Анализ Клиентских Сред

Оценивание сходства пользователей и ресурсов путем выявления скрытых тематических профилей.

Постановка задачи

Исходными данными являются протоколы действий пользователей. Каждая запись протокола описывает событие «пользователь *u* выбрал ресурс *r*». В зависимости от конкретной задачи (предметной области) запись может содержать следующую информацию: идентификатор пользователя, название ресурса, тип, время начала и продолжительность действия и т.д. Например, действием может быть посещение страницы, выбор товара или услуги в интернет-магазине, проставление рейтинга просмотренного фильма и т. д.

Введем некоторые понятия, необходимые для формальной постановки задачи.

Пусть есть R — множество ресурсов,

U — множество пользователей.

Задан протокол пользования $D = (u_i, r_i)$, $i = 1,...,l$. То есть множество событий типа пользователь $u \in U$ использовал ресурс $r \in R$, l— число записей в протоколе.

По нему строится матрица пользования $A_{|U| \times |R|}$. В зависимости от конкретной задачи A_{ur} могут нести различную информацию о пользовательском поведении. Это может быть бинарная информация о посещении или не посещении заданного ресурса данным пользователем, частота (или число) пользований ресурса r пользователем u, стоимость или рейтинг, проставленный пользователем *u* для ресурса *r* и т.д.

Матрица $A_{|U| \times |R|}$ чаще всего бывает сильно разреженной (далеко не все пользователи используют все ресурсы и, наоборот, далеко не всеми ресурсами пользуются все пользователи). Поэтому множество посещений нам часто удобнее хранить не в виде громоздкой матрицы, а в виде набора непустых ее элементов: $D_A = \{(u,r) \mid A_{ur} > 0\}$. D_u и D_r — множества непустых элементов в матрице посещений при фиксированном *u* и при фиксированном *r* соответственно.

Задача заключается в том, чтобы по имеющимся данным построить функции сходства (метрики) на множестве пользователей $\rho_U : U \times U \to R_+$ и на множестве ресурсов $\rho_R : R \times R \to R_+$ таким образом, чтобы близкими по метрике ρ_U были пользователи, которые пользуются схожими ресурсами, и близкими по метрике ρ_R были ресурсы, которыми пользуются схожие клиенты.

В данной работе рассматривается подход, при котором сначала по исходным данным вычисляются векторные описания ресурсов и пользователей, называемые в работе профилями. Затем функции сходства определяются как евклидовы метрики над профилями. Причем компоненты профилей интерпретируются как тематики, которыми могут интересоваться пользователи и которые могут удовлетворять (в той или иной степени) ресурсы.

Пусть T — множество тем в профиле.

Тематический профиль ресурса *r* и пользователя *u* запишем следующим образом:

$$P_r = (p_{1r}, p_{2r}, ..., p_{|T|r})$$

$$P_u = (p_{1u}, p_{2u}, ..., p_{|T|u}).$$

Подчеркнем, что природа профилей пользователей и ресурсов одинакова, поэтому, зная профили для всех ресурсов и пользователей, мы можем оценить расстояния (например, как среднеквадратичное отклонение) как от ресурса до ресурса и от пользователя до пользователя, так и от пользователя до ресурса:

$$\rho_R(r, r') = \rho(P_r, P_{r'}) = \sqrt{\sum_{t=1}^{|T|} (p_{1r} - p_{1r'})^2}.$$

Аналогично

$$\rho_U(u, u') = \rho(P_u, P_{u'}) = \sqrt{\sum_{t=1}^{|T|} (p_{1u} - p_{1u'})^2}$$

и

$$\rho(r, u) = \rho(P_r, P_u) = \sqrt{\sum_{t=1}^{|T|} (p_{1r} - p_{1u})^2}.$$

Вероятностная постановка задачи

Допустим, что у каждого пользователя $u \in U$ имеется некоторое множество интересов или потребностей, которые мы будем называть темами. Множество всех возможных тем обозначим через *T*. Допустим, пользователь *u* имеет интерес $t \in T$ с вероятностью $p(t \mid u)$. В свою очередь, каждый ресурс соответствуем некоторому множеству тем. Допустим, ресурс *r* удовлетворяет теме *t* с вероятностью *p(t|r)*.

Профиль пользователя *u* определим как вектор значений вероятностей $p(t \mid u)$, $t = 1,...,|T|$, причем $\sum_{t \in T} p(t \mid u) = 1$.

Аналогично, профиль ресурса *r* — это вектор вероятностей $p(t \mid r)$, $t = 1,...,|T|$, причем $\sum_{t \in T} p(t \mid r) = 1$.

Обозначим через *p(u,r)* — вероятность выбора ресурса *r* пользователем *u.*

Метод восстановления профилей, предлагаемый в данной работе опирается на EM-алгоритм разделения смеси распределений.

Коллаборативная фильтрация на основе EM-алгоритма

Распишем вероятность выбора ресурса *r* пользователем *u*:

$$p(u, r) = \sum_t p(u)\, p(t \mid u) q(r \mid t, u),$$

где $p(u) \equiv p_u$ — априорная вероятность того, что выбор будет сделан пользователем *u,* $\sum_{u \in U} p_u = 1$*;*

$p(t \mid u) = p_{tu}, \sum_{t \in T} p_{tu} = 1$ для всех $u \in U$ — вероятность того, что пользователь u в данный момент интересуется темой t;

$q(r \mid t, u) = q(r \mid t)$ — апостериорная вероятность того, что будет выбран ресурс r при условии, что выбор делает пользователь u, интересующийся темой t.

Таким образом, вероятность $p(u, r)$ будем записывать как сумму произведений этих трех вероятностей по всем темам.

Апостериорную вероятность $q(r \mid t)$ найдем по формуле Байеса:

$$q(r \mid t) = \frac{q_{tr} q_r}{\sum_{s \in R} q_{ts} q_s}.$$

Подставим это выражение в формулу для $p(u, r)$:

$$p(u \mid r) = \sum_t p_u q_r p_{tu} \frac{q_{tr}}{\sum_{s \in R} q_{ts} q_s} \tag{4}$$

Приведенные выше формулы аналогичным образом можно записать относительно ресурсов*:*

$$p(u, r) = \sum_{t \in T} q(r) q(t \mid r) p(u \mid t, r),$$

где $q(r) = q_r$ — значения априорной вероятности выбора ресурса *r;*

$q(t \mid r) = q_{tr}$ — вероятность того, что ресурсу r соответствует тема t;

$p(u \mid t, r) = p(u \mid t)$ — апостериорная вероятность того, что пользователь u интересуется темой *t,* зайдя на ресурс *r*. Здесь и в предыдущем случае мы опираемся на гипотезу независимости посещения пользователем различных ресурсов и считаем, что $p(u \mid t, r)$ не зависит от *r*.

Таким образом, мы можем записать формулу для вероятности $p(u \mid r)$ через профили ресурсов:

$$p(u, r) = \sum_{t \in T} p_u q_r q_{tr} \frac{p_{tu}}{\sum_{s \in R} p_{ts} p_s} \tag{5}$$

Воспользуемся принципом максимума правдоподобия для выборки посещений $D = (u_i, r_i)_{i=1}^{l}$:

$$\ln \prod_{i=1}^{l} p(u_i, r_i) \to \max_{p_{tu}, q_{tr}} \tag{6}$$

Основная идея алгоритма восстановления профилей p_{tu} и q_{tr} по выборке D заключается в последовательном чередовании двух действий:

1) оптимизировать p_{tu} при фиксированном q_{tr},
2) оптимизировать q_{tr} при фиксированном p_{tu}

и так далее в итерациях пока не будет достигнута сходимость.

Рассмотрим задачу максимума правдоподобия (МП) для формулы (4).

Запишем Лагранжиан:

$$L(p_{tu}) = \sum_{(u,r)\in D} \ln p_u \sum_{t\in T} p_{tu} \frac{q_{tr} q_r}{\sum_s q_{ts} q_s} - \sum_u \lambda_u \left(\sum_{t\in T} p_{tu} - 1\right) \to \max_{p_{tu}}$$

При ограничениях:

$\sum_t p_{tu} = 1, \forall u \in U$;

$p_{tu} \geq 0, \forall t \in T, \forall u \in U$.

Продифференцируем по p_{tu} и прировняем к нулю:

$\frac{\partial L}{\partial p_{tu}} = \sum_{r\in D_u} \frac{1}{p_{tu}} \frac{p_{tu} q(r \mid t)}{\sum_{s\in T} p_{su} q(r \mid s)} - \lambda_u = 0$. Здесь мы умножили и разделили на p_{tu} .

Введем обозначение для так называемых скрытых компонент:

$$H_{tr}(u) = \frac{p_{tu} q(r \mid t)}{\sum_{s\in T} p_{su} q(r \mid s)}, \qquad (7)$$

и заметим, что это выражение есть ни что иное, как формула Байеса. Можно дать вероятностную интерпретацию данных компонент следующим образом: вероятность того, что пользователь *u*, зайдя на ресурс *r,* интересовался темой *t*.

Тогда

$\sum_{t\in T} H_{tr}(u) = 1$.

Далее разнесем два члена уравнения в разные части равенства, помножим обе части на p_{tu} и просуммируем по t:

$$\sum_{t\in T} \sum_{r\in D_u} \frac{p_{tu} q(r \mid t)}{\sum_s p_{su} q(r \mid s)} = \sum_{t\in T} \lambda_u p_{tu} .$$

Учитывая, что $\sum_{t\in T} p_{tu} = 1$ и $\mid D_u \mid = p_u \mid R \mid$, получаем:

$\sum_{r\in D_u} 1 = \lambda_u$, следовательно $\lambda_u = \mid R \mid p_u$.

Таким образом, получаем

$$p_{tu} = \frac{\sum_{r\in D_u} H_{tr}(u)}{\sum_{r\in D_u} 1} .$$

Идея заключается в том, чтобы последовательно вычислять скрытые компоненты $H_{tr}(u)$ при заданном p_{tu} по формуле (7), а затем вычислять p_{tu} при заданном $H_{tr}(u)$ и так

далее в итерациях, пока не будет достигнута сходимость (значения будут меняться мало).

Найдем начальное приближение для $H_{tr}(u)$, для этого выпишем еще раз выражение для скрытых компонент:

$$H_{tr}(u) = \frac{p_{tu}\dfrac{q_{tr}q_r}{\sum\limits_{x\in T} q_{tx}q_x}}{\sum\limits_{s\in T} p_{su}\dfrac{q_{sr}q_r}{\sum\limits_{x\in T} q_{sx}q_x}}.$$

Пусть начальное приближение для p_{tu} – равномерное $p_{tu} = \frac{1}{|T|}$, тогда:

$$H_{tr}(u) = \frac{q(r\,|\,t)}{\sum\limits_{s\in T} q(r\,|\,s)}, \text{ где } q(r\,|\,t) = \frac{q_{tr}q_r}{\sum\limits_{r'\in R} q_{tr'}q_{r'}}.$$

Показанным выше способом мы можем оптимизировать p_{tu} при фиксированном q_{tr}. Совершенно аналогичным способом можно решить задачу максимума правдоподобия (6) для формулы (5). Тогда мы сможем решить обратную задачу: оптимизировать q_{tr} при фиксированном p_{tu}.

Теперь можно выписать алгоритм.

1.1.1 Описание алгоритма

Вход:

$|U|$ — число пользователей;

$|R|$ — число ресурсов;

$|T|$ — число тем в профиле;

$D = (u_i r_i)_{i=1}^{l}$ — выборка посещений (из нее получается матрица посещений $A_{|U|\times|R|}$;

Выход:

p_{tu} — профили пользователей;

q_{tr} — профили ресурсов;

Алгоритм:

1: построить списки D_u и D_r непустых значений в матрице A.

2: задать случайное начальное приближение для q_{tr} такое, что $\sum\limits_{t=1}^{|T|} q_{tr} = 1, \forall r$.

3: вычислить $q_r = \frac{\sum 1}{|R|}$ и $p_u = \frac{\sum 1}{|U|}$ — средние значения для каждого пользователя и для каждого ресурса.

4: **повторять**

I. Оптимизируем p_{tu} *при фиксированном* q_{tr}*:*

5: **для всех** $t \in T$

6: вычислить суммы $S_t = \sum_{r' \in R} q_{tr'} q_{r'}$

7: **для всех** $r \in R$

8: вычислить $q(r \mid t) = \frac{q_{tr} q_r}{S_t}$

9: вычислить $H_{tr}(u) = \frac{q(r \mid t)}{\sum_{s \in T} q(r \mid s)}$ (не зависит от u)

10: **повторять**

11: **для всех** $u \in U$, $t \in T$

12: вычислить $p_{tu} = \frac{\sum_{r \in D_u} H_{tr}(u)}{\sum_{r \in D_u} 1}$

13: **для всех** $u \in U$, $r \in R$

14: вычислить суммы $S_{ur} = \sum_{s \in T} p_{su} q(r \mid s)$

15: **для всех** $t \in T$

16: вычислить $H_{tr}(u) = \frac{p_{tu} q(r \mid t)}{S_{ur}}$

17: **пока** не сойдется

II. Оптимизируем q_{tr} *при фиксированном* p_{tu}*:*

18: **для всех** $t \in T$

19: вычислить суммы $S_t = \sum_{u' \in U} p_{tu'} p_{u'}$

20: **для всех** $u \in U$

21: вычислить $p(u \mid t) = \frac{p_{tu} p_u}{S_t}$

22: вычислить $H^*_{tu}(r) = \frac{p(u \mid t)}{\sum_{s \in T} p(u \mid s)}$

23: **повторять**

24: **для всех** $r \in R$, $t \in T$

23: вычислить $q_{tr} = \dfrac{\sum_{u \in D_r} H^*_{tu}(r)}{\sum_{u \in D_r} 1}$

25: **для всех** $u \in U$, $r \in R$

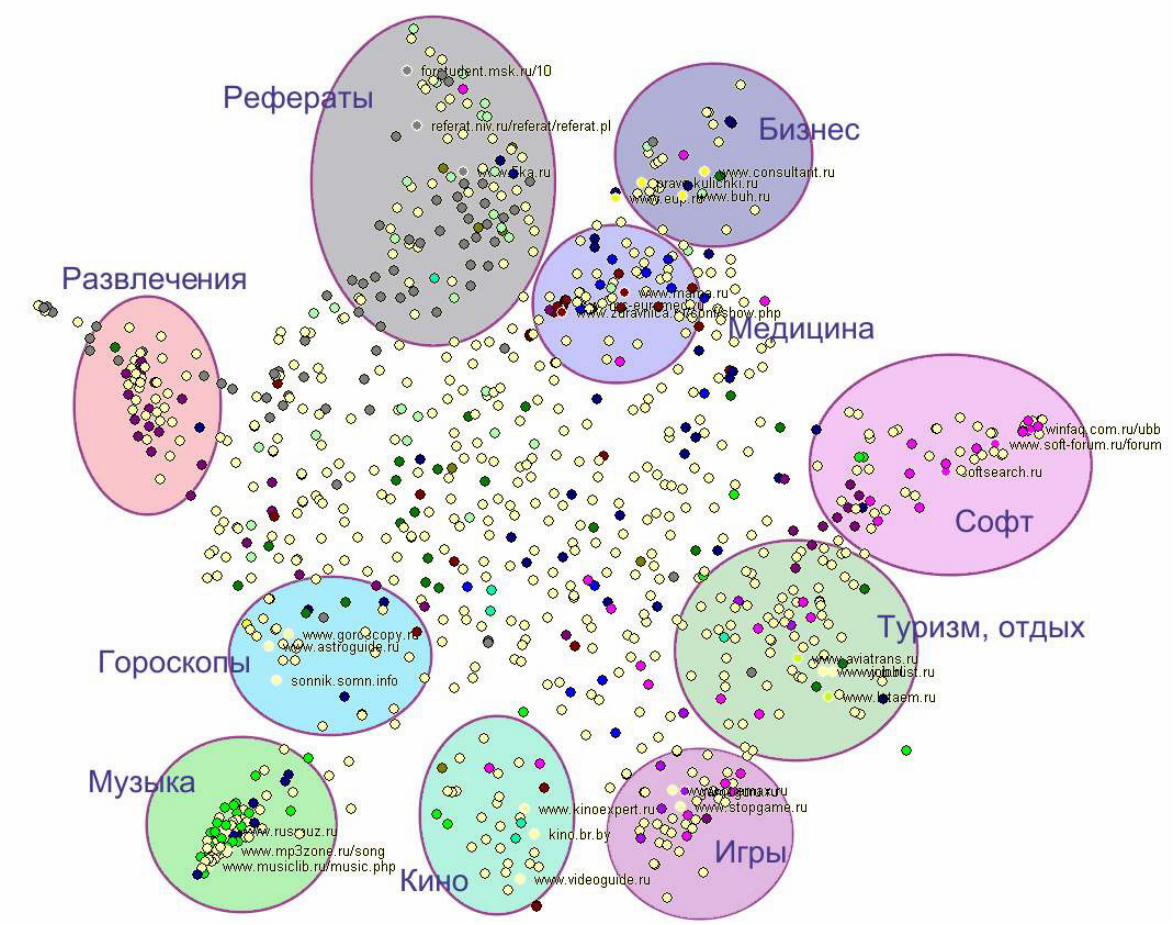
26: вычислить суммы $S_{ru} = \sum_{s \in T} q_{sr}\, p(u \mid s)$

27: **для всех** $t \in T$

28: вычислить $H^*_{tu}(r) = \dfrac{q_{tr}\, p(u \mid t)}{S_{ru}}$

29: **пока** не сойдется

30: **пока** не сойдется



Задачи

Оптимизировать алгоритм

- разреженное хранение матриц и скрытых компонент H
- добиться высокой скорости обработки данных
- функционал качества считать по профилям, а не по метрике
- оптимизировать параметры алгоритма на реальных и на модельных данных: число внутренних, внешних итераций, количество тем

Исследовать расходимость алгоритма

- профили известны, куда уйдет алгоритм?
- какого размера должны быть выборки и профили, чтобы не уходил?
- нужно ли накладывать ограничения типа некоррелированности профилей?

Очевидно, что если тематики пересекаются по сайтам, то посещения будут размытыми, но если есть жесткая кластеризация: некоторые группы пользователей посещают только некоторые группы сайтов, то в профилях должны получаться чисто нулевые компоненты. То есть если ко-кластерная структура существует в данных, то алгоритм ее выявит.

Доказать теорему:

Пусть

$U = U_1 \cup U_2 \cup ... \cup U_n$,

$R = R_1 \cup R_2 \cup ... \cup R_m$.

Пользователи из группы U_i посещают только ресурсы из группы R_j.

Тогда в профиле ненулевыми могут оказаться только компоненты, соответствующие «своей тематике».

Робастные алгоритмы

Игнорируют малые пересечения между тематиками и все равно выдают нулевые компоненты профилей.

Рассмотреть эвристику: «занулять незначительные компоненты профилей на каждой итерации».

Выбирать порог значимости.

Разреженность матрицы расстояний

Найти окрестность ближайших соседей для каждого ресурса. Все дальнейшие анализы проводить только для этой окрестности.

Если профили не пересекаются, то эти ресурсы считать «бесконечно далекими» и эту пару вообще нигде не учитывать.

Исследовать теоретически сходимость алгоритма

Построить теоретические оценки для количества внутренних и внешних итераций.

Иерархические профили

Можно будет существенно оптимизировать работу алгоритма, если учитывать иерархию тем в профиле.

Написать статьи

Предполагается как минимум 2 статьи. Первая с более подробным описанием практических результатов, а вторая, более глубокая, с уклоном на теоретические результаты.