

Методы коллаборативной фильтрации и их применения

К. В. Воронцов
vokov@forecsys.ru
<http://www.ccas.ru/voron>

Вычислительный Центр им. А. А. Дородницына РАН;
ЗАО «Форексис»

ВШЭ, семинар Б. Г. Миркина, 10 ноября 2008

Содержание

- 1 Постановка задачи и приложения**
 - Постановка задачи
 - Примеры приложений
- 2 Обзор методов**
 - Модели, основанные на хранении данных
 - Латентные модели
- 3 Вероятностные латентные семантические модели**
 - Постановка задачи и EM-алгоритм
 - Результаты экспериментов

Определения и обозначения

U — множество субъектов (клиентов, пользователей: users);

R — множество объектов (ресурсов, товаров, предметов: items);

Y — пространство описаний транзакций;

Сырые исходные данные:

$D = (u_i, r_i, y_i)_{i=1}^m \in U \times R \times Y$ — протокол транзакций;

Агрегированные данные:

$F = \|f_{ur}\|$ — матрица кросс-табуляции размера $|U| \times |R|$,

где $f_{ur} = \text{aggr}\{(u_i, r_i, y_i) \in D \mid u_i = u, r_i = r\}$

Задачи:

- прогнозирование незаполненных ячеек f_{ur} ;
- оценивание сходства: $\rho(u, u')$, $\rho(r, r')$, $\rho(u, r)$;
- выявление скрытых интересов $p(t|u)$, $q(t|r)$ относительно заданного либо неизвестного набора тем $t = 1, \dots, T$.

Пример 1. Рекомендующая система на основе бинарных данных

U — пользователи Интернет;

R — ресурсы (сайты, документы, новости, и т.п.);

$f_{ur} = [\text{пользователь } u \text{ посетил ресурс } r];$

Основная гипотеза Web Usage Mining:

- Действия (посещения) пользователя характеризуют его интересы, вкусы, привычки, возможности.

Задачи персонализации:

- выдать оценку ресурса r для пользователя u ;
- выдать пользователю u ранжированный список рекомендуемых ресурсов;
- сгенерировать для ресурса r список близких ресурсов.

Пример 2. Рекомендующая система на основе бинарных данных

U — клиенты интернет-магазина (amazon.com и др.);

R — товары (книги, видео, музыка, и т.п.);

f_{ur} = [клиент u купил товар r];

Задачи персонализации предложений:

- выдать оценку товара r для клиента u ;
- выдать клиенту u список рекомендуемых товаров;
- предложить скидку на совместную покупку (cross-selling);
- информировать клиента о новом товаре (up-selling);
- сегментировать клиентскую базу;
выделить целевые аудитории по интересам.

Пример 3. Рекомендующая система на основе рейтингов

U — клиенты интернет-магазина (netflix.com и др.);

R — товары (книги, видео, музыка, и т.п.);

f_{ur} = рейтинг, который клиент u выставил товару r ;

Задачи персонализации предложений — те же.

Пример: конкурс Netflix [www.netflixprize.com]

- с октября 2006 до сих пор; **главный приз — \$10⁶**;
- $|U| = 0.48 \cdot 10^6$; $|R| = 1.7 \cdot 10^4$;
- 10^8 рейтингов $\{1, 2, 3, 4, 5\}$;
- точность прогнозов оценивается по тестовой выборке D' :

$$\text{RMSE}^2 = \sum_{(u,r) \in D'} (f_{ur} - \hat{f}_{ur})^2;$$

- требуется: уменьшить RMSE с 0.9514 до 0.8563 (на 10%)
текущий рекорд от 30.09.2008: 0.8616 (9.44%).

Пример 4. Анализ текстов

U — текстовые документы (статьи, новости, и т.п.);

R — ключевые слова или выражения;

f_{ur} = частота встречаемости слова r в тексте u .

Задачи анализа текстов:

- кластеризация текстов: сгруппировать тексты по тематике;
- определить тематику нового текста (например, новости);
- найти тексты той же тематики, что данный текст;
ранжировать найденные тексты по сходству;
- построить иерархический каталог текстов;
описать каждый раздел набором ключевых слов.

Пример 5. Социальные сети, форумы, блоги

U — пользователи;

R — текстовые документы (форумы, блоги);

K — ключи (ключевые слова или выражения);

f_{ur} = [пользователь u участвует в r];

g_{rk} = частота встречаемости ключа k в тексте r ;

h_{uv} = [пользователю u интересен пользователь v].

Некоторые задачи анализа социальной сети:

- рекомендовать пользователю интересные ему блоги, найти единомышленников (like-minded people);
- охарактеризовать интересы пользователя ключами;
- найти все блоги по данным или похожим ключам;
- найти все блоги, похожие на данный;
- построить иерархический тематический каталог блогов.

Два основных подхода

1 Модели, основанные на хранении исходных данных (Memory-Based Collaborative Filtering)

- хранение всей исходной матрицы данных F ;
- сходство клиентов — это корреляция строк матрицы F ;
- сходство объектов — это корреляция столбцов матрицы F .

2 Латентные модели (Latent Models for Collaborative Filtering)

- оценивание профилей клиентов и объектов
(*профиль — это вектор скрытых характеристик*);
- хранение профилей вместо хранения F ;
- сходство клиентов и объектов — это сходство их профилей.

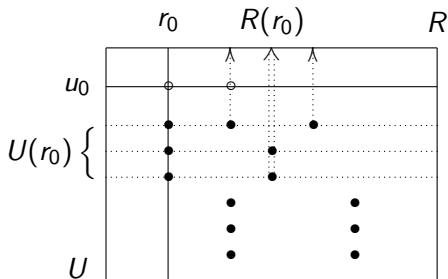
Подборки статей по коллаборативной фильтрации:

jamesthornton.com/cf

www.adastral.ucl.ac.uk/~junwang/CollaborativeFiltering.html

Тривиальная рекомендующая система

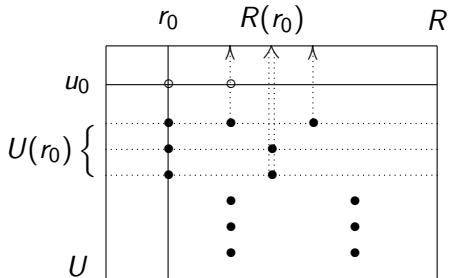
«клиенты, купившие r_0 ,
также покупали $R(r_0)$ »
[Amazon.com]



- 1 $U(r_0) := \{u \in U \mid f_{ur_0} \neq \emptyset, u \neq u_0\};$
- 2 $R(r_0) := \left\{r \in R \mid B(r) = \frac{|U(r_0) \cap U(r)|}{|U(r_0) \cup U(r)|} > 0\right\},$
где $B(r)$ — одна из возможных мер близости r к r_0 ;
- 3 отсортировать $r \in R(r_0)$ по убыванию $B(r)$, взять top N .

Тривиальная рекомендующая система

«клиенты, купившие r_0 ,
также покупали $R(r_0)$ »
[Amazon.com]

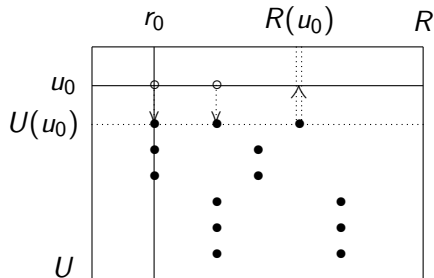


Недостатки:

- рекомендации тривиальны (предлагается всё наиболее популярное);
- не учитываются интересы конкретного пользователя u_0 ;
- не учитывается степень сходства ресурсов r и r_0 ;
- проблема «холодного старта»;
- надо хранить всю матрицу F .

От клиента (user-based CF)

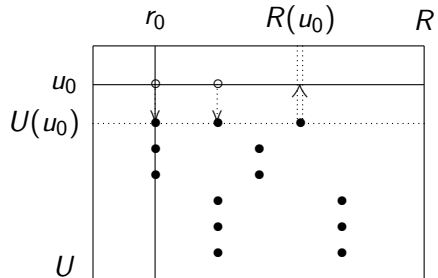
«клиенты, похожие на u_0 ,
также покупали $R(u_0)$ »



- 1 $U(u_0) := \{u \in U \mid \text{corr}(u_0, u) > \alpha\};$
- 2 $R(u_0) := \left\{r \in R \mid B(r) = \frac{|U(r_0) \cap U(r)|}{|U(u_0) \cup U(r)|} > 0\right\};$
- 3 отсортировать $r \in R(u_0)$ по убыванию $B(r)$, взять top N ;

От клиента (user-based CF)

«клиенты, похожие на u_0 ,
также покупали $R(u_0)$ »

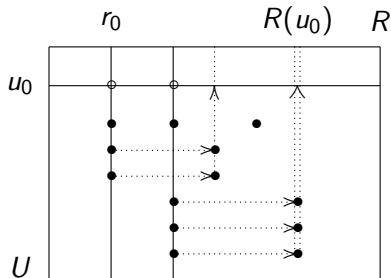


Недостатки:

- рекомендации тривиальны;
- не учитываются интересы конкретного пользователя u_0 ;
- не учитывается степень сходства ресурсов r и r_0 ;
- проблема «холодного старта»;
- надо хранить всю матрицу F ;
- нечего рекомендовать нетипичным/новым пользователям.

От объекта (item-based CF)

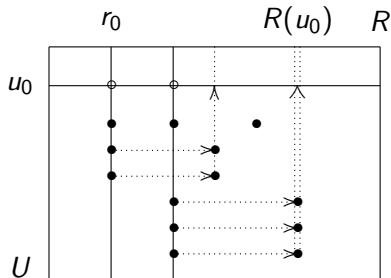
«вместе с объектами,
которые покупал u_0 ,
часто покупают $R(u_0)$ »



- 1 $R(u_0) := \{r \in R \mid \exists r_0: f_{u_0 r_0} \neq \emptyset \text{ и } B(r) = \text{corr}(r, r_0) > \alpha\};$
- 2 сортировка $r \in R(u_0)$ по убыванию $B(r)$, взять top N ;

От объекта (item-based CF)

«вместе с объектами,
которые покупал u_0 ,
часто покупают $R(u_0)$ »



Недостатки:

- рекомендации часто тривиальны (нет коллаборативности);
- не учитывается степень сходства ресурсов r и r_0 ;
- проблема «холодного старта»;
- надо хранить попарные корреляции между объектами;
- нечего рекомендовать нетипичным пользователям.

Резюме по Memory-Based методам

Преимущества:

- Легко понять;
- Легко реализовать.

Недостатки:

- Не хватает теоретического обоснования:
придумано много способов оценить сходство...
придумано много гибридных (item-user-based) методов...
... не ясно, как выбрать лучший;
- Все методы требуют хранения огромной матрицы F .

Далее:

- *Латентные модели — лишены этих недостатков.*

Понятие латентной модели

Латентная модель: по данным D оцениваются векторы:

$(p_{su})_{s \in S}$ — профили клиентов $u \in U$;

$(q_{tr})_{t \in T}$ — профили объектов $r \in R$.

Типы латентных моделей (основные идеи):

❶ Ко-кластеризация:

— жёсткая:
$$\begin{cases} p_{su} = [\text{клиент } u \text{ принадлежит кластеру } s]; \\ q_{tr} = [\text{объект } r \text{ принадлежит кластеру } t]; \end{cases}$$

— мягкая: p_{su} , q_{tr} — степени принадлежности кластерам.

❷ Матричная факторизация: $S = T$;

по p_{tu} , q_{tr} должны восстанавливаться f_{ur} .

❸ Вероятностные (байесовские) модели: $S = T$;

$p_{tu} = p(t|u)$, $q_{tr} = q(t|r)$.

Ко-кластеризация (бикластеризация)

Пусть f_{ur} — рейтинги;

$G = (g(u))_{u \in U}$ — кластеризации клиентов;

$H = (h(r))_{r \in R}$ — кластеризации объектов;

Модель усреднения по блокам (Block Average):

$$\hat{f}_{ur}(G, H) = \bar{f}_{g(u), h(r)} + (\bar{f}_u - \bar{f}_{g(u)}) + (\bar{f}_r - \bar{f}_{h(r)});$$

$\bar{f}_{g(u), h(r)}$ — средние по ко-кластерам;

$\bar{f}_{g(u)}$ и $\bar{f}_{h(r)}$ — средние по кластерам;

\bar{f}_u и \bar{f}_r — средние по клиентам и по объектам;

Функционал качества кластеризации:

$$\sum_{(u,r) \in D} (\hat{f}_{ur}(G, H) - f_{ur})^2 \rightarrow \min_{G, H};$$

Ко-кластеризация: простой алгоритм

Алгоритм ВВАС (Bregman Block Average Co-clustering)

- Инициализировать случайные кластеризации $g(u)$, $h(r)$;
- Повторять пока кластеризации изменяются:
 - 1 Вычислить все средние:
 $\bar{f}_{g(u),h(r)}$; $\bar{f}_{g(u)}$; $\bar{f}_{h(r)}$; \bar{f}_u ; \bar{f}_r ;
 - 2 Вычислить новые кластеризации для всех клиентов $u \in U$:
$$g(u) := \arg \min_g \sum_{r \in D_u} (\hat{f}_{ur}(g, H) - f_{ur})^2;$$
 - 3 Вычислить новые кластеризации для всех объектов $r \in R$:
$$h(r) := \arg \min_h \sum_{u \in D_r} (\hat{f}_{ur}(G, h) - f_{ur})^2;$$

George T., Merugu S. A scalable collaborative filtering framework based on co-clustering // 5-th IEEE int. conf. on Data Mining, 2005, Pp. 27–30.

Banerjee A., et al. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation // 10-th KDDM, 2004, Pp. 509–514.

Матричные разложения

T — множество тем (интересов): $|T| \ll |U|$, $|T| \ll |R|$;

p_{tu} — неизвестный *профиль клиента* u ; $P = (p_{tu})_{|T| \times |U|}$;

q_{tr} — неизвестный *профиль объекта* r ; $Q = (q_{tr})_{|T| \times |R|}$;

Задача: найти разложение $f_{ur} = \sum_{t \in T} \lambda_t p_{tu} q_{tr}$; $F = P^T \Lambda Q$;

Методы решения:

SVD — сингулярное разложение (плохо интерпретируется!);

NNMF — неотрицательное разложение: $p_{tu} \geq 0$, $q_{tr} \geq 0$;

Вероятностная интерпретация:

$$\underbrace{p(u, r)}_{f_{ur} ?} = \sum_{t \in T} \underbrace{p(t)}_{\lambda_t} \cdot \underbrace{p(u|t)}_{p_{tu}} \cdot \underbrace{q(r|t)}_{q_{tr}};$$

$$q(t|r) = \frac{q_{tr} p(t)}{\sum_{\tau \in T} q_{\tau r} p(\tau)}; \quad p(t|u) = \frac{p_{tu} p(t)}{\sum_{\tau \in T} p_{\tau u} p(\tau)}$$

Байесовская модель посещений

T — множество тем (интересов);

$p_{tu} = p(t|u)$ — неизвестный *профиль клиента* u ;

$q_{tr} = q(t|r)$ — неизвестный *профиль объекта* r ;

$p_u = p(u)$ — априорная вероятность клиента u ;

$q_r = q(r)$ — априорная вероятность объекта r ;

Вероятность посещения (u, r) записывается двумя способами:

$$p(u, r) = \begin{cases} \sum_{t \in T} p_u p_{tu} q(r|t, u); & q(r|t) = \frac{q_{tr} q_r}{\sum_{r' \in R} q_{tr'} q_{r'}}; \\ \sum_{t \in T} q_r q_{tr} p(u|t, r); & p(u|t) = \frac{p_{tu} p_u}{\sum_{u' \in U} p_{tu'} p_{u'}}; \end{cases}$$

Задача: оценить профили p_{tu} , q_{tr} .

Принцип максимума правдоподобия: $\sum_{i=1}^m \ln p(u_i, r_i) \rightarrow \max_{p_{tu}, q_{tr}}.$

Общая идея: алгоритм согласования профилей

Повторять итерации, пока профили не сойдутся:

- ① Настройка профилей клиентов p_{tu} при фиксированных q_{tr} :

$$\left\{ \begin{array}{l} \sum_{i=1}^m \ln \left(\sum_{t \in T} p_u p_{tu} q(r|t) \right) \rightarrow \max_{p_{tu}}; \\ \sum_{t \in T} p_{tu} = 1, \quad \forall u \in U; \end{array} \right.$$

- ② Настройка профилей объектов q_{tr} при фиксированных p_{tu} :

$$\left\{ \begin{array}{l} \sum_{i=1}^m \ln \left(\sum_{t \in T} q_r q_{tr} p(u|t) \right) \rightarrow \max_{q_{tr}}; \\ \sum_{t \in T} q_{tr} = 1, \quad \forall r \in R; \end{array} \right.$$

ЕМ-алгоритм (настройка профилей клиентов)

Скрытые переменные $H_{tr}(u) \equiv p(t|r, u)$ — апостериорная вероятность темы t при посещении объекта r клиентом u .

ЕМ-алгоритм:

повторять, пока профили p_{tu} не сойдутся

- **Е-шаг** (вычисление скрытых переменных):

для всех объектов $r \in R$, клиентов $u \in U$, тем $t \in T$

$$H_{tr}(u) := \frac{p_{tu} q(r|t)}{\sum_{t' \in T} p_{t'u} q(r|t')};$$

- **М-шаг** (максимизация правдоподобия):

для всех клиентов $u \in U$, тем $t \in T$

$$p_{tu} := \frac{1}{D_u} \sum_{r \in D_u} H_{tr}(u), \quad \text{где} \quad D_u = \{r: (u, r) \in D\};$$

Симметризованный ЕМ-алгоритм

Инициализировать профили q_{tr} и p_{tu} ;

Повторять итерации, пока все профили не сойдутся:

- 1 Фиксировать q_{tr} ;

Вычислить $q(r|t)$ по формуле Байеса;

Повторять, пока профили клиентов не сойдутся:

- Е-шаг: вычислить скрытые переменные $H_{tr}(u)$;
- М-шаг: вычислить профили клиентов p_{tu} ;

- 2 Фиксировать p_{tu} ;

Вычислить $p(u|t)$ по формуле Байеса;

Повторять, пока профили объектов не сойдутся:

- Е-шаг: вычислить скрытые переменные $H_{tu}(r)$;
- М-шаг: вычислить профили объектов q_{tr} ;

Обобщения, модификации, применения

- Если $f_{ur} \in Z = \{1, 2, \dots, z_{\max}\}$ — рейтинги, то вместо $p(u, r) = P(f_{ur} \neq \emptyset)$ надо оценивать $(z_{\max} - 1)$ вероятностей $p_z(u, r) = P(f_{ur} \leq z)$, $z \in Z$;
- Динамическое обновление профилей при пополнении D ;
- Иерархические профили;
- Учёт априорной информации через начальное приближение профилей:
 - тематический каталог объектов;
 - соц-дем (анкеты) клиентов;
- Унифицированный профиль объектов и клиентов;
- Долгосрочный и краткосрочный профили;
- Оценивание сходства по частям профиля.

Данные поисковой машины Яндекс

Исходные данные:

7 дней работы поисковой машины Яндекс; объём лога 3.6 Гб;
14 606 пользователей;
207 312 запросов;
1 972 636 документов было выдано;
129 600 документов были выбраны пользователями.

Фрагмент лога:

```
1098353321109615996 (номер пользователя)
  французская кухня (запрос) 1110473322 (время запроса) 113906 0
    http://www.naturel.ru/ (сайт или документ)
    http://www.kuking.net/c7.htm 1110473328 (время клика)
    http://www.cooking-book.ru/national/french/
    ...
  жаренное мясо в вине 1110473174 1349 0
    ...
...
```

Данные поисковой машины Яндекс

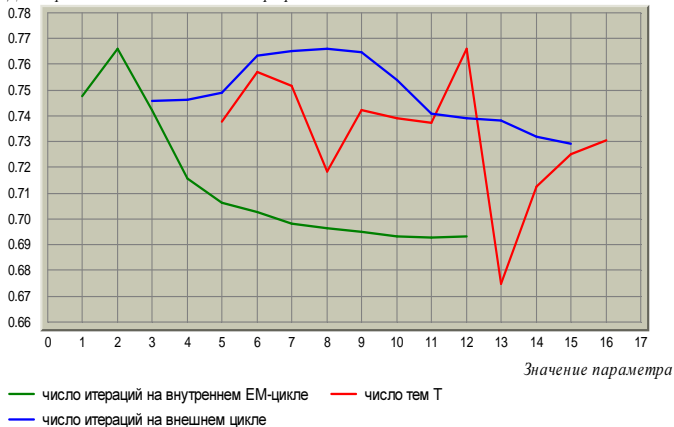
Схема эксперимента:

- Отбор наиболее посещаемых сайтов, $|R| = 1024$.
- Отбор наиболее активных пользователей, $|U| = 7300$.
- Введение критериев качества профилей:
 - 400 сайтов заранее классифицированы на $|T| = 12$ тематических классов;
 - Q_1 = доля неправильно восстановленных профилей;
 - Q_2 = число ошибок классификации методом kNN ;
- Оптимизация параметров по критерию качества.
- Построение профилей и оценок сходства сайтов.
- Визуализация: глобальные и локальные карты сходства.

Результаты: оптимизация числа итераций и $|T|$

Двух итераций на внутреннем цикле уже достаточно!

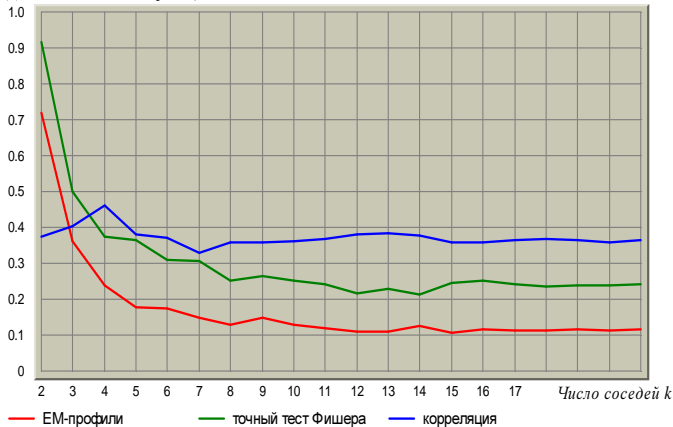
Доля правильно восстановленных профилей



Результаты: подбор меры сходства

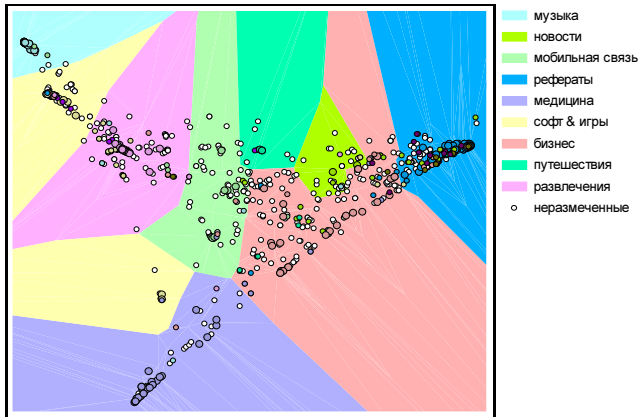
оценки сходства по точному тесту Фишера (FET) лучше корреляций, а по профилям — ещё лучше!

Доля ошибок классификации методом kNN



Результаты: карта сходства Интернета

Многомерное шкалирование по FET-оценкам сходства



Результат: сайты сами сгруппировались по тематикам!

Что такое «многомерное шкалирование» и «карта сходства»?

Дано: попарные расстояния R_{ij} между n объектами.

Найти: координаты этих объектов на плоскости $(x_i, y_i)_{i=1}^n$:

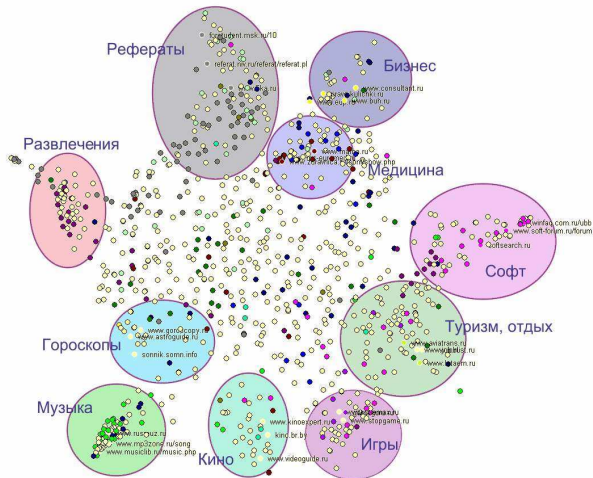
$$S = \sum_{i < j} \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - R_{ij} \right)^2 \rightarrow \min_{(x_i, y_i)_{i=1}^n}$$

Карта сходства (Similarity Map) — это средство разведочного анализа многомерных данных:

- точечный график $(x_i, y_i)_{i=1}^n$;
- близким объектам соответствуют близкие точки;
- оси графика не имеют интерпретации;
- возможны искажения.

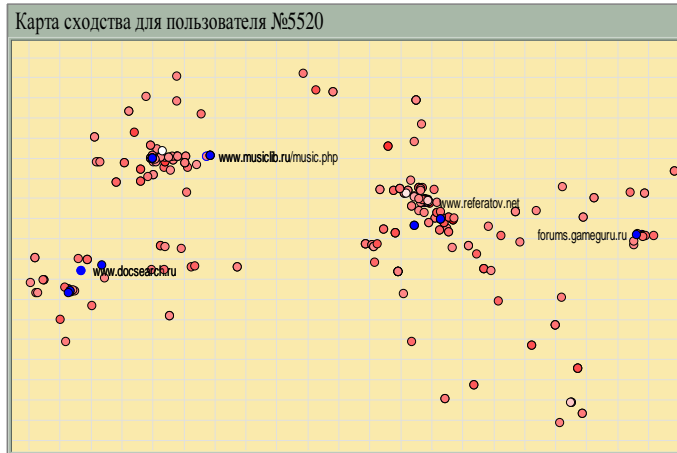
Результаты: карта сходства Интернета

Многомерное шкалирование по профилям, $|T| = 12$



Результаты: локальная карта пользователя

Визуальное представление персональных рекомендаций:



Резюме

Коллаборативная фильтрация — это набор методов для решения задач персонализации и **анализа клиентских сред**.

Простые методы *основаны на хранении исходных данных*.

Латентные модели, основанные на оценивании профилей клиентов и объектов, обладают рядом преимуществ:

- оценки сходства клиентов и объектов более адекватны;
- с профилями можно делать многое:
 - содержательно интерпретировать;
 - частично оценивать по априорным данным;
 - обновлять динамически по мере поступления данных;
 - сравнивать целиком или по фрагментам;
- снимается проблема «холодного старта»;
- резко сокращается объём хранимых данных;

- ❶ Спасибо за внимание!
- ❷ Вопросы?
- ❸ Ссылки: вики www.MachineLearning.ru
 - «Участник:Vokov»
 - «Анализ клиентских сред»
 - «Коллаборативная фильтрация»
- ❹ Ещё ссылки (хорошие подборки статей по CF):
 - jamesthornton.com/cf
 - www.adastral.ucl.ac.uk/~junwang/CollaborativeFiltering.html