучшим на Международном конкурсе [dunnhumby] среди 287 решений. Не смотря на то, что все алгоритмы действовали по схожей схеме, пожалуй, только в этом была учтена специфика конъюнкции «И» в критерии качества решения. **Учёт стабильности поведения клиента помог существенно улучшить результат.**

Действительно, **решать задачи анализа данных несложно**, решения базируются на достаточно простых принципах, но **надо быть внимательным к условиям задачи и требованиям к ответу**!

Замечание. Решение, которое заняло на конкурсе первое место, лишь незначительно отличалось от описанного выше. Вероятности визитов вычислялись чуть сложнее (но по

[9] Имеется в виду, что в среднем на 100 объектов – 16 правильно классифицированных, а 84 = (100 – 16) – неправильно, из которых 2 объекта можно правильно классифицировать.

величине не сильно отличались от тех, что получаются описанным методом). Кроме того, мы не указали, какие именно суммы покупок используются для прогноза в конкретный день недели (это было определено простым перебором различных вариантов). Также в конкурсном алгоритме стабильность поведения пользователя в конкретный день недели не умножалась на вероятность визита в этот день, была использована другая формула, которая имеет эвристическую природу.

Вопрос. В описанном алгоритме прогноз для конкретного клиента осуществляется только на основе его статистики. Почему не учитывается корреляция с другими клиентами? Возможно, некоторые клиенты любят ходить в магазин вместе? Или наоборот, они живут в одной семье, и если один сделал покупку, то другой в этот день не пойдёт покупать.

Ответ. Да, Вы правы, такие ситуации возможны, и прогноз следовало бы делать не по одному клиенту, а используя многомерный временной ряд всех клиентов. Мы даже пробовали делать это в рамках рассматриваемой задачи. К сожалению, никаких корреляций выявить не удалось. Именно поэтому мы сосредоточились на предсказании по локальной информации.

Вопрос. Почему нет учёта специфики дней, на которые надо делать прогноз. Это могут быть праздничные и предпраздничные дни, дни, в которые магазины работают по особому графику и т.д. В такие дни вероятности визитов могут повышаться или понижаться, да и суммы покупок могут отличаться от типичных.

Ответ. Да, Вы снова правы. Более того, такие эффекты хорошо заметны на графиках (см. рис. 10). Но здесь речь была не о построении прогнозной системы, а о решении конкретной задачи. Здесь, как Вы видели, надо было предсказать первый визит в начале апреля, поэтому никаких «особых» дней и странностей в поведении клиентов не ожидалось.

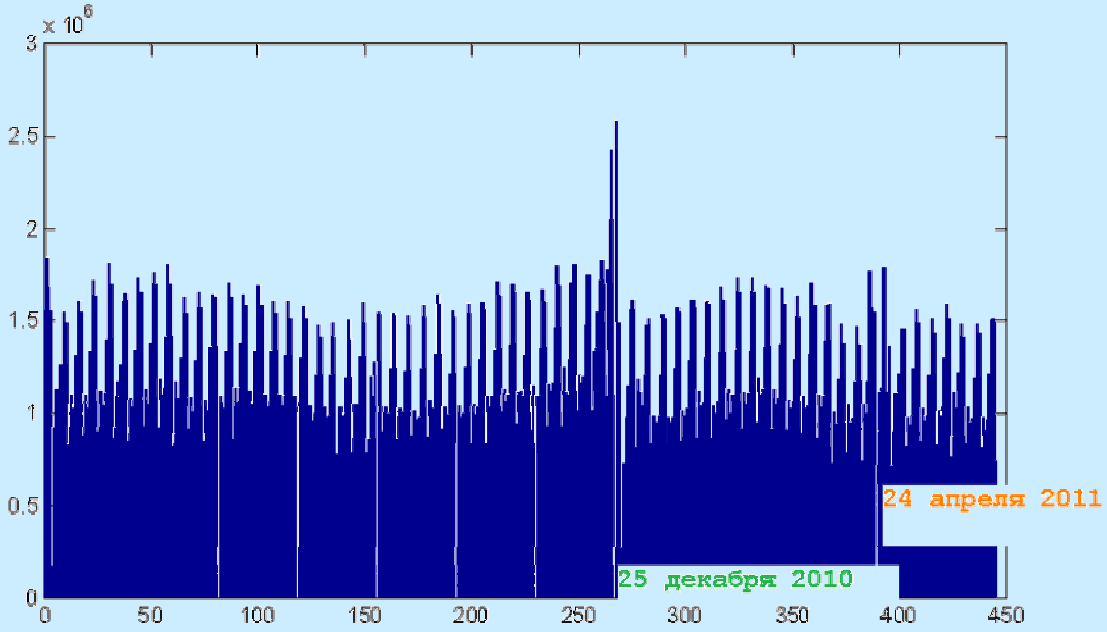


Рис. 10. Выручка магазина по дням («эффект Рождества»).

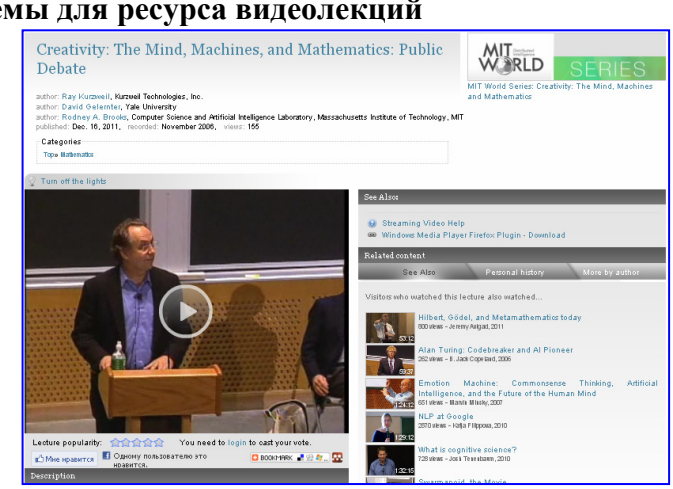
Отметим, что мы описали основную идею алгоритма (и дальше будем описывать

лишь идеи). Описание тонкостей реализации мы опустили, поскольку это займёт время и вызовет много вопросов ненужных на вводной лекции. Но также мы опустили **технологию решения задачи**: почему именно был построен такой алгоритм, какие ещё алгоритмы исследовались и т.д. Это также не тема для вводной лекции, но если кратко, то основные действия таковы:

1. Сначала надо посмотреть на данные (это поможет выдвинуть основные гипотезы о закономерностях, которые в них есть).
2. Затем надо реализовать несколько простейших алгоритмов (которые основаны на увиденных закономерностях или стандартных методах решения подобных задач)
3. Исследовать эти алгоритмы и сами данные (какой подход лучше работает, как работают алгоритмы в совокупности, как оптимальнее делить выборку на обучение и контроль для отладки и т.д.)
4. Написать «каркас» для базового алгоритма на основе сделанных выводов (каркас – это программа, которую легко модифицировать: вводить новые параметры в алгоритм, использовать разные процедуры обработки данных и т.д.)
5. Провести эксперименты с каркасом и его модификациями (здесь и проверяются разные гипотезы об учёте некоторых закономерностях, например из вашего вопроса, даже, если они не были увидены на первых трёх этапах).

Что бы ещё посмотреть?

разработка рекомендательной системы для ресурса видеолекций



Теперь рассмотрим задачу из области построения рекомендательных систем. Это такие программные средства, которые облегчают пользователю выбор. Например, в Интернет-магазине рекомендуют товары для покупки, в социальной сети – группы, в которые следует вступить, а пользователям сотовых компаний – тарифы и услуги. Конечно, подобные рекомендации надо давать на основе информации о пользователе и так, чтобы они были ему полезны, иначе он сочтёт предложения спамом и откажется от пользования данным ресурсом.

Наша задача – рекомендация лекций ресурса VideoLectures.Net[10] [VL] на основе статистической информации о популярности. Обычно такая информация записывается в виде матрицы, число строк которой совпадает с числом пользователей, а число столбцов – с числом услуг, *ij*-й элемент матрицы – информация об использовании *i*-м пользователем *j*-й услуги[11]. В нашем случае пользователи – люди, заходящие на сайт VideoLectures.Net для просмотра лекций, а услуги – сами лекции, в матрице единица означает просмотр пользователем лекции, а ноль – тот факт, что данный пользователь не смотрел лекцию.

---

[10] Это фактически современный «научный YouTube». Здесь выложены видеолекции и доклады ведущих мировых специалистов из разных областей науки.

[11] Это может быть просто факт использования («1» – пользовался, «0» – нет), но часто известна ещё дополнительная информация (когда пользовался, в каком режиме, сколько раз и т.д.).

Такие задачи обычно решают **методами коллаборативной фильтрации**. Основная идея здесь – **похожие пользователи смотрят похожие лекции**, и наоборот, **похожие лекции просматриваются похожими пользователями**. Например, два специалиста по биологии с одной кафедры, наверняка, заинтересуются схожими лекциями. При реализации первой идеи ищут похожих пользователей и рекомендуют то, что чаще смотрели они. Некоторые алгоритмы основаны на специальных приближениях и разложениях матриц «пользователь – лекция».

В нашей задаче всё будет немножко сложнее, поскольку нельзя использовать стандартные методы коллаборативной фильтрации, ведь сама матрица не задана. Данные были представлены на конкурсе «ECML/PKDD Discovery Challenge 2011» [VL Challenge]. При публикации подобных данных в открытом доступе стараются, чтобы по ним нельзя было ничего узнать о конкретном пользователе, поэтому **данные «обезличивают и усредняют»**. В нашем случае организаторы придумали перевести данные в так называемые «пост-троечные последовательности». Это статистическая информация о популярности лекций после просмотра какой-то тройки лекций. Покажем на примере, как они строятся. Пусть какой-то пользователь просмотрел следующие лекции

$$102 \to 33 \to 2 \to 34 \to 35 \to 2 \to 102 \to 17 \to 36,$$

(это последовательность с учётом порядка просмотра), удаляем из неё повторы:

$$102 \to 33 \to 2 \to 34 \to 35 \to 17 \to 36$$

(теперь она отображает первые просмотры лекций). Для тройки {2,33,35} данный пользователь считается посмотревшим все три лекции тройки. После этой тройки он смотрел лекции с номерами из {17,36}. Если пост-троечная последовательность для тройки {2,33,35} выглядит так

7x **{2,33,35}**: 2x **9**, 5x **13**, 3x **17**, 1x **30**, 1x **36**,

то это означает, что семь пользователей просмотрели лекции {2,33,35}, два из них просмотрели лекцию №9 после лекций тройки, пять – лекцию №13 и т.д. Наш пользователь внёс свой вклад в формирование последовательности, прибавив единицу к лекциям №17 и №36[12].

Рис. 11. Формирование пост-троечной последовательности для {2,33,35}.

Можно записать «пост-троечную последовательность» с помощью целочисленного вектора,

$$v(\{a,b,c\}) = (v_1(\{a,b,c\}),\ldots,v_L(\{a,b,c\})),$$

[12] Он, кстати, оказался единственным, кто посмотрел лекцию №36 после лекций из {2,33,35}.

где L – число лекций, $v_j(\{a,b,c\})$ – сколько раз была просмотрена j-я лекция после тройки $\{a,b,c\}$ (неформально говоря, это популярность j-й лекции после просмотра лекций из $\{a,b,c\}$) Наш пользователь «добавляет единицы» к 17-му и 36-му элементам вектора). Пост-троечные последовательности можно использовать для рекомендации. Если пользователь посмотрел три лекции $\{a,b,c\}$, то можно рекомендовать ему лекции, соответствующие максимальным элементам вектора $v(\{a,b,c\})$.

Наша задача ставится следующим образом. Перечислены некоторые пост-троечные последовательности (это **обучающая выборка**), необходимо для новой тройки вычислить пост-троечную последовательность, точнее порядок на её элементах, ведь на практике надо знать наибольшие элементы, чтобы рекомендовать соответствующие лекции человеку, просмотревшему лекции тройки[13].

Опишем простейший метод её решения, чтобы ещё раз продемонстрировать, что решать подобные задачи не очень сложно. Если нет информации о тройке $\{a,b,c\}$, то мы смотрим информацию о тройках $\{a,b,d\}$ для всех d, при которых они входят в обучающую выборку. Они «максимально близки» к нашей тройке (отличаются от неё одним элементом), поэтому и их пост-троечные последовательности будут похожи[14]. Для объединения информации о всех тройках $\{a,b,d\}$ мы суммируем векторы $v(\{a,b,d\})$[15]. В результате получаем вектор $s(\{a,b\})$. Аналогично поступаем для троек вида $\{a,c,d\}$ и $\{b,c,d\}$ – получаем векторы $s(\{a,c\})$, $s(\{b,c\})$.

Элементы пост-троечной последовательности для $\{a,b,c\}$ «должны» встречаться в пост-троечных последовательностях $\{a,b,d\}$, $\{a,c,d\}$, $\{b,c,d\}$, т.е. следует «пересечь полученные информации». Для этого логично взять поэлементный минимум векторов $s(\{a,b\})$, $s(\{a,c\})$, $s(\{b,c\})$, хотя, как было установлено на практике, лучше работает операция поэлементного умножения[16]:

$$s(\{a,b\})\cdot s(\{b,c\})\cdot s(\{a,c\}),$$

а ещё лучше –

$$(s(\{a,b\})+\varepsilon)\cdot(s(\{b,c\})+\varepsilon)\cdot(s(\{a,c\})+\varepsilon),$$

поскольку при таком умножении не происходит зануления большинства элементов вектора (и потери информации)[17].

---

[13] Ясно как организаторы сформировали обучающую и контрольную выборки: они сгенерировали по имеющейся статистике пост-троечные последовательности (для всех самых популярных троек лекций), разбили её на две части, одну – предоставили участникам, вторую – оставили для контроля их решений.

[14] Заметим, что это одна из ключевых гипотез, применяемых при решении задач анализа данных: «**если похожи описания объектов, то похожи и свойства объектов**».

[15] Суммирование соответствует объединению мультимножеств (множеств с кратным вхождением элементов).

[16] Именно эта операция часто используется в теории нечётких множеств для пересечения множеств. Для придумывания подобных алгоритмов полезно знать некоторые разделы теории нечётких множеств и мультимножеств.

[17] Значение параметра ε выбиралось в результате оптимизации качества алгоритма.

Вот, собственно, и всё. Порядок на элементах полученного вектора хорошо соответствует порядку на элементах «настоящего пост-троечного вектора». Рекомендация лекций, соответствующих наибольшим элементам, эффективна примерно на 60%, т.е. в 60% случаях мы угадываем и пользователь подтверждает, что это именно та лекция, которая была ему интересна. Отметим, что наилучший алгоритм для решения этой задачи имеет эффективность 62% и является незначительной модификацией описанного.

Замечание. Модификация, о которой идёт речь, заключается в следующем. Некоторые пост-троечные последовательности короткие, а некоторые – длинные, наверное, будет неверным просто суммировать их при «объединении информации». Также некоторые лекции смотрят существенно чаще, чем остальные (они входят во многие пост-троечные последовательности), что также надо учитывать[18]. Это всё учитывается с помощью нормировок. Например, перед суммированием векторы $v(\{a,b,d\})$ можно поделить на их норму (или на число просмотров тройки $\{a,b,d\}$). На практике просто приходится перебрать разные способы нормировок.

Вопрос. Есть ли список стандартных нормировок, которые приходится перебирать при оптимизации качества алгоритма?
Ответ. Есть несколько стандартных нормировок векторов: деление на норму вектора (например l_1 или l_2), приведение элементов вектора на отрезок [0,1] (т.е. линейное преобразование, которое переводит минимальный элемент в 0, а максимальный – в 1). Но нельзя считать это списком всех нормировок, поскольку на практике иногда здорово работают какие-то «экзотические». Кроме того, часто нормировку сочетают с другими преобразованиями. И ещё, надо учитывать, что в общем случае нормируются не векторы, а строки матриц, поэтому применяют и всякие постолбцовые преобразования, см., например, преобразования типа TF*IDF [Маннинг и др., 2011].

Вопрос. Почему при решении задачи мы никак не учитывали описания лекций. Неужели они тоже были недоступны? Ведь их использование могло существенно улучшить качество!
Ответ. Они были доступны, но их использование не улучшило качество алгоритма! Это кажется удивительным, но подтверждается каждый раз при решении реальных задач: **ЕСЛИ ИЗВЕСТНО ХОРОШЕЕ СТАТИСТИЧЕСКОЕ ОПИСАНИЕ ОБЪЕКТА, ЕГО ВЗАИМОДЕЙСТВИЯ С ДРУГИМИ ОБЪЕКТАМИ, ТО ПРИЗНАКОВОЕ ОПИСАНИЕ НЕ УЛУЧШАЕТ ПРОГНОЗИРОВАНИЕ ЕГО ПОВЕДЕНИЯ**. Первый раз я столкнулся с этим при анализе поведения в социальных сетях (ниже мы рассмотрим одну из таких задач). Если вы знаете, что какой-то пользователь каждый день качает музыку, то завтра он будет делать то же самое. Это максимально надёжный прогноз. И не важно, как у него заполнена анкета на персональной страничке, его прошлое поведение определяет будущее. Вот, если бы мы не знали его прошлое поведение, пришлось бы анализировать признаки.

[18] Похожие проблемы возникают в анализе данных при обработке текста. К счастью, там есть рецепт, проверенный годами: TF*IDF–преобразование [Маннинг и др., 2011].

В данной задаче пост-троечные последовательности оказались хорошим статистическим описанием поведения пользователей и популярности лекций. Это достаточно удивительно, **стоит повнимательнее присмотреться к этой математической модели**.

В табл. мы приводим пример пост-троечной последовательности. Как видно, она совсем «не очевидна»: три заглавные лекции из разных тем, а в самой пост-троечной последовательности содержатся лекции из совершенно других тем. Например, самая популярная лекция – по кластеризации, что никак не связано с темами остальных лекций.

Автор, **область, кратность** (в п-т посл-ти)	**Название**
Anastasia Krithara Text Mining	Active, Semi-Supervised Learning for Textual Information Access
Isabelle Guyon Machine Learning	Introduction to Machine Learning
Mikaela Keller Statistics	Basics of probability and statistics
Ulrike von Luxburg, Clustering, **5x**	Lectures on Clustering
William Cohen, Text Mining, **4x**	Text Classification
John Shawe-Taylor, Statistical Learning, **3x**	Statistical Learning Theory
Cynthia Rudin, Boosting, **3x**	The Dynamics of AdaBoost

Табл. Пример пост-троечной последовательности.

Вопрос. Насколько разумно такое огрубление информации, которое предложили организаторы соревнования: использование троек? Неужели предоставление всей матрицы «пользователи–лекции» нарушает какие-то правило конфиденциальности? Ведь пользователи всё равно обезличены?

Ответ. Как ни удивительно, но при наличии такой матрицы можно установить, какая строчка, какому человеку соответствует. Мы не будем останавливаться на методах такого восстановления. Собственно, это уже не совсем «анализ данных», однако упомянем об одном интересном случае… На соревновании [SN Challenge] (см. ниже) была предложена матрица смежности графа социальной сети [flickr]. Вершины графа – пользователи и фотографии, а рёбра – различные отношения между ними. Например, факт отметки пользователя на фотографии соответствовал ребру. Интересно, что одна из команд, участвовавших в соревновании, восстановила информацию (что именно соответствует каждой вершине) с достаточно большой точностью. Это, кстати, и позволило ей выиграть.

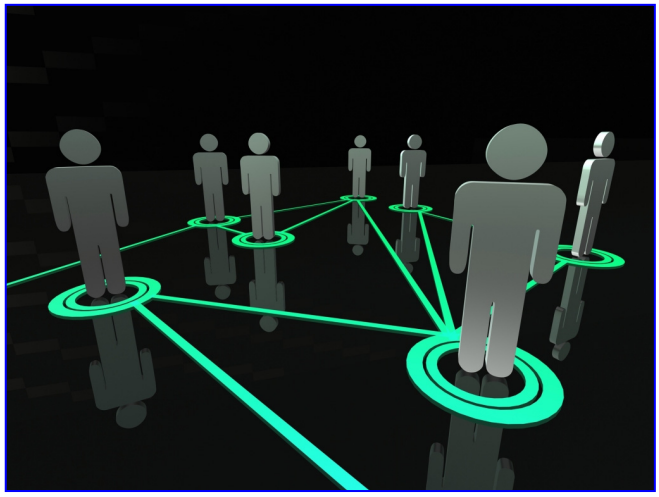
На самом деле, **«правильное обезличивание данных» – новое и очень актуальное направление исследований**. Нужно уметь преобразовывать исходную информацию так, чтобы по ней нельзя было восстанавливать персональные данные людей, но при этом можно было решать задачи анализа данных с высокой точностью.

Вторая задача конкурса [VL Challenge] была гораздо сложнее. Теперь у нас нет никакой статистической информации. Это случается, когда в систему (в данном случае на ресурс VideoLectures.Net) вошёл новый пользователь и о нём нет никакой информации, кроме его первых действий. Ему надо рекомендовать лекции, о которых также нет статистической информации, например, недавно загруженные на сайт. Такая потребность часто возникает, когда приходится рекомендовать новые товары и услуги (они новые и спрос на них не известен)[19]. Что же тогда дано? Достаточно много… мы знаем описания лекций: названия, аннотации, слайды, тексты, их рубрики, авторов (включая адреса электронной почты и персональные сайты), даты съёмки. Это известно для новых лекций и для лекции, которую пользователь начал смотреть на ресурсе VideoLectures.Net, однако у отдельных лекций некоторые данные могут отсутствовать (не у всех лекций выложены слайды и тексты), что вносит дополнительные трудности в задачу. Такие задачи решаются **контентно-ориентированными методами**. Используя нашу традиционную гипотезу «похожие лекции должны быть примерно одинаково интересны», логично рекомендовать новые лекции, похожие на ту, что просмотрел пользователь. Основная проблема: **что значит «похожесть», как её вычислить**? Во вводной лекции мы не будем описывать методы определения похожести и алгоритм решения задачи, поскольку это потребует достаточно долгого описания технологии решения целого класса задач. Отметим только, что, например, сравнить заголовки лекций – задача не совсем тривиальная. Так, вхождение одинаковых слов в заголовки не говорит о похожести самих лекций. Вот несколько примеров заголовков:

«Байесовские сети: теория и применение»,
«Применение нейронных сетей в задачах социологии»,
«Применение PHP при создании социальных сетей»,
«Запрет на применение рыболовных сетей: юридические аспекты».

С кем бы подружиться?

предсказание действий пользователя социальной сети и рекомендации друзей



Когда меня учили анализу данных, то чаще всего рассматривали задачи в т.н. «стандартных постановках», когда информация задана матрицей «объект-признак»[20]. Надо сказать, что за всю свою жизнь я лишь два раза сталкивался с задачами в стандарной постановке (это были задачи скоринга, и, видимо, такая постановка просто диктуется традицией банковского анализа данных). Все остальные оказались «нестандартными». Вот ещё одна из них…

[19] Подобные задачи называются «cold start» или «new user – new item».

[20] Так обычно ставятся задачи классификации, регрессии и кластеризации: объекты записаны по строчкам матрицы «объект–признак», в первых двух задачах выделяют также целевой признак (метки классов или регрессионные метки), значения которого известны лишь для части объектов (обучающей выборки). Также рассматривают задачи, в которых объекты заданы попарными расстояниями или

В математике «социальная сеть» – это динамический граф[21]. Если представить реальных пользователей какой-нибудь «традиционной» социальной сети («вконтакте», «одноклассники» или «фейсбук») вершинами графа, а отношения дружбы между ними – рёбрами, то получим пример такого динамического графа: постоянно добавляются и исчезают вершины и рёбра. Кстати, в общем случае, вершины – не обязательно пользователи, некоторые из них могут быть группами, а соединение пользователя ребром с вершиной-группой обозначает факт вступления в группу. Ещё пример: пользователи сотового оператора и услуги являются вершинами, а звонки и использование услуг – рёбрами. Аналогично с клиентами банка, Интернет-магазина и т.д.

Очень актуальной задачей является **предсказание, как граф будет меняться в ближайшее время**. В упрощённой постановке – какие рёбра будут появляться и исчезать. Ведь если мы знаем, что пользователь сети подключит некоторую услугу, то мы сможем предложить её ему раньше:
- он оценит заботу оператора (актуальное предложение, которое не является «спамом»),
- начнёт пользоваться услугой раньше (и раньше за неё платить).

Если же мы предскажем, что кто-то отключит услугу, то можем попробовать его удержать от этого шага (предложить более выгодные условия) или предоставить альтернативную услугу. В социальных Интернет-сетях хорошее предсказание появления рёбер позволит сделать правильную рекомендательную систему «кого Вам зафрендить». Именно о такой задаче – **предсказание появления рёбер динамического графа** на примере социальной Интернет-сети – и пойдёт дальше речь. В литературе она называется «**Link Prediction Problem**[22]» (LPP).

Задача LPP для данных социальной сети [flickr] была предложена участникам соревнования «IJCNN Social Network Challenge». Был задан граф в фиксированный момент времени, его удобно задавать матрицей смежности, т.е. матрицей размера $q \times q$, где q – число вершин, а ij-й элемент равен единице, если i-я вершина соединена с j-й. Предложенный граф не совсем соответствовал реальному: из реального графа социальной сети изъяли 4480 рёбер, участникам предложили множество из 8960 пар вершин, половина из них – изъятые рёбра, а другая половина – пары не являющиеся рёбрами. Необходимо отличить рёбра от «не-рёбер», т.е. выдать 8960 значений чисел из отрезка [0,1], которые соответствуют «уверенности алгоритма» в том, что соответствующие пары вершин являются рёбрами.

Замечание. В более традиционной формулировке требовалось бы выдать значения из множества {0,1}, т.е. ответы алгоритма «ребро», «не ребро». Сейчас всё чаще от алгоритмов требуют оценку принадлежности к классу (неформально её можно

классы описаны вероятностными распределениями, см. также [Дьяконов, 2010]. В любом случае, на практике всё гораздо сложнее…

[21] Подробнее о теории социальных сетей см. на Википедии http://en.wikipedia.org/wiki/Social_network. Напомним, что граф – пара (V, E), где V – множество вершин, $V \neq \varnothing$, а $E \subseteq V \times V$ – множество рёбер.

[22] К сожалению, в русскоязычной литературе эта задача практически не встречается, хотя её актуальность не вызывает сомнений.

интерпретировать как вероятность, хотя надо учитывать также функционал, с помощью которого оценивается правильность ответа). В этой задаче решение оценивалось с помощью функционала AUC-ROC [Дьяконов, 2010], [Воронцов, 2010].

Отметим также, что многие современные алгоритмы классификации вместе с ответом о принадлежности к классу автоматически получают оценку такой принадлежности.

Граф имеет гигантские размеры: число вершин ~ 1′100′000, число рёбер ~ 7′200′000. Кроме того, граф «почти двудольный»: его вершины можно разделить на два множества A, B. Вершины множества B попарно не соединены рёбрами, а остальные связи допускаются. Множество A соответствует пользователям, B – фотографиям, связи означают отношения пользователей и отметки пользователей на фотографии. Кстати, такие отметки наиболее частые события.

Как решаются подобные задачи? Оказывается, тоже просто. Допустим, у нас есть граф обычной социальной сети, в которой вершины – пользователи, а рёбра соответствуют отношениям дружбы. **Первая идея** решения реализует **принцип «друг моего друга»**:

если Иван дружит с Сергеем, а Сергей с Петром,
то Иван подружится (дружит) с Петром
или
если (A, B) – ребро, (B, C) – ребро,
то (A, C) – ребро или станет ребром.

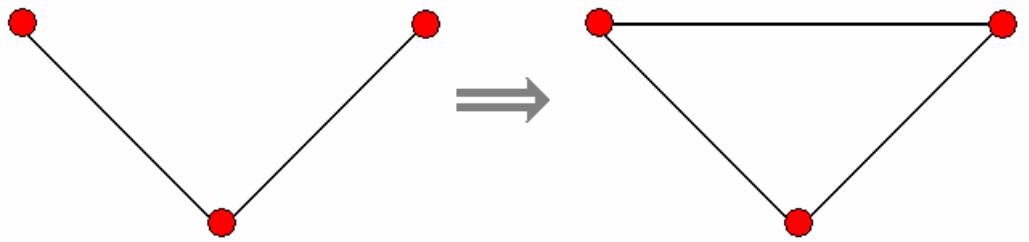


Рис. 12. Иллюстрация принципа «друг моего друга».

В нашем «почти двудольном» графе его придётся чуть-чуть подкорректировать: если на одной фотографии отмечены Мария и Анна, на другой отмечена Мария, то, скорее всего, там есть и Анна, см. рис. 13.

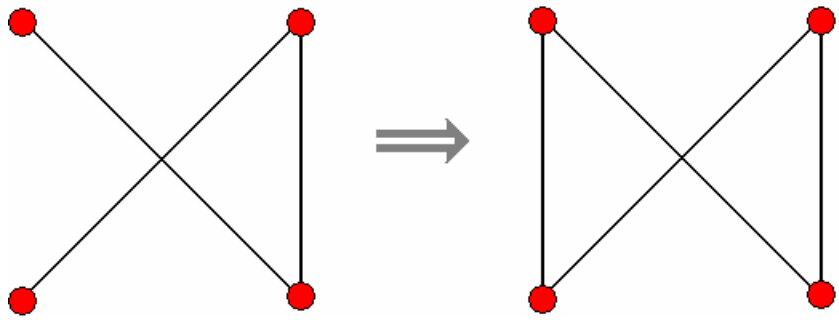


Рис. 13. Аналог принципа «друг моего друга» для двудольных графов.

Заметим, что чем больше общих друзей имеют Иван и Пётр, тем более вероятней, что они подружатся. Пусть $\Gamma(x)$ – множество соседей вершины x, тогда число вершин

$|\Gamma(x) \cap \Gamma(y)|$ смежных с вершинами x и y является хорошей мерой похожести вершин[23]. Иногда используют величину $|\Gamma(x)| \cdot |\Gamma(y)|$, называемую **коэффициентом предпочтительности**. Чтобы получить значение похожее на вероятность (по крайней мере, из отрезка [0,1]), используют нормировку:

$$\frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}.$$

Это выражение называется **коэффициентом Жаккара**. Также надо учесть, что не все общие друзья свидетельствуют о возможной дружбе. Например, в социальных сетях есть достаточно общительные люди, которые «френдят всех подряд[24]» (наличие такого общего друга не свидетельствует о возможности дружбы), поэтому надо учитывать число друзей у каждого общего друга. В математических терминах это можно формализовать с помощью **коэффициента Адамик/Адара**:

$$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|}.$$

Все рассмотренные признаки нетрудно распространить на случай нашего «почти двудольного» графа. Мы оставляем это в качестве домашнего задания, а сами рассмотрим ещё несколько классических методов решения задачи LPP. Если обобщить рассмотренную идею «друг моего друга», то можно учитывать целые цепочки общих друзей. В математических терминах это можно формализовать **признаком Katz**:

$$\sum_{l=1}^{\infty} \beta^l \operatorname{path}_l(x, y),$$

где $\operatorname{path}_l(x, y)$ – число путей длины l между вершинами x и y. На практике суммируют по небольшим значениям l или пользуются «волшебной» формулой: признак равен xy-му элементу матрицы

$$(I - \beta M)^{-1} - I,$$

где M – матрица смежности графа[25]. Также можно устроить случайные блуждания по графу из вершины x, вероятность попадания в вершину y будет вероятностью появления ребра между этими вершинами. Можно это проиллюстрировать так: Иван стал навещать своих друзей, они его стали звать к своим друзьям и т.д. Подобные идеи реализованы в известном алгоритме **PageRank**. Можно устраивать случайные блуждания из вершины y или сразу из двух вершин: x и y.

Итак, мы придумали достаточно много эвристических оценок вероятности появления ребра[26]. Какую же из них выбрать? На практике часто выбирают «все сразу». Как раз при этом и переходят к стандартной постановке. Для каждой пары вершин (x, y), между которой нас интересует возможность появления ребра, вычисляют значения всех

[23] Это стандартная терминология. Можно сказать, что мы ищем вершину y максимально похожую на вершину x, чтобы соединить их ребром. Поэтому оцениваем их похожесть. Функции такой оценки часто называются мерами похожести (сходства). Ничего общего с математической «мерой» они не имеют.

[24] Например, когда в социальной сети появляется какой-нибудь известный человек (президент, футболист или поп-звезда), многие пользователи «френдят» его. Конечно, это не означает даже, что их интересы схожи, и что они смогут сами подружиться (тем более, через этого известного человека).

[25] Попробуйте доказать эту формулу.

[26] Здесь термин «вероятность» употребляется образно…

перечисленных характеристик. При этом мы можем также сделать это для настоящих рёбер и не-рёбер. Поэтому у нас есть обучающая выборка: перечень значений признаков, соответствующих рёбрам и не-рёбрам. По этой выборке надо понять закономерность: как значения признаков определяют вероятность появления ребра. В рассматриваемой задаче всё относительно просто: чаще используют линейную закономерность, т.е. ищут линейную комбинацию значений признаков, которая соответствует вероятности появления ребра. Методы построения такой линейной комбинации мы рассмотрим в следующих лекциях. Таким образом, **наша изначальная «нестандартная» задача свелась к стандартной признаковой задаче классификации**. Такое сведение – достаточно универсальный метод, и здесь он действительно здорово работает.

Вопрос. Можно ли любую задачу свести к задаче в стандартной постановке?
Ответ. В интенсивно развивающихся прикладных науках сложно говорить про «любую» задачу, поскольку каждый день появляются всё новые и новые. Практически все задачи, с которыми я сталкивался, можно было свести. Но тут надо чётко определить, что мы понимаем под сведением. Дело в том, что есть даже целая теория о каноническом виде алгоритмов [Журавлёв, 1998], из которой, в частности, следует, что сведение всегда существует (и неявно выполняется любым алгоритмом, решающим задачу). Но часто оно искусственно, и метод сведения только усложняет задачу. В нашем примере – оно естественно, кроме того, задачу LPP пока не научились решать без подобного сведения.

Вопрос. Сколько признаков надо придумать, чтобы решение в стандартной постановке было возможно? И какие это должны быть признаки? Наверное, есть ещё много, которые мы не успели рассмотреть.
Ответ. Это очень «неудобный» вопрос, поскольку не существует универсальных рекомендаций по генерации признаков. Считается, что **профессионализм аналитика данных** как раз и **заключается в том**, чтобы в подобных задачах **правильно выбрать признаковое пространство**. Признаюсь, эту задачу я как раз решил не очень успешно: недооценил возможности признаков типа PageRank[27], хотя и придумал много своих «оригинальных». Основная рекомендация здесь: чтение литературы и фантазия. Многое зависит от того, какими методами решать стандартную задачу. Некоторые методы очень чувствительны к шумовым признакам (в которых нет закономерностей, помогающих решить задачу), поэтому для них нужно внимательно подходить к генерации признакового пространства.

Заключение

Как стать аналитиком данных?

Мы рассмотрели несколько задач и указали методы их решения. Конечно, это далеко не полный перечень задач анализа данных и на следующей лекции мы продолжим. Но зато наши задачи достаточно оригинальные. Никогда не понимал тех, кто

[27] И это послужило для меня хорошим уроком! Такие уроки и составляют опыт…

в качестве примера подобных задач приводит «ирисы», задачи из UCI-репозитория[28] или перечень: классифицировать болезнь, решить задачу скоринга, спрогнозировать курс акции. Каждый день появляются новые задачи, интересные, неожиданные. Буквально только что я закончил решать задачу о прогнозировании ответов студентов на вопросы тестов [What Challenge]. Никогда раньше не думал о подобной постановке задачи, а она имеет смысл, поскольку позволяет построить рекомендательную систему. Эта система, учитывая знания студента (статистику ответов на тесты раньше), пробегает по всем вопросам, находит «проблемные» и рекомендует студенту повторить соответствующие темы! Интересно также, что развита целая область науки, которая занимается подобными задачами. Не каждый правильно ответит, что важнее знать для предсказания правильности ответа студента: его средний балл, сложность вопроса (средний балл, вычисленный по ответам на него) или время, которое потратил студент на ответ[29].

Что же на этой лекции не удалось охватить? Увы, как принято для вводных лекций – главного:

- как именно «догадаться» до описанных решений,
- что делать с данными: как их загружать, хранить и обрабатывать,
- в какой среде программировать алгоритмы,
- как эти алгоритмы тестировать и т.д.

Всё это – темы следующих лекций, всё это – уже описание технологии решения задач. Причём даже технолог**ий**, поскольку есть разные подходы. Одни сосредоточены на извлечении правильных закономерностей из данных (что-то похожее сделано при решении первой задачи), другие – на генерации признаков и сведении к стандартным постановкам (последняя задача). А есть также задачи, для которых нет стандартных методов, и приходится всё выдумывать заново (вторая задача).

Напоследок хотелось бы дать несколько советов тем, кто заинтересовался анализом данных и решил изучить эту область подробнее. Во-первых, **анализ данных это практика, практика и ещё раз практика**. Надо решать реальные задачи, много, из разных областей. Поскольку, например, классификация сигналов и текстов две совершенно разные области. Специалисты, которые с лёгкостью построят алгоритм диагностики двигателя на основе сигналов датчиков, возможно, не смогут сделать простейший спам-фильтр для электронных писем. Но очень желательно получить базовые навыки при работе с разными объектами: сигналами, текстами, изображениями, графами, признаковыми описаниями и т.д. Кроме того, это позволит вам выбрать задачи по душе.

Во-вторых, **важно грамотно выбрать себе учебные курсы и наставников**. В принципе, можно всему научиться самому. Ведь мы не имеем дело с областью, где есть какие-то секреты, передающиеся из уст в уста. Наоборот, есть много грамотных

[28] Репозиторий реальных и модельных задач машинного обучения, созданный в университете г.Ирвин (Калифорния, США). http://archive.ics.uci.edu/ml/ Одна из задач репозитория – «Ирисы Фишера» (на примере этой задачи Р. Фишер демонстрировал эффективность дискриминантного анализа).

[29] На самом деле, важнее знать время, также важна сложность вопроса, а вот средний балл студента оказывается плохим признаком в этой задаче. Правда, если вычислять средний балл по ответам на похожие вопросы, вопросы из такой же темы и т.д., то он становится очень хорошим.

учебных курсов[30], исходников программ и данных. Однако есть также много-много тонкостей, которые не описаны ни в одном учебнике (например, в нашей первой задаче – вычисление стабильности). Кроме того, очень полезно, когда одну задачу решают несколько людей параллельно. Дело в том, что при решении таких задач приходится сталкиваться с очень специфическим программированием. Допустим, ваш алгоритм выдал 89% верных ответов. Вопрос: много это или мало? Если мало, то в чём дело: вы неправильно запрограммировали алгоритм, выбрали неверные параметры алгоритма или сам алгоритм плохой и не подходит для решения данной задачи? Если работа дублируется, то ошибки в программе и неверные параметры удаётся быстро найти. А если она дублируется специалистом, то вопросы оценки результата и приемлемости модели тоже решаются быстро.

В-третьих, полезно запомнить, что **на решение задачи анализа данных требуется много времени**. Конечно, есть исключения. Задачу с пост-троечными последовательностями я обдумывал ровно 15 секунд, саму программу написал за 15 минут. Правда, целый день пришлось потратить на перебор разных нормировок (хотя этот процесс можно автоматизировать). И то, такой перебор пришлось сделать только после того, как меня обошли в рейтинге на конкурсе, в котором задача была представлена. Но это, пожалуй, единственное исключение. Уже второй задаче конкурса (рекомендация лекций по контенту) пришлось уделить гораздо больше времени, поскольку пришлось писать процедуры обработки текстов, их сравнения, вводить разные меры сходства лекций и т.д. и т.п. К счастью, пот***о***м это время окупается. После приобретения опыта то, что раньше отнимало 2 часа, теперь отнимает 20 минут.

И, наконец, повторим основное правило анализа данных: **всё решает эксперимент**. Это просто звучит, но долго приходится объяснять студентам. Они часто меня засыпают вопросами: какие значения параметров алгоритма выбрать, как разбить выборку на обучение и контроль, нужно ли удалять шумовые объекты и т.д. Ответ один: надо попробовать по-разному! Попробовать разные значения параметров, попробовать разные разбиения, попробовать удалять шумовые объекты и оставлять их. И в эксперименте выбрать лучшую стратегию.

Автор благодарен Илье Толстихину, Наталье Дышкант, Михаилу Фигурнову, Екатерине Малышевой и Елене Платоновой за вопросы и замечания.

[30] Например, [Воронцов, 2010].

[KAGGLE] http://www.kaggle.com
Платформа для проведения соревнований по интеллектуальному анализу данных.

[dunnhumby] http://www.kaggle.com/c/dunnhumbychallenge/
Страничка соревнования по прогнозированию визитов клиентов супермаркета и сумм покупок «dunnhumby's Shopper Challenge».

[Дуда, Харт, 1976] *Дуда Р., Харт П.* Распознавание образов и анализ сцен. – М.: Мир, 1976.

[VL] http://www.VideoLectures.Net
Сайт-репозиторий видеолекций.

[VL Challenge] http://tunedit.org/challenge/VLNetChallenge?m=summary
Страничка соревнования «VideoLectures.Net Recommender System Challenge (ECML/PKDD Discovery Challenge 2011)» по разработке рекомендательной системы для ресурса VideoLectures.Net.

[Маннинг и др., 2011] *Жаккар Маннинг К., Рагхаван П., Шютце Х.* Введение в информационный поиск – Вильямс, 2011.

[flickr] http://www.flickr.com/
Сайт социальной сети с возможностями по обмену фотографиями.

[SN Challenge] http://www.kaggle.com/socialNetwork?viewtype=results
Страничка соревнования «IJCNN Social Network Challenge» по прогнозированию появления рёбер в динамическом графе социальной сети.

[Дьяконов, 2010] http://www.machinelearning.ru/wiki/images/7/7e/Dj2010up.pdf
Учебные пособия (в одном pdf-файле)
Дьяконов А.Г. Практикум на ЭВМ кафедры математических методов прогнозирования (логические игры, обучение по прецедентам): Учебное пособие. – М.: Издательский отдел факультета ВМиК МГУ им. М.В. Ломоносова; МАКС Пресс, 2010. – 164с.: ил. (ISBN 978-5-89407-431-3)
Дьяконов А.Г. Практикум на ЭВМ кафедры математических методов прогнозирования (системы WEKA, RapidMiner и MatLab): Учебное пособие. – М.: Издательский отдел факультета ВМиК МГУ им. М.В. Ломоносова; МАКС Пресс, 2010. – 133с.: ил. (ISBN 978-5-89407-432-0)

[Журавлёв, 1998] *Журавлёв Ю.И.* Избранные научные труды. – М.: «Магистр», 1998. – 420 с.

[What Challenge] http://www.kaggle.com/c/WhatDoYouKnow/
Соревнование по предсказыванию ответов студентов на тесты.

[PageRank] *А. Шкондин* PageRank: Больше ссылок хороших и важных
http://www.developing.ru/seo/pagerank.html

[Воронцов, 2010] *Константин Воронцов* Курс лекций Математические методы обучения по прецедентам, МФТИ, 2004-2008. см. www.MachineLearning.ru

[Дьяконов, 2011] *Дьяконов А.Г.* Научно-популярная лекция «“Шаманство” в анализе данных»
http://alexanderdyakonov.narod.ru/lpotdyakonov.pdf

[ML] www.MachineLearning.ru Вики-ресурс, посвященный машинному обучению и интеллектуальному анализу данных.

Иллюстрации взяты с сайтов:
http://status-delovoy.com.ua/news.php?news_id=642
http://www.VideoLectures.Net.com
http://magazynt3.pl/Vademecum-blogera/
иллюстрация на первой странице – скриншот со столбцовыми диаграммами в системе MATLAB (получены при решении задачи о прогнозировании визитов клиентов)

Также использованы материалы из раздела **«ЛИТЕРАТУРА И ССЫЛКИ слушателям»**, в большей степени
[KAGGLE] http://www.kaggle.com
[VL Challenge] http://tunedit.org/challenge/VLNetChallenge?m=summary
[Маннинг и др., 2011] *Жаккар Маннинг К., Рагхаван П., Шютце Х.* Введение в информационный поиск – Вильямс, 2011.

ГЛОССАРИЙ
к лекции «ВВЕДЕНИЕ В АНАЛИЗ ДАННЫХ»

Анализ данных (data mining) – наука об обнаружении в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности, а также процесс такого обнаружения. Подразделяется на задачи классификации, прогнозирования и другие.
(из Википедии http://ru.wikipedia.org/wiki/Data_mining)

Классификация – отнесение объекта к одному из заранее известных классов.

Обучение (обучающая выборка) – выборка, по которой производится настройка (оптимизация параметров) алгоритма анализа данных (алгоритма классификации, регрессии или кластеризации).
http://www.machinelearning.ru/wiki/index.php?title=Выборка

Контроль (контрольная выборка) – выборка, по которой оценивается качество построенного алгоритма.

Качество алгоритма определяет его способность решать поставленную задачу с приемлемой точностью. Обычно формализуется каким-то функционалом, который вычисляется по ответам алгоритма и верным ответам (например, процентом правильных ответов).

Рекомендательная система – программные средства, которые предсказывают, какие объекты (товары, фильмы, музыка, книги, веб-сайты и т.д.) будут интересны пользователю.

Граф – это совокупность непустого множества вершин и множества пар вершин (множества рёбер).

Двудольный граф – граф, множество вершин которого можно разбить на две части таким образом, что каждое ребро графа соединяет какую-то вершину из одной части (**доли**) с какой-то вершиной другой части, то есть не существует ребра, соединяющего две вершины из одной и той же части.