Коллаборативная фильтрация

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

6 ноября 2013

Содержание

Определения и обозначения

U — множество субъектов (users/пользователей/клиентов);
I — множество объектов (items/предметов/товаров/ресурсов);
Y — пространство описаний транзакций;
$D = (u_t, i_t, y_t)_{t=1}^m \in U \times I \times Y$ — транзакционные данные;

Агрегированные данные:

$R = \|r_{ui}\|$ — матрица кросс-табуляции размера $|U| \times |I|$,
где $r_{ui} = \text{aggr}\{(u_t, i_t, y_t) \in D \mid u_t = u,\ i_t = i\}$

Задачи:

- прогнозирование незаполненных ячеек r_{ui};
- оценивание сходства: $\rho(u, u')$, $\rho(i, i')$, $\rho(u, i)$;
- формирование списка рекомендаций для u или для i.

Пример 1. Рекомендательная система по посещениям

U — пользователи Интернет;
I — страницы (сайты, документы, новости, и т.п.);
$r_{ui} = [$пользователь u посетил страницу $i]$;

Основная гипотеза Web Usage Mining:

- Посещения пользователя характеризуют его интересы, вкусы, привычки, возможности.

Задачи персонализации предложений:

- для пользователя u:
  — выдать оценку страницы i;
  — выдать ранжированный список рекомендуемых страниц;
- для страницы i: выдать список страниц, близких к i.

Пример: http://SurfingBird.ru

Пример 2. Рекомендательная система по покупкам

U — клиенты интернет-магазина;
I — товары (книги, видео, музыка, и т.п.);
$r_{ui} = [\text{клиент } u \text{ купил товар } i]$;

Задачи персонализации предложений:

- выдать оценку товара i для клиента u;
- выдать клиенту u список рекомендуемых товаров;
- предложить совместную покупку (cross-selling);
- информировать клиента о новом товаре (up-selling);
- сегментировать клиентскую базу;
  выделить интересы клиентов (найти целевые аудитории).

Примеры:

http://amazon.com, http://ozon.ru, http://netflix.com

Пример 3. Рекомендательная система на основе рейтингов

U — клиенты интернет-магазина;

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

r_{ui} = рейтинг, который клиент u выставил товару i;

Задачи персонализации предложений — те же.

Пример: конкурс Netflix [www.netflixprize.com]

- 2 октября 2006 — 21 сентября 2009; главный приз — $\$10^6$;
- $|U| = 0.48 \cdot 10^6$; $|I| = 17 \cdot 10^3$;
- 10^8 рейтингов $\{1, 2, 3, 4, 5\}$;
- точность прогнозов оценивается по тестовой выборке D':

$$\text{RMSE}^2 = \frac{1}{|D'|} \sum_{(u,i) \in D'} \left(r_{ui} - \hat{r}_{ui}\right)^2;$$

- задача: уменьшить RMSE с 0.9514 до 0.8563 (на 10%).

Пример 4. Анализ текстов

U — текстовые документы (статьи, новости, и т.п.);
I — ключевые слова или выражения;
r_{ui} = частота встречаемости слова i в тексте u.

Задачи тематического моделирования (topic modeling):

- Поиск научной информации:
  по тексту u определить его тематику;
  найти тексты по данной тематике;
- Выявление трендов и фронта исследований,
  «где передний край науки по данной теме?»
- Поиск экспертов (expert search), рецензентов, проектов
- Анализ и агрегирование новостных потоков
- Аннотация генома

Пример 5. Анализ изображений

U — изображения;
I — найденные на изображениях элементы;
$r_{ui} = [\text{изображение } u \text{ содержит элемент } i]$.

Задачи тематического моделирования (topic modeling):

- кластеризовать изображения по темам;
- построить иерархический каталог тем;
- по изображению u найти схожие;

Задачи связывания изображений и текстов:

- аннотировать (тегировать) изображение;
- по изображению найти описание;
- по описанию найти изображение;

Пример 6. Социальные сети, форумы, блоги

U — пользователи;
D — текстовые документы (обсуждения, блоги);
W — ключи (ключевые слова или выражения);
$r_{ud} = [\text{пользователь } u \text{ прочитал/написал } d]$;
$g_{rw} =$ частота встречаемости ключа w в тексте d;
$h_{uv} = [\text{пользователь } u \text{ — друг пользователя } v]$.

Некоторые задачи анализа социальной сети:

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

Два основных подхода в коллаборативной фильтрации

1. **Корреляционные модели**
   (Memory-Based Collaborative Filtering)
   - хранение всей исходной матрицы данных R
   - сходство клиентов — корреляция строк матрицы R
   - сходство объектов — корреляция столбцов матрицы R

2. **Латентные модели**
   (Latent Models for Collaborative Filtering)
   - оценивание профилей клиентов и объектов
     (*профиль — это вектор скрытых характеристик*)
   - хранение профилей вместо хранения R
   - сходство клиентов и объектов — сходство их профилей

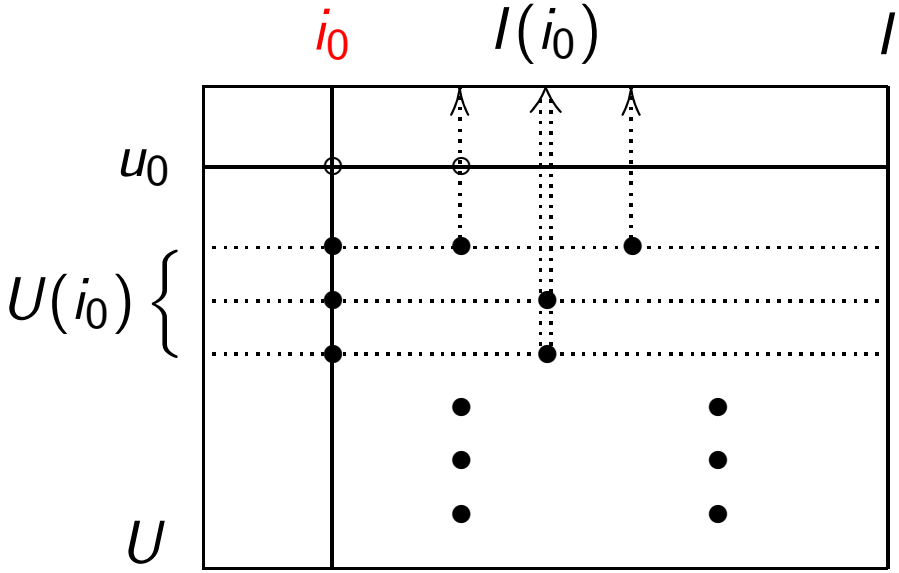
Подборки статей по коллаборативной фильтрации:

jamesthornton.com/cf

http://web4.cs.ucl.ac.uk/staff/jun.wang/blog/tag/recommendation

Тривиальная рекомендательная система

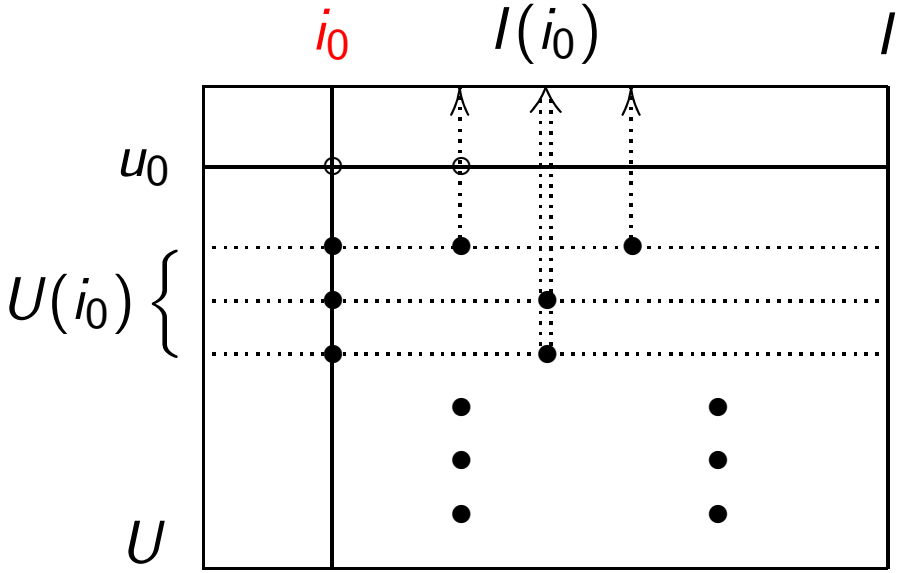
«клиенты, купившие i_0, также покупали $I(i_0)$» [Amazon.com]



1. $U(i_0) := \left\{ u \in U \;\middle|\; r_{ui_0} \neq \varnothing,\ u \neq u_0 \right\}$ — коллаборация;
2. $I(i_0) := \left\{ i \in I \;\middle|\; \text{sim}(i, i_0) = \frac{|U(i_0) \cap U(i)|}{|U(i_0) \cup U(i)|} > \delta \right\}$,
   где $\text{sim}(i, i_0)$ — одна из возможных мер сходства i и i_0;
3. отсортировать $I(i_0)$ по убыванию $\text{sim}(i, i_0)$, взять top N.

Тривиальная рекомендательная система

«клиенты, купившие i_0, также покупали $I(i_0)$»
[Amazon.com]

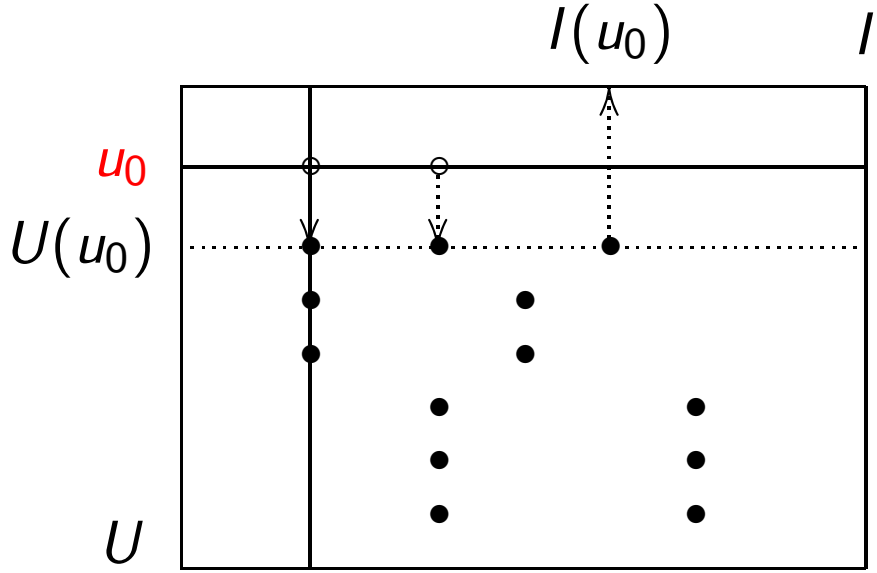


Недостатки:

- рекомендации тривиальны
  (предлагается всё наиболее популярное);
- не учитываются интересы конкретного пользователя u_0;
- проблема «холодного старта»;
  (новый товар никому не рекомендуется)
- надо хранить всю матрицу R.

От клиента (user-based CF)

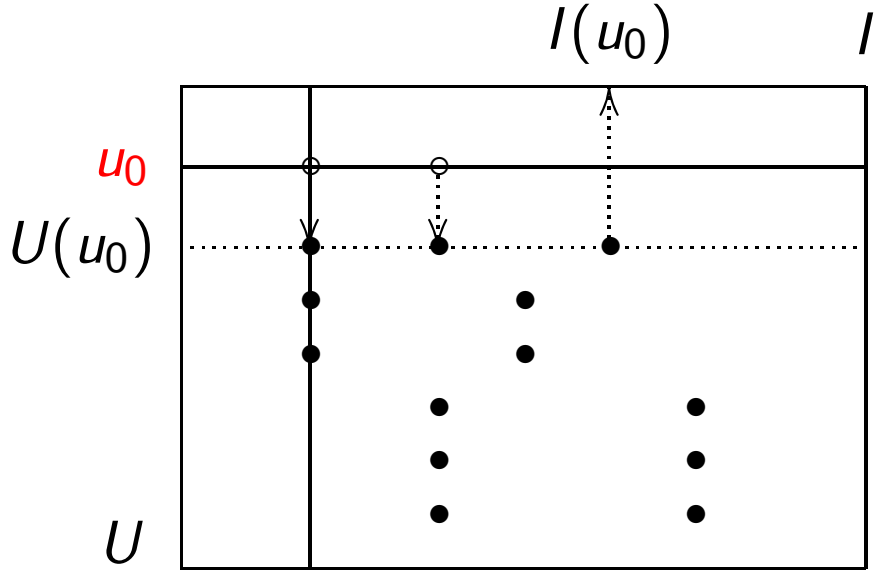
«клиенты, похожие на u_0, также покупали $I(u_0)$»



1. $U(u_0) := \{u \in U \mid \text{sim}(u_0, u) > \alpha\}$ — коллаборация;
   $\text{sim}(u_0, u)$ — одна из возможных мер близости u к u_0;
2. $I(u_0) := \left\{i \in I \;\middle|\; B(i) = \frac{|U(u_0) \cap U(i)|}{|U(u_0) \cup U(i)|} > 0\right\}$;
   где $U(i) := \{u \in U \mid r_{ui} \neq \varnothing\}$;
3. отсортировать $i \in I(u_0)$ по убыванию $B(i)$, взять top N;

От клиента (user-based CF)

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

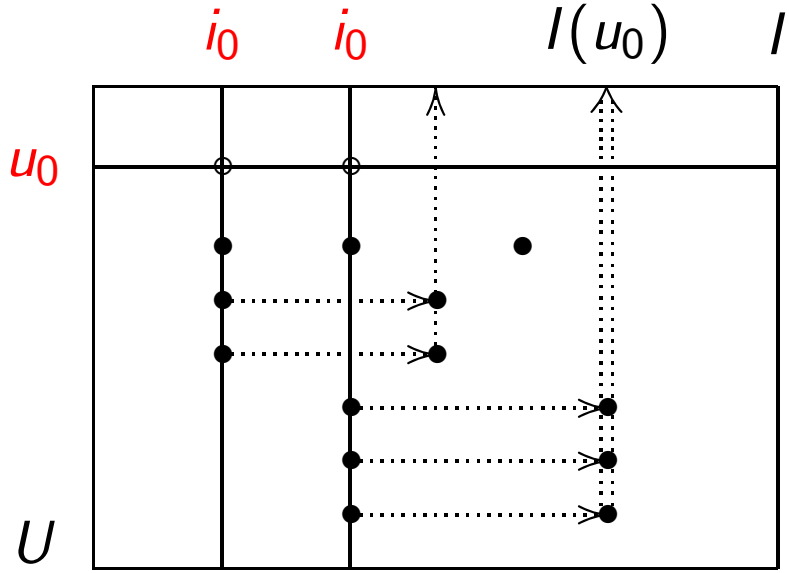


Недостатки:

- ~~рекомендации тривиальны~~;
- ~~не учитываются интересы конкретного пользователя u_0~~;
- проблема «холодного старта»;
- надо хранить всю матрицу R;
- нечего рекомендовать нетипичным/новым пользователям.

От объекта (item-based CF)

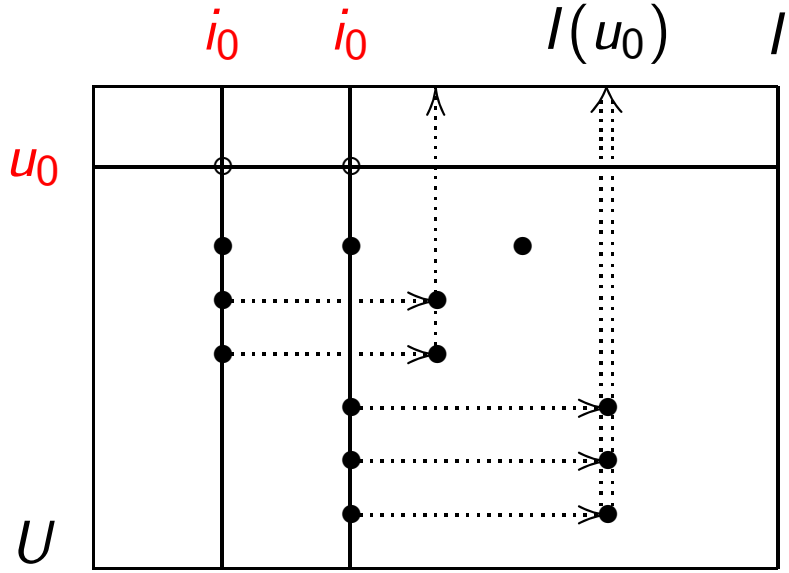
«вместе с объектами,
которые покупал u_0,
часто покупают $I(u_0)$»



1. $I(u_0) := \{i \in I \mid \exists i_0 : r_{u_0 i_0} \neq \varnothing$ и $B(i) = \mathrm{sim}(i, i_0) > \alpha\}$;
   где $\mathrm{sim}(i, i_0)$ — одна из возможных мер сходства i и i_0;
2. сортировка $i \in I(u_0)$ по убыванию $B(i)$, взять top N;

От объекта (item-based CF)

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



Недостатки:

- рекомендации часто тривиальны (нет коллаборативности);
- проблема «холодного старта»;
- надо хранить всю матрицу R;
- ~~нечего рекомендовать нетипичным пользователям~~.

Восстановление пропущенных значений (рейтингов)

Непараметрическая регрессия Надарайя–Ватсона:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum\limits_{u' \in U_\alpha(u)} \text{sim}(u,u')(r_{u'i} - \bar{r}_{u'})}{\sum\limits_{u' \in U_\alpha(u)} \text{sim}(u,u')},$$

где $\bar{r}_u = \frac{1}{|I(u)|} \sum\limits_{i \in I(u)} r_{ui}$ — средний рейтинг клиента u,

$I(u)$ — множество объектов, которые клиент u оценил,

$\text{sim}(u,u')$ — сглаживающее ядро, функция близости u и u',

$U_\alpha(u) = \{u' \mid \text{sim}(u',u) > \alpha\}$ — коллаборация клиента u.

Недостатки:

- проблема «холодного старта»;
- надо хранить всю матрицу R;

Напоминание. Регрессия и метод наименьших квадратов

- X — объекты (часто $\mathbb{R}^n$); Y — ответы (часто $\mathbb{R}$, реже $\mathbb{R}^m$);
  $X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка;
  $y_i = y(x_i)$, $y : X \to Y$ — неизвестная зависимость;
- $a(x) = f(x, \alpha)$ — модель зависимости,
  $\alpha \in \mathbb{R}^p$ — вектор параметров модели.
- Метод наименьших квадратов (МНК):

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} w_i \big(f(x_i, \alpha) - y_i\big)^2 \to \min_\alpha,$$

где w_i — вес, степень важности i-го объекта.

Напоминание. Формула Надарая–Ватсона

Приближение константой $a(x) = \alpha$ в окрестности $x \in X$:

$$Q(\alpha; X^\ell) = \sum_{i=1}^{\ell} w_i(x)\big(\alpha - y_i\big)^2 \to \min_{\alpha \in \mathbb{R}};$$

где $w_i(x) = K\left(\frac{\rho(x,x_i)}{h}\right)$ — веса объектов x_i относительно x;
$K(r)$ — *ядро*, невозрастающее, ограниченное, гладкое;
h — ширина окна сглаживания.

Формула *ядерного сглаживания* Надарая–Ватсона:

$$a_h(x; X^\ell) = \frac{\sum\limits_{i=1}^{\ell} y_i w_i(x)}{\sum\limits_{i=1}^{\ell} w_i(x)} = \frac{\sum\limits_{i=1}^{\ell} y_i K\left(\frac{\rho(x,x_i)}{h}\right)}{\sum\limits_{i=1}^{\ell} K\left(\frac{\rho(x,x_i)}{h}\right)}.$$

Функции близости, используемые в корреляционных методах

- корреляция Пирсона:

$$\text{sim}(u,u') = \frac{\sum\limits_{i \in I(u,u')} (r_{ui} - \bar{r}_u)(r_{u'i} - \bar{r}_{u'})}{\sqrt{\sum\limits_{i \in I(u,u')} (r_{ui} - \bar{r}_u)^2 \sum\limits_{i \in I(u,u')} (r_{u'i} - \bar{r}_{u'})^2}};$$

- косинусная мера близости:

$$\text{sim}(u,u') = \frac{\sum\limits_{i \in I(u,u')} r_{ui} r_{u'i}}{\sqrt{\sum\limits_{i \in I(u,u')} r_{ui}^2 \sum\limits_{i \in I(u,u')} r_{u'i}^2}};$$

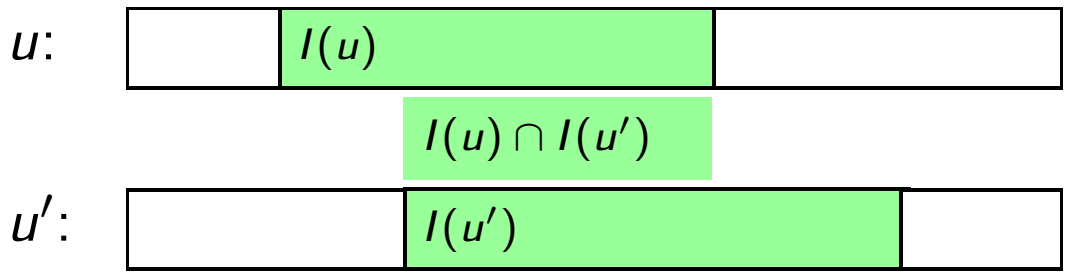
где $I(u,u') = \begin{cases} I(u) \cup I(u'), & \text{для бинарных данных,} \\ I(u) \cap I(u'), & \text{для рейтинговых данных.} \end{cases}$

- статистические критерии:
  χ^2, точный тест Фишера (для бинарных данных).

Функции близости на основе точного теста Фишера (FET)

Рассмотрим случай бинарных данных, $r_{ui} \in \{0, 1\}$.

Нулевая гипотеза:
клиенты u и u' совершают свой выбор независимо.



Вероятность случайной реализации i совместных выборов

$$p(i) = P\{|I(u) \cap I(u')| = i\} = \frac{C^{i}_{|I(u)|} C^{|I(u')|-i}_{|I|-|I(u)|}}{C^{|I(u')|}_{|I|}}.$$

Функция близости $I(u, u') = -\log p\big(|I(u) \cap I(u')|\big)$.

Резюме по Memory-Based методам

Преимущества для бизнес-приложений:

- Легко понять.
- Легко реализовать.

Недостатки:

- Не хватает теоретического обоснования:
  придумано много способов оценить сходство...
  придумано много гибридных (item-user-based) методов...
  ...и не ясно, что лучше;
- Все методы требуют хранения огромной матрицы R.
- Проблема «холодного старта».

Далее:

- *Латентные модели* — лишены этих недостатков.

Понятие латентной модели

Латентная модель: по данным D оцениваются векторы:
$(p_{tu})_{t\in G}$ — профили клиентов $u \in U$, $\quad |G| \ll |I|$;
$(q_{ti})_{t\in H}$ — профили объектов $i \in I$, $\quad |H| \ll |U|$.

Типы латентных моделей (основные идеи):

1. Ко-кластеризация:
   — жёсткая: $\begin{cases} p_{tu} = \big[\text{клиент } u \text{ принадлежит кластеру } t \in G\big]; \\ q_{ti} = \big[\text{объект } i \text{ принадлежит кластеру } t \in H\big]; \end{cases}$
   — мягкая: p_{tu}, q_{ti} — степени принадлежности кластерам.
2. Матричные разложения: $G \equiv H$ — множество тем;
   по p_{tu}, q_{ti} должны восстанавливаться r_{ui}.
3. Вероятностные модели: $G \equiv H$ — множество тем;
   $p_{tu} = p(t|u)$, $q_{ti} = q(t|i)$.

Бикластеризация (ко-кластеризация)

Пусть r_{ui} — вещественные числа или рейтинги;

$g\colon U \to G$ — функции кластеризации клиентов ($|G| < \infty$);

$h\colon I \to H$ — функции кластеризации объектов ($|H| < \infty$);

Модель усреднения по блокам (Block Average):

$$\hat{r}_{ui}(g,h) = \bar{r}_{g(u),h(i)} + (\bar{r}_u - \bar{r}_{g(u)}) + (\bar{r}_i - \bar{r}_{h(i)});$$

$\bar{r}_{g(u),h(i)}$ — средние по *бикластерам*;

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

$\bar{r}_u$ и $\bar{r}_i$ — средние по клиентам и по объектам;

Функционал качества бикластеризации:

$$\sum_{(u,i)\in D} \big(\hat{r}_{ui}(g,h) - r_{ui}\big)^2 \to \min_{g,h};$$

Напоминание. Задача кластеризации

Дано:

X — пространство объектов;

$X^\ell = \{x_i\}_{i=1}^\ell$ — обучающая выборка;

$\rho\colon X \times X \to [0, \infty)$ — функция расстояния между объектами.

Найти:

Y — множество кластеров и

$a\colon X \to Y$ — алгоритм кластеризации, такие, что:

— каждый кластер состоит из близких объектов;

— объекты разных кластеров существенно различны.

Кластеризация — это *обучение без учителя*.

Напоминание. Метод k-средних (k-means)

Пусть $X = \mathbb{R}^n$.

1: задать начальные приближения центров μ_y, $y \in Y$;
2: **повторять**
3: отнести каждый x_i к ближайшему центру:
$y_i := \arg\min_{y \in Y} \rho(x_i, \mu_y)$, $i = 1, \ldots, \ell$;
4: вычислить новые положения центров:
$\mu_y = \arg\min_{\mu} \sum_{i=1}^{\ell} [y_i = y](\mu - x_i)^2$; аналитическое решение этой задачи наименьших квадратов:
$$\mu_{yj} := \frac{\sum_{i=1}^{\ell} [y_i = y] f_j(x_i)}{\sum_{i=1}^{\ell} [y_i = y]}, \quad y \in Y, \quad j = 1, \ldots, n;$$
5: **пока** y_i не перестанут изменяться;

Алгоритм бикластеризации, похожий на k-means

Алгоритм BBAC (Bregman Block Average Co-clustering)

1: инициализировать случайные кластеризации $g(u)$, $h(i)$;
2: **пока** кластеризации изменяются
3: вычислить средние по бикластерам $\bar{r}_{gh}$ и кластерам $\bar{r}_g$, $\bar{r}_h$;
4: вычислить новые кластеризации для всех клиентов $u \in U$:

$$g(u) := \arg\min_g \sum_i \big(\hat{r}_{ui}(g, h(i)) - r_{ui}\big)^2;$$

5: вычислить новые кластеризации для всех объектов $i \in I$:

$$h(i) := \arg\min_h \sum_u \big(\hat{r}_{ui}(g(u), h) - r_{ui}\big)^2;$$

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): $|T| \ll |U|$, $|T| \ll |I|$;

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

q_{ti} — неизвестный *профиль объекта* i; $Q = \left(q_{ti}\right)_{|T|\times|I|}$;

Задача: найти разложение $r_{ui} = \sum\limits_{t\in T} \pi_t p_{tu} q_{ti}$;

Матричная запись: $R = P^{\mathrm{T}} \Delta Q$, $\Delta = \operatorname{diag}\left(\pi_1, \ldots, \pi_{|T|}\right)$;

Вероятностный смысл: $\underbrace{p(u,i)}_{r_{ui}?} = \sum\limits_{t\in T} \underbrace{p(t)}_{\pi_t} \cdot \underbrace{p(u|t)}_{p_{tu}} \cdot \underbrace{q(i|t)}_{q_{ti}}$;

Методы решения:

SVD — сингулярное разложение (плохо интерпретируется);

NNMF — неотрицательное матричное разложение: $p_{tu} \geqslant 0$, $q_{ti} \geqslant 0$;

PLSA — вероятностный латентный семантический анализ.

Разреженный SVD (Singular Value Decomposition)

Обычный не разреженный SVD: $\|R - P^{\mathsf{T}}Q\|^2 \to \min\limits_{P,Q}$.

Разреженный SVD: $\sum\limits_{(u,i)\in D} \Bigg(\underbrace{r_{ui} - \bar{r}_u - \bar{r}_i - \sum\limits_{t\in T} p_{tu}q_{ti}}_{\varepsilon_{ui}} \Bigg)^2 \to \min\limits_{P,Q}$.

Метод стохастического градиента:
перебираем все $(u,i) \in D$ многократно в случайном порядке
и делаем каждый раз градиентный шаг для задачи $\varepsilon_{ui}^2 \to \min\limits_{p_u,q_i}$:

$$p_{tu} := p_{tu} + \eta\varepsilon_{ui}q_{ti}, \quad t \in T;$$
$$q_{ti} := q_{ti} + \eta\varepsilon_{ui}p_{ti}, \quad t \in T;$$

Tacáks G., Pilászy I., Németh B., Tikk D. Scalable collaborative filtering approaches for large recommendation systems // JMLR, 2009, No. 10, Pp. 623–656.

Разреженный SVD (Singular Value Decomposition)

Преимущества метода стохастического градиента:

- легко вводится регуляризация:
$$\varepsilon_{ui}^2 + \lambda\|p_u\|^2 + \mu\|q_i\|^2 \to \min_{p_u, q_i};$$
- легко вводятся ограничения неотрицательности:
$$p_{tu} \geqslant 0, \quad q_{ti} \geqslant 0 \quad \text{(метод проекции градиента)};$$
- легко вводятся обобщение для ранговых данных:
$$\sum_{(u,i)\in D} \left(r_{ui} - \bar{r}_u - \bar{r}_i - \beta\left(\sum_{t\in T} p_{tu} q_{ti}\right)\right)^2 \to \min_{P,Q,\beta}.$$
- легко реализуются все виды инкрементности: добавление
  – ещё одного клиента u,
  – ещё одного объекта i,
  – ещё одного значения r_{ui}.
- высокая численная эффективность на больших данных;

NNMF (Non-Negative Matrix Factorization)

Метод чередующихся наименьших квадратов (Alternating Least Squares, ALS):

$$D = \left\| R - \sum_{t \in T} p_t q_t^{\mathsf{T}} \right\|^2 = \left\| R_t - p_t q_t^{\mathsf{T}} \right\|^2 \to \min_{\{p_t \geqslant 0,\, q_t \geqslant 0\}}$$

Идея: искать поочерёдно то строки p_t, то строки q_t при фиксированных остальных $s \neq t$, $R_t = R - \sum_{s \in T \setminus t} p_s q_s^{\mathsf{T}}$.

$$\frac{\partial D}{\partial p_t} = 0 \quad \Rightarrow \quad (p_t^{\mathsf{T}} q_t - R_t) q_t^{\mathsf{T}} = 0 \quad \Rightarrow \quad p_t = \left(\frac{q_t R_t^{\mathsf{T}}}{q_t q_t^{\mathsf{T}}} \right)_+$$

$$\frac{\partial D}{\partial q_t} = 0 \quad \Rightarrow \quad p_t (p_t^{\mathsf{T}} q_t - R_t) = 0 \quad \Rightarrow \quad q_t = \left(\frac{p_t R_t}{p_t p_t^{\mathsf{T}}} \right)_+$$

Cichocki A., Zdunek R., Amari S., Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization // Springer LNCS, 2007, v.4666, pp.169–176.

Данные Яндекс (Интернет-математика 2005)

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
    ...
...
```

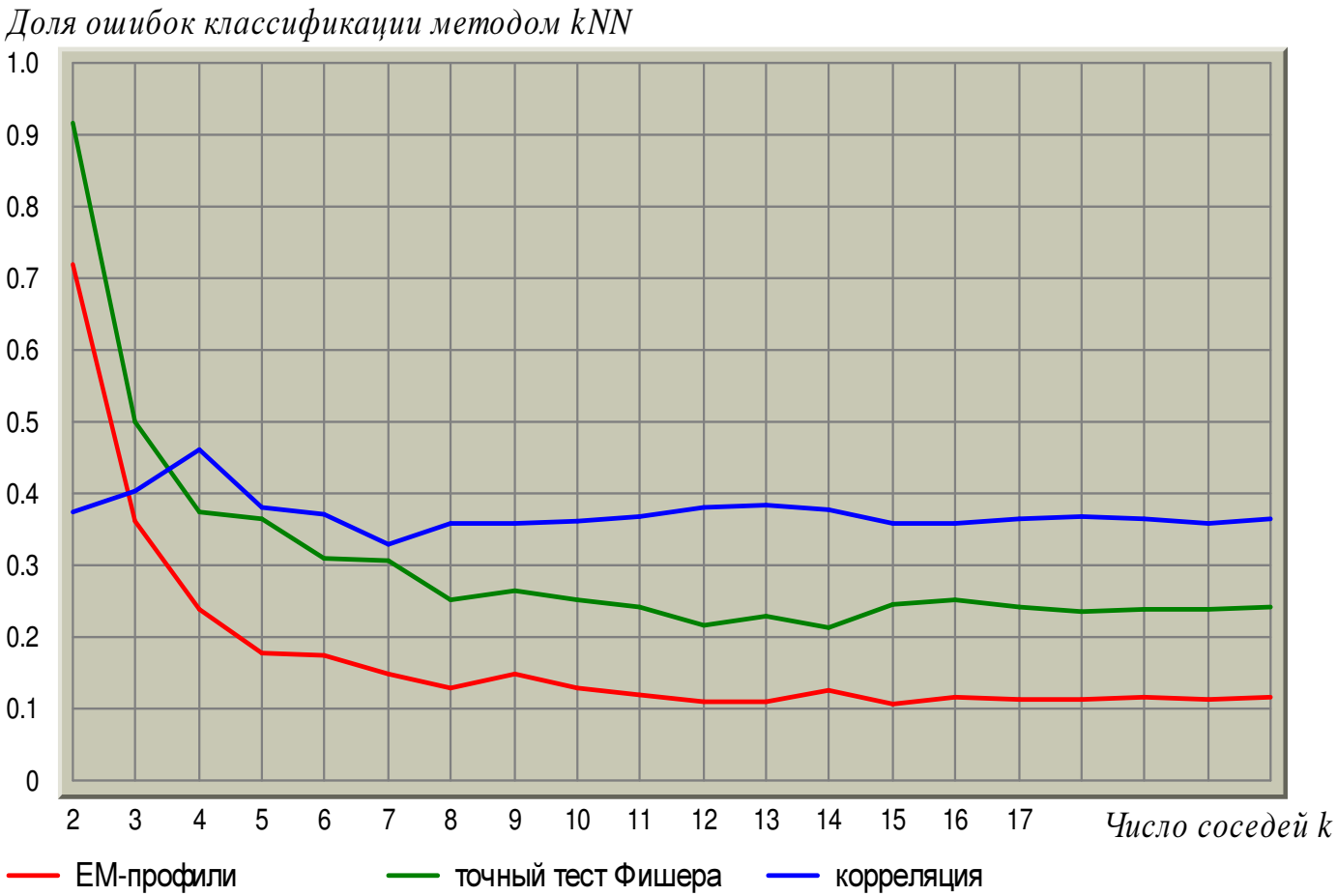
Данные Яндекс (Интернет-математика 2005)

Схема эксперимента:

- Отбор наиболее посещаемых сайтов, $|I| = 1024$.
- 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: подбор меры сходства

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



Задача многомерного шкалирования (multidimensional scaling)

Дано: попарные расстояния R_{ij} между n объектами.
Найти: координаты этих объектов на плоскости $(x_i, y_i)_{i=1}^n$:

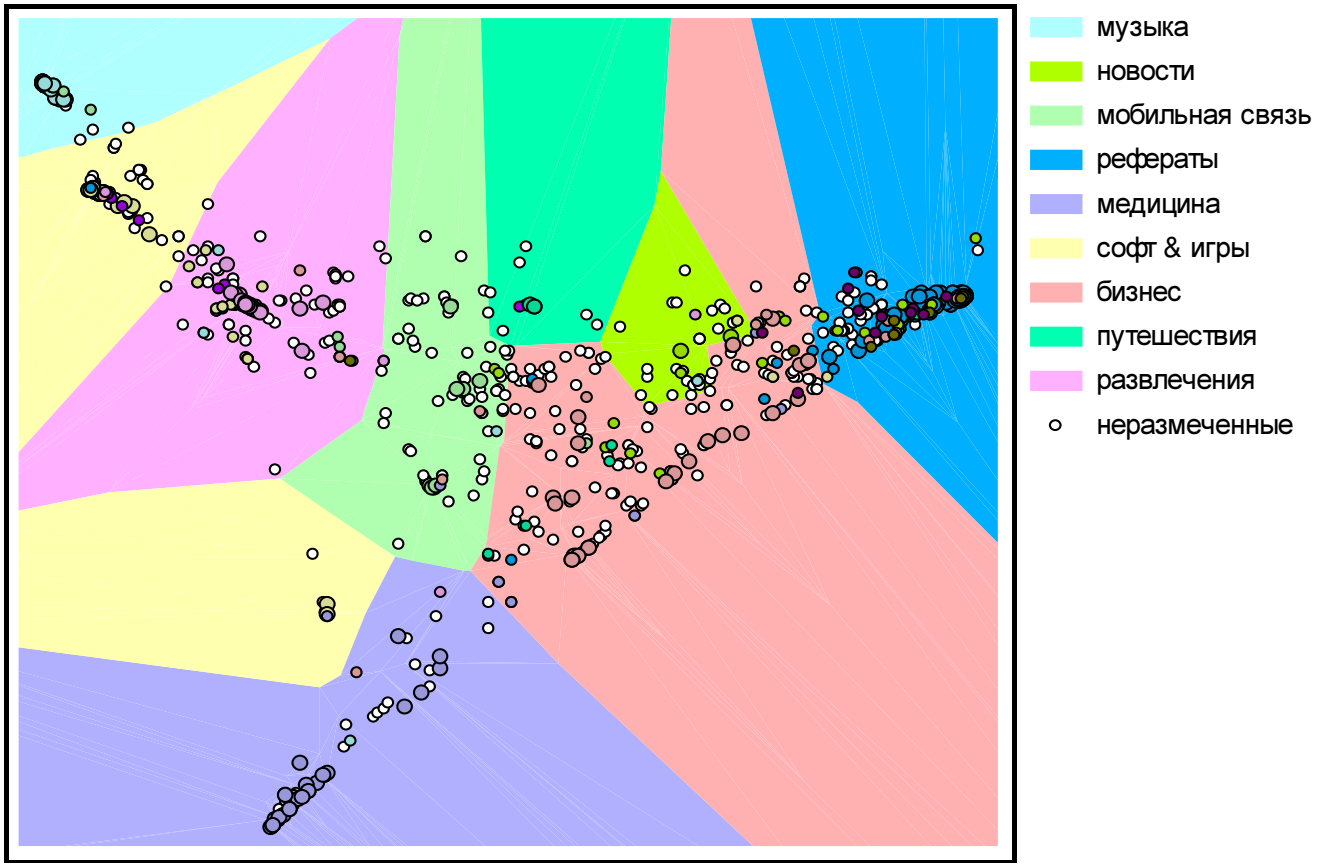
$$S = \sum_{i<j} \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - R_{ij} \right)^2 \to \min_{(x_i, y_i)_{i=1}^n}$$

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.

Карта поисковых интересов пользователей Рунета

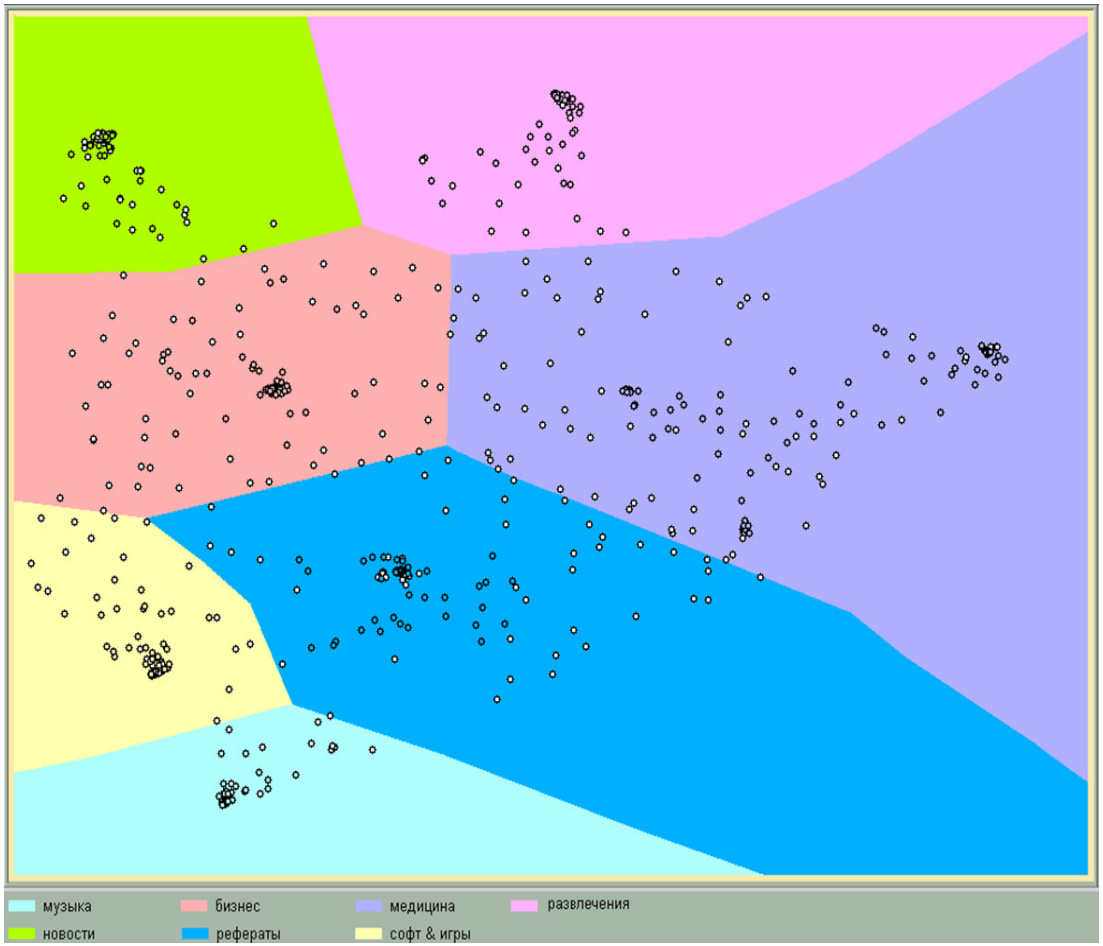
Многомерное шкалирование по FET-оценкам сходства, $|T| = 9$



Результат: темы удаётся проинтерпретировать :)

Карта поисковых интересов пользователей Рунета

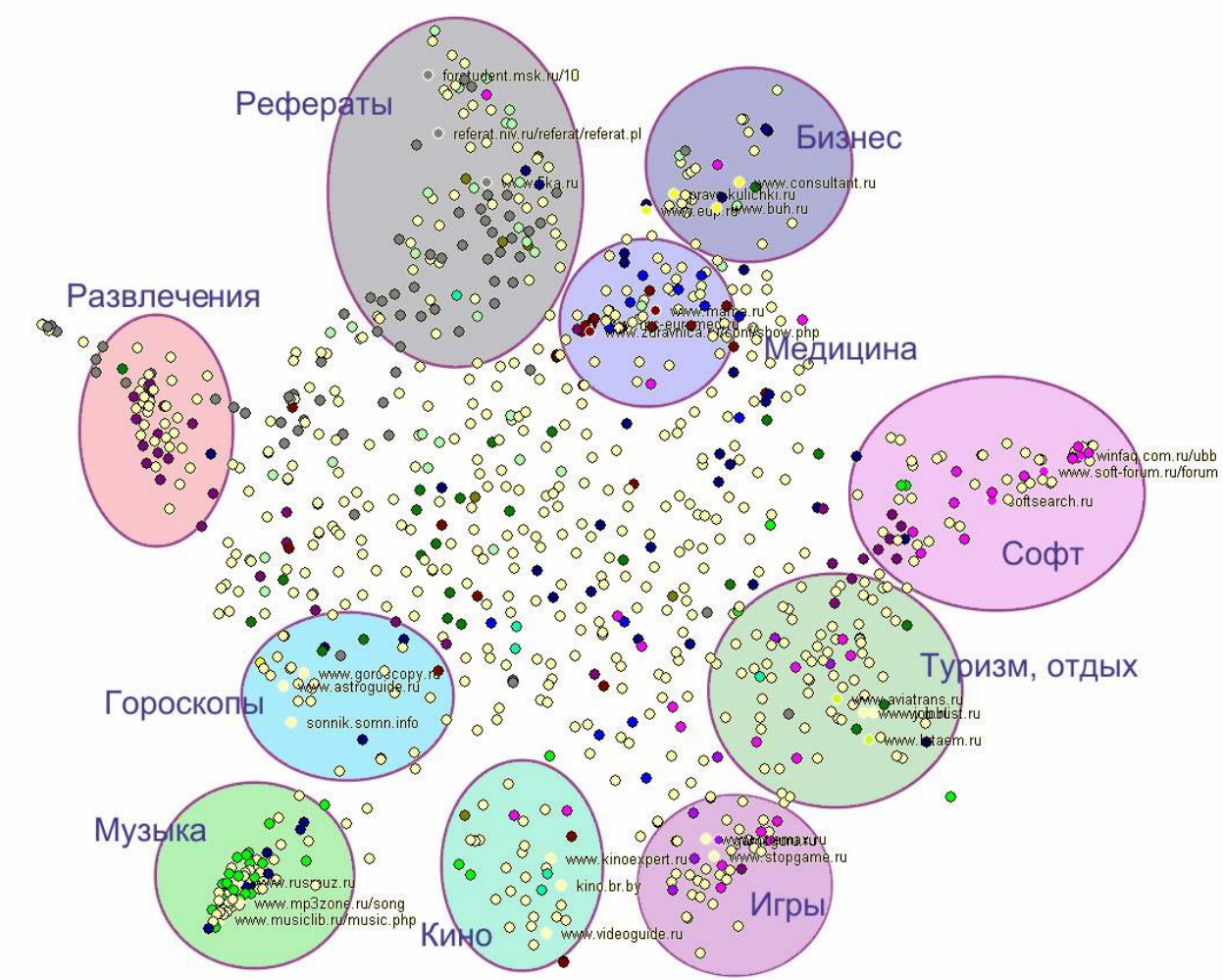
Многомерное шкалирование по профилям, $|T| = 7$



Результат кажется более содержательным :)

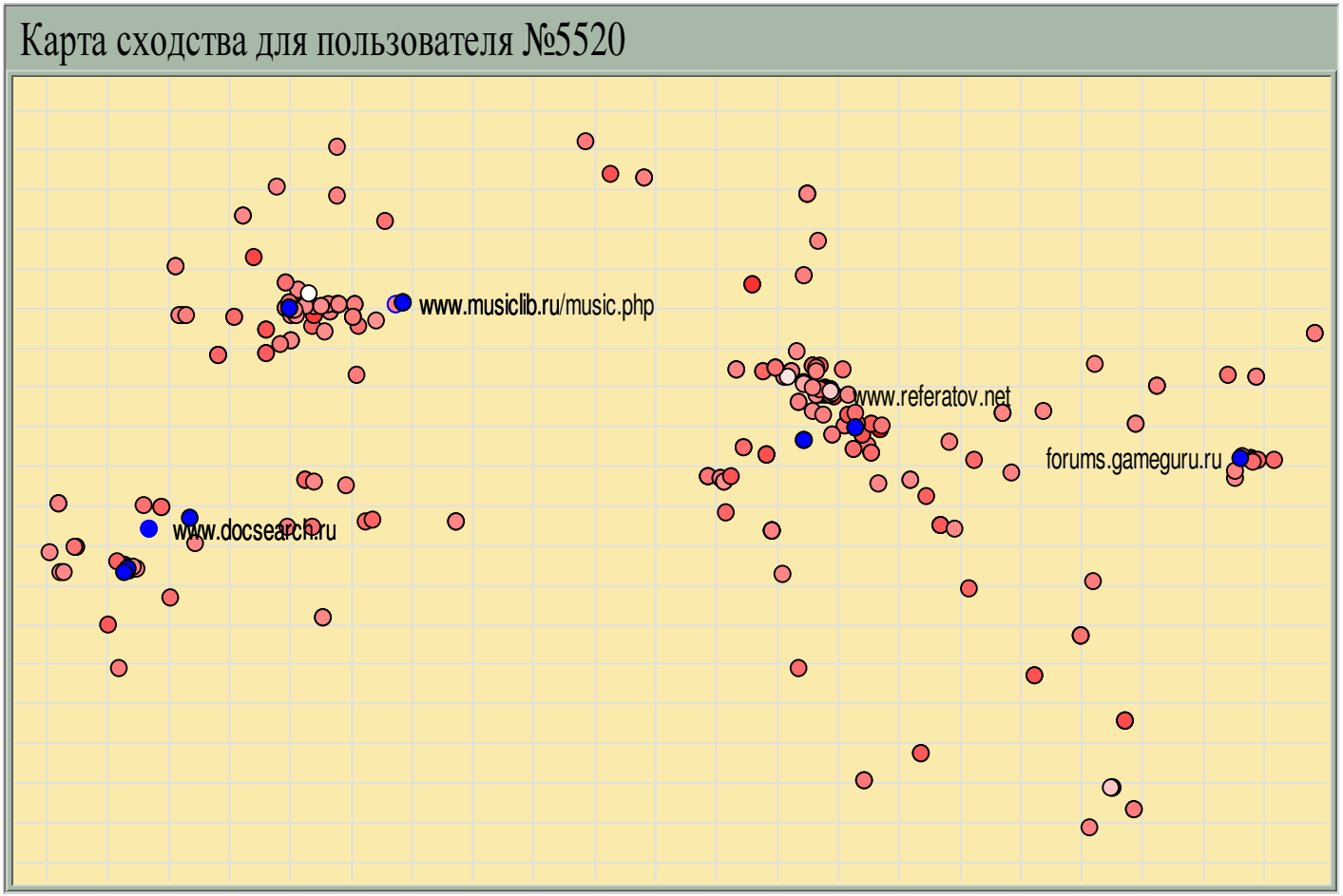
Карта поисковых интересов пользователей Рунета

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



Ещё одна визуализация: локальная карта пользователя

Визуальное представление персональных рекомендаций:



Резюме

Коллаборативная фильтрация (Collaborative Filtering) — это набор методов для построения рекомендательных систем (Recommender Systems).

Латентные модели обладают рядом преимуществ:

- тематические профили содержательно интерпретируемы, могут оцениваться по внешним данным,
- что позволяет решать проблему «холодного старта»
- и строить тематическую кластеризацию (таксономию);
- оценки сходства клиентов и объектов более адекватны;
- резко сокращается объём хранимых данных.