Санкт-Петербургский государственный Политехнический университет
(СпбГПУ Политех)
Институт прикладной математики и информатики

Кафедра Прикладная математика и информатика

наброски квалификационной работы

Автоматическая интерпретация результатов коллаборативной фильрации

Выполнил Кокарев В.В.

Спб. 2015

Оглавление

0.1 Введение

В большом числе отраслей (продажа книг, фильмов, мобильных приложения, бытовой технике и т.д.) количество контента, из которого пользователь может выбрать очень велико. Например, в магазине приложений 'ITUNES', находится более $2 * 10^6$ различных приложений. Очевидно, что обычный человек не может просмотреть весь ассортимент предлагаемых товаров и может отказаться от покупки, если не найдет среди всего многообразия товаров тот, который его (пользователя) заинтересует.

Магазины прибегают к множеству ухищрений, что бы помочь пользователю. Например, добавляют возможность фильтрации по каким-либо критериям (категория, популярность, цена и т.д.) или добавляют поиск по товарам. Одним из наиболее эффектвных приемов является таргетированный показ товаов на главной странице или показ таргетировнной рекламы. [todo вставить определение таргетирования]

Таргетированная реклама (или выдача) способна на несколько порядков [ссылка на источник] увеличить число покупок, а так же сократить время, которое требуется пользователю для выбора товара. Как правило, таргетирование осуществляется с помощью различных алгоритмов машинного обучения, которые ранжируют (сортируют) все многообразие товаров по (предсказанной) вероятности их приобритения конкретным пользователем, основываясь, как на известную информацию про конкретного человека (историю его покупок, предпочтения, возраст, материальный достаток и т.д.), так и на характеристики конкретного товара (цена, популярность, ообласть применимости и др.).

Очевидным способом увеличения подаж является улучшение качества рекомендаций. Есть 2 направления для улучшения качества рекомемендаций:

- увеличения объема данных, на которых собирается статистика (далее обучающее множество)
- улучшение самого алгоритма ранжирования

Улучшение или разработка нового алгоритма - процесс, требующий значительных усилий на исследования и нет никаких гарантий успеха (зачастую, большие материальные затраты на исследования дают очень маленький прирост в качестве, который не окупается). Так же существует логарифмееческая зависимость между размером обучаещего множества и качеством результирующей модели (с помощью которой производится таргетирование), это означает, что бы улучшить качество в 10 раз, необходимо увеличиь размер обучающего множества в 100 раз [ссылка на источник]. Т.е. оба способа улучшения качества рекомендаций дают маленький прирост при большой трудоемкости. Кроме того, каждое следующее улучшение требует еще больше ресурсов и дает меньший прирост качества, поэтому необходимо искать альтернативные спобы увеличения числа покупок том же качестве рекомендаций (фиксированном алгоритме ранжирования и фиксированном обучающем множестве).

Психологи доказали, что люди намного охотнее принимают советы, содержащие обоснование. Например, сопроводительный текст к рекомендации "Мы советуем Вам установить приложение XYZ, потому что оно нравится пользователям, похожим на Вас"является более предпочительным (с точки зрения продуктовых метрик), чем "Мы советуем Вам установить приложение XYZ". Интуитивно понятно, что текст "оно нравится пользователям, похожим на Вас вызывает гораздо меньше доверия у пользователя, чем "оно нравится Вашему другу Ивану Иванову"или "оно похоже на YXZ_{free}, которым Вы пользуетесь". Отсюда рождается гипотеза, что улучшая качество сопроводительного текста (обоснования рекомендации) можно увеличить уровень продаж, а так же увеличить лояльность пользователя к конкретному магазину.

0.2 Неформальная постановка задачи

Обычно алгоритмы ранжирования работают по принципу черного ящика и дают ответ без какого-либо обоснования (исключением являются knowlage-based алгоритмы, но спектр их применения крайне ограничен). Необходимо построить алгоритм, который способен для каждого конкрентного пользователя подобрать наиболее релевантное человеко-интерпритируемое объяснение предложенных рекоммендаций.

Т.к. невозможно формализовать понятие релевантности соповодительного текста, для сравнения качества двух алгоритмов предлагается сравнивать величины, которые по нашему предположению должны сильно коррелировать с релевантностью сопроводительного текста, например, число покупок.

0.3 Формальная постановка задачи

Пусть:

- $\mathbf{A}$ - множество всех товаров, дополненное фиктивным товаром $\emptyset_A$
- $\mathbf{U}$ — множество всех пользователей, дополненное фиктивным пользователем $\emptyset_U$
- $\mathbf{L}$ — множество всех возможных сопроводительных текстов (или шаблонов текстов)
- $Ml : \mathbf{U} \to \mathbf{A}^n$ — алгорим таргетирования, который для каждого пользователя возвращает n рекомендаций товаров.
- T — некторая функция качества [TODO: T - ОНЛАЙН МЕТРИКА В A/B ЭКСПЕРИМЕНТЕ. АККУРАТНО ВВЕСТИ ПОНЯТИЕ ОНЛАЙН МЕТРИК, ОПИСАТЬ КАК ДЕЛАЕТСЯ A/B ТЕСТИРОВАНИЕ]

Задача:
Построить отображние $\mathbf{U} \times Ml \to \mathbf{L}^n$, которое для каждой пары пользователь u и приложения $a \in Ml(u)$ выбирает такой сопроводительный текст $l \in \mathbf{L}$, что $T \to max$. [TODO корректно ли ставить задачу оптимизации, если мы не собираемся честно оптимизировать T, а собираемся предложить способы построения алгоритма, увеличивающего зн-е T, оносительно некоторого base-line???]

0.4 Уточнение предметной области

Не умаляя общности, в данной работе будет рассматриваться магазин мобильных приложения для платформы Android. Это необходимо для более полного описания исходных данных и понимания предложенных оптимизаций. Однако, предложенные в работе принципы могут быть применены к произвольной предметной области.

0.4.1 Исходные данные

Для магазина мобильных приложений все знания можно условно разделить на 3 типа:

- зания про пользователей
- знания про приложения
- знания про пару пользователь - приложение (статистика использования конкретного приложения конкретным пользователем).

Знания про приложения можно получить из магазина приложений Google Play. К таким заниям относятся:

- категория (игры, навигация, etc)
- текстовое описание
- комментарии пользователей
- средняя оценка пользователей
- кол-во установок
- версии операционной системы, для которых доступно данное приложение
- etc.

Так же можно извлечь данные из анонимной статистики, собираемой на устройствах пользователя:

- среднее время от открытия до закрытия приложения
- распределение числа запусков по времени суток
- кол-во установок и удалений за период времени
- среднее время от установки на устройство пользователя до удаления
- etc

Знания про пользователя можно извлечь как из статистики с устройства пользователя, так и анализируя поведение пользователя в магазине (в приложении или на веб-интерфейсе):

- список установленных приложений
- установки/запуски/удаления приложений с привязкой ко времени и координатам
- факт просмотра конкретного приложения в магазине
- модель телефона
- версия ОС
- etc.

Для проведения дальнейших рассуждений необходимо принять гипотезу о том, что собираемые данные про пользователя (приложение) являются достаточно полным описанием, т.е. позволяют судить о предпочтениях, вкусах и интересах пользователя.

0.4.2 Алгоритм тагрегетирования

Описанная выше задача сопоставления пользователю набора приложений, на основе знаний о конкретном пользователе, о всех остальных пользователях и всех приложениях носит название "Задача коллаборативной фильтрации"[ссылка].

Определение 1 *Коллаборативная фильтрация, совместная фильтрация (англ. collaborative filtering) — это один из методов построения прогнозов (рекомендаций) в рекомендательных системах, использующий известные предпочтения (оценки) группы пользователей для прогнозирования неизвестных предпочтений другого пользователя. Его основное допущение состоит в следующем: те, кто одинаково оценивали какие-либо предметы в прошлом, склонны давать похожие оценки другим предметам и в будущем. [TODO: заменить на определение из Сегарана]*

Для решения данной задачи разработано множество методов, которые можно разделить на следующие группы:

- item-based
- user-based
- модельные
- гибридные

[TODO надо кратко сказать про каждый? вообще, это давно разжевано и не совсем по теме работы]

Одним из наиболее успешных (с точки зрения качества рекомендаций) методов является Factorization Machines (FM) [ссылка на результаты KDD]. Большинство широко-известных методов коллаборативной фильтрации являются частными случаями FM (примеры сведения будут показаны далее) [ссылка на Рендла], что позволяет легко адаптировать приведенные в данной работе рассуждения, к большинству основных алгоритмов коллаборативной фильтрации.

0.4.3 Формальное представление знаний

Прежде чем углубиться в рассмотрение алгоритма FM необходимо научиться описывать имеющиеся знания про объекты предметной области с помощью формальных математических объектов.

Определение 2 *Фактор* — $f : \boldsymbol{A} \times \boldsymbol{U} :\to \boldsymbol{B} \subset \boldsymbol{R}^{n_f}$. *Отображение сопоставляющее паре пользователь-приложение вещественный вектор размерности* n_f.

Заметим, что множества **A** и **U** содержат фиктивные элементы. Это необходимо, что бы описывать факторы, которые зависят только от пользователя или только от приложения. Примеры факторов:

- $Rating : \mathbf{A} \to (0, 5)$
- $UserId : \mathbf{U} - > \{0, 1\}^{|\mathbf{U}|}$ — отображение сопоставляющее каждому пользователю уникальный вектор, состоящий из $|\mathbf{U}| - 1$ нуля и одной еденицы
- $TimeDistribution : \mathbf{A} \times \mathbf{U} \to \mathbf{R}^{n_{TimeDistribution}}$ — отображение, сопоставляющее каждой паре пользователь-приложение вектор $t = (t_1, \dots t_{n_{TimeDistribution}})$, где t_i - доля запусков данного приложения конкретным пользователем в i-й промежуток времени, $i = 1 \dots n_{TimeDistribution}$.

Пусть:

- *Concat* — операция конкатенации произвольного числа векторов
- $\{\mathbf{Factors}_i\}_{i=1}^{FactorsCount}$ — индексированное множество факторов (для удобства обозначим за $f_k 1 \leq k \leq FactorsCount$ - значение фактора $\mathbf{Factors}_k$ для фиксированной пары пользователь - приложение),

тогда всю информацию про пару пользователь-приложение можно представить в виде: $x = Concat(f_1, \dots, f_{FactorsCount})$. Будем называть x - *вектором факторов*.

Замечание: предложенный способ является достаточно универсальным для описания произвольных объектов в пространстве $\mathbf{R}^n$, однако, вектора факторов могут получаться сильно разреженными. Например, для рассматриваемой задачи вектор факторов имеет длину порядка $\sim 0.5 * 10^8$, при этом кол-во не нулевых элементов $\sim 10^3$.

0.5 Введение в Factorization Machines

[TODO РАССТАВИТЬ ССЫЛКИ НА ИСТОЧНИКИ] На протяжении долгого времени, одним из самых популярных алгоритмов машинного обучения был метод опорных векторов (SVM) , однако, SVM не нашел широкого применения в задачах коллаборативной фильтрации, т.к. не способен построить надежную разделяющую поверхность в нелинейных пространствах, опираясь на сильно разреженные данные. В задачах данного типа хорошо зарекомендовали себя методы, основывающиеся на идеи матричной/тензорной факторизации, например, PARAFAC [можно вставить ссылку на Хершмана]. С другой стороны, модели на основе матричной факторизации сильно ограничивают тип входных данных и не могут быть применены к стандартным векторам факторов. Т.е. данные модели не позволяют использовать все знания про объекты предметной области, которыми мы обладаем. Большинство популярных методов (MF, SVD, SVD++, PITF) позволяют использовать лишь знания про то, как и какие товары оценил пользователь.

Модель FM является SVM с полиномиальным ядром (см. kernel trick) с одним важным отличием: коэффициенты представляются в факторизованном виде (т.е. в виде пары векторов v, u, таких что $< u, v >= a \in \mathbf{R}$).

В общем случае, алгоритм доолжен апроксимировать некоторую целевую функцию, опираясь на ее значения на некотором обучающем множестве. Обозначим эту функцию как $y(x)$, где x - вектор факторов. $y(x)$ может быть оценкой, которую поставил пользователь приложению в магазине (задача регрессии) или вероятностью того, что приложение (не) понравится пользователю (задача классификации). [ПРО ВЕРОЯТНОСТЬ (НЕ)ПОНРАВИТСЯ - НЕ СОВСЕМ ТАК, ОБЫЧНО ВОССТАНАВЛИВАЮТ ПАРАМЕТР ЛОГИТА, ПОТОМ ЕГО ПОДСТАВЛЯЮТ В ЛОГИСТИЧЕСКУЮ РЕГРЕССИЮ И ПОЛУЧАЮТ НЕПОСРЕДСТВЕННО ВЕРОЯТНОСТЬ].

Модельное уравнение для FM 2-го порядка (модель более высоких порядков не имеют практической ценности):

$$\tilde{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j, \tag{1}$$

где $w_0 \in \mathbf{R}, w \in \mathbf{R}^n, V \in \mathbf{R}^{n \times k}$ - парметры модели, которые необходимо подобрать.

- w_0 — общее смещение модели (вероятность, что случайное приложение понравится случайному пользователю)
- w_i — вес i-й компоненты вектора факторов x в данной модели
- v_i — вещественный вектор размера k, описывающий i-й компонент вектора факторов. k - входной параметр алгоритма, описывающий глубину факторизации
- $\tilde{w}_{i,j} =< v_i, v_j >$ — вес взаимодействия i-й и j-й компоненты вектора факторов. Основной особенностью FM является то, что вместо прямого использования $\tilde{w}_{i,j}$ как параметра модели, используется их факторизованное представление. Именно это позволяет методу работать с сильно разреженными данными.

Подбор параметров модели (w_0, w, V) осуществляется с помощью различных алгоритмов стохастической оптимизации, например, MCMC, SGD, ASGD, ALS.

[ДОКАЗАТЬ, ЧТО ВЫЧИСЛЕНИЕ $\tilde{y}(x)$ МОЖНО СДЕЛАТЬ ЗА ЛИНЕЙНОЕ ВРЕМЯ. ПРИВЕСТИ ПРИМЕРЫ СВЕДЕНИЯ FM К БОЛЕЕ ПРОСТЫМ МОДЕЛЯМ. НАПИСАТЬ ВСЕ-ТАКИ ПРО ЦЕЛЕВУЮ ФУНКЦИЮ (ДЛЯ НАШЕЙ ЛКАССИФИКАЦИИ - ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ)].

0.6 Сравнение двух алгоритмов интерпретации (АИ)

[TODO ЧЕСТНО ОПИСАТЬ КАК СРАВНИВАТЬ, КОГДА МОЖНО СЧИТАТЬ ЭКСПЕРИМЕНТ ЗАВЕРШЕННЫМ (ПРО ПЕРЕКРЫТИЕ ДОВЕРИТЕЛЬНЫХ ИНТЕРВАЛОВ), ВПИЛИТЬ АКСЕССОРОВ]

0.7 Базовый АИ

Начнем наши рассуждения с приведения наивного АИ, с которым будем сравнивать все последующие вариации. Вектор факторов, используемый для предсказания значения вероятности покупки является конкатенацией значений факторов. Сопоставим каждому фактору $\mathbf{Factors}_k 1 \leq k \leq FactorsCount$ некоторый сопроводительный текст, который, по нашему мнению, объясняет семантику фактора. Например, фактору *Rating* можно сопоставить текст "У этого приложения отличный рейтинг".

Стоит отметить тот факт, что способ сопоставление факторов конкретным текстам, безусловно, сильно влияет на результат. Однако, данное отображение будет зафиксировано для большинства предложенных методов, что позволит сравнивать сами методы, без учета влияния формулировок текстов.

Теперь попытаемся выяснить какой фактор вносит наибольший положительный вклад в результирующую вероятность и выберем сопроводительный текст, соответствующий данному фактору.

Data: $\{f_i\}_{i=1}^{FactorsCount}$ - мн-во значений факторов
Result: i - номер фактора, вносящего наибольший положительный вклад

$maxContribution = 0$;
$idx = +\infty$
for $i = 1, \ldots, FactorsCount$ **do**
    contirbution $= \tilde{y}(Concat(0, \ldots, 0, f_i, 0, \ldots, 0, \ldots 0))$
    if $contirbution > maxContribution$ **then**
        $idx = i$
        $maxContribution = contribution$
    else
        NOP
    end
end
return idx

Algorithm 1: Псевдокод наивного определения вклада фактора.

Заметим, что

$$\tilde{y}(Concat(f_1, \ldots, f_n)) - \tilde{y}(Concat(f_1, \ldots, f_{i-1}, 0, \ldots, 0, f_{i+i}, \ldots, f_n)) \equiv \tilde{y}(Concat(0, \ldots, 0, f_i, 0, \ldots, 0, \ldots 0)), \quad (2)$$

однако, второй вариант является более эффективным с точки зрения числа операций (при условии эффективной реализации вычислений с разреженными векторами).

Предложенный пособ обладает очевидным недостатком, он полагается на абсолютные значения изменения предсказанной вероятности. Но некоторые факторы всегда будут вносить больший абсолютный вклад, чем другие. Например, это может быть вызвано разной нормировкой различных факторов или тем, что одни факторы содержат больше информации, чем другие.

0.8 Нормальный АИ

Учитывая замечания [TODO МОЖНО ПРИВЕСТИ ПРИМЕРЫ РАБОТЫ АЛГОРИТМА, ЧТО БЫ ПОНЯТЬ, ПОЧЕМУ ОН НЕ ОК???], изложенные ранее, постараемся модифицировать базовый АИ.

Введем гипотезу, что значение фактора в данной рекомендации зависит не от абсолютного значения вклада, данного фактора в вероятность (как в базовом подходе), а от того, насколько больше этот вклад в конкретном случае, чем в среднем. Т.е. предлагается сравнивать не абсолютные значения прироста предсказанной вероятности, а относительные приросты для каждого фактора. Такой подход позволяет нивилировать различные нормировки факторов, а так же их априорную информативность.

Введем обозначение: $c_{f_i}(x) = \tilde{y}(Concat(f_1, \ldots, f_n)) - \tilde{y}(Concat(f_1, \ldots, f_{i-1}, 0, \ldots, 0, f_{i+i}, \ldots, f_n))$. В качестве первого приближения, предположим, что значения c_{f_i} распределены нормально $\sim \mathcal{N}(\mu_{f_i}, \sigma^2_{f_i})$. Для оценки параметров $\mu_{f_i}, \sigma^2_{f_i}$ воспользуемся методом максимального правдоподобия, согласно которому:

$$\begin{cases} \tilde{\mu}_{f_i} = \overline{X} & \text{— выборочное среднее} \\ \tilde{\sigma}^2_{f_i} = \overline{S^2} & \text{— выборочная дисперсия} \end{cases}$$

Выбрку для оценки парметров можно получить с помощью простого вероятностного сэмплирования мн-ва $\mathbf{U} \times \mathbf{A}$. [TODO: А МОЖНО ЧТО-НИБУДЬ ПОИНТЕРЕСНЕЕ ТИПА ПРОПОРЦИОНАЛЬНОГО ИЛИ СТРАТИФИЦИРОВАННОГО (НО ЗАЧЕМ?), ОПИСАТЬ ИХ ПОДРОБНЕЕ? ДАЕТ ЛИ ЭТО ГАРАНТИЮ НЕСМЕЩЕННОСТИ ОЦЕНОК, ЕСЛИ ПРИНЯТЬ ГИПОТЕЗУ О ТОМ, ЧТО $\mathbf{U} \times \mathbf{A}$ РЕПРЕЗЕТНАТИВНА?]

Data: $\{f_i\}_{i=1}^{FactorsCount}$ - мн-во значений факторов, $\mathbf{X}$ - репрезентативная выборка векторов факторов (фактически, выборка пар пользователь-приложение)
Result: i - номер фактора, вносящего наибольший положительный вклад

$maxContribution = 0$;
$idx = +\infty$
for $i = 1, \ldots, FactorsCount$ **do**
  // оцениваем параметры распределения привосста вероятности от i-го фактора
  $\tilde{\mu}_{f_i}, \tilde{\sigma}^2_{f_i} = MaxLikelihood(\mathbf{X}, i)$
  contirbution = $\frac{\tilde{y}(Concat(0,\ldots,0,f_i,0,\ldots,0,\ldots0)) - \tilde{\mu}_{f_i}}{\tilde{\sigma}_{f_i}}$
  if $contirbution > maxContribution$ **then**
    $idx = i$
    $maxContribution = contribution$
  else
    NOP
  end
end
return idx

Algorithm 2: Псевдокод определения фактора с наибольшим относительным вкладом.

0.9 План ны ближайшие дни

- заменить пару TODO нормальным текстом
- показать, что гипотеза о нормальности распределения в предыдущем пункте не принимается
- описать 3 способа восстановления плотности по эмпирическим данным (ядерные оценки, параметрический подход, смесь)

- написать модификацию алгоритма с применением EM [?], посмотреть, что будет. Попытаться восстановить как смесь гауссианов