

Коллаборативная фильтрация и матричные разложения

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)»

2 декабря 2016

- 1 Постановка задачи и приложения**
 - Постановка задачи
 - Примеры приложений
 - Модели коллаборативной фильтрации
- 2 Корреляционные модели**
 - Модели, основанные на хранении данных
 - Задача восстановления пропущенных значений
 - Функции близости
- 3 Модели латентной семантики**
 - Матричные разложения
 - Учёт дополнительных признаков данных
 - Измерение качества рекомендаций

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} = [клиент u 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 — \$10⁶;
- $|U| = 0.48 \cdot 10^6$; $|I| = 17 \cdot 10^3$;
- 10⁸ рейтингов $\{1, 2, 3, 4, 5\}$;
- точность прогнозов оценивается по тестовой выборке D' :

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

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

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

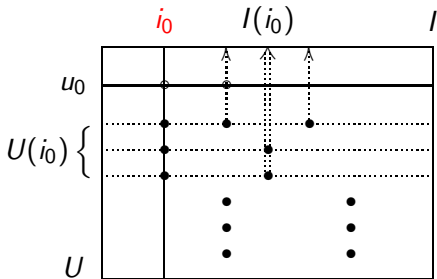
Подборки статей по коллаборативной фильтрации:

jamesthornton.com/cf

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

Тривиальная рекомендательная система

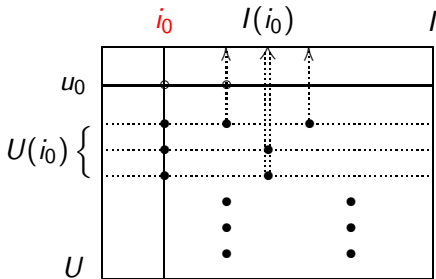
«клиенты, купившие i_0 ,
 также покупали $I(i_0)$ »
 [Amazon.com]



- 1 $U(i_0) := \{u \in U \mid r_{ui_0} \neq \emptyset, u \neq u_0\}$ — коллаборация;
- 2 $I(i_0) := \left\{ i \in I \mid \text{sim}(i, i_0) = \frac{|U(i_0) \cap U(i)|}{|U(i_0) \cup U(i)|} > \delta \right\}$,
 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)

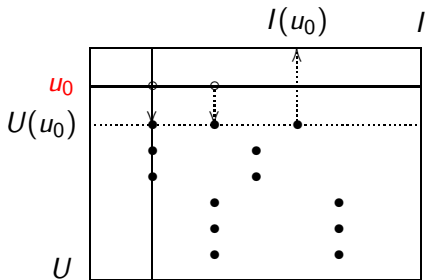
«клиенты, похожие на u_0 ,
также покупали $I(u_0)$ »



- 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 $I(u_0) := \left\{i \in I \mid B(i) = \frac{|U(u_0) \cap U(i)|}{|U(u_0) \cup U(i)|} > 0\right\}$;
где $U(i) := \{u \in U \mid r_{ui} \neq \emptyset\}$;
- 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 $I(u_0)$ »

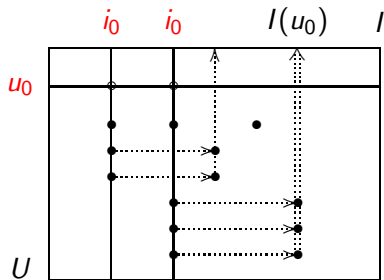


Недостатки:

- 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 $I(u_0)$ »



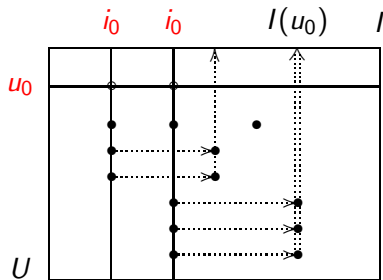
- 1 $I(u_0) := \{i \in I \mid \exists i_0: r_{u_0 i_0} \neq \emptyset \text{ и } B(i) = \text{sim}(i, i_0) > \alpha\}$;
где $\text{sim}(i, i_0)$ — одна из возможных мер сходства i и i_0 ;
- 2 сортировка $i \in I(u_0)$ по убыванию $B(i)$, взять top N ;

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.~~



Непараметрическая регрессия для восстановления пропусков

$$\text{User-based: } \hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in U_\alpha(u)} \text{sim}(u, v)(r_{vi} - \bar{r}_v)}{\sum_{v \in U_\alpha(u)} \text{sim}(u, v)}$$

$$\text{Item-based: } \hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in I_\alpha(i)} \text{sim}(i, j)(r_{uj} - \bar{r}_j)}{\sum_{j \in I_\alpha(i)} \text{sim}(i, j)}$$

\bar{r}_u и \bar{r}_i — средний рейтинг клиента u и объекта i ,

$\text{sim}(u, v)$ и $\text{sim}(i, j)$ — функции близости (u, v) и (i, j) ,

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

$I_\alpha(i) = \{j \mid \text{sim}(i, j) > \alpha\}$ — множество объектов, близких к i .

Функции близости, используемые в корреляционных методах

- корреляция Пирсона:

$$\text{sim}(u, v) = \frac{\sum_{i \in I(u, v)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I(u, v)} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I(u, v)} (r_{vi} - \bar{r}_v)^2}};$$

- косинусная мера близости:

$$\text{sim}(u, v) = \frac{\sum_{i \in I(u, v)} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I(u, v)} r_{ui}^2 \sum_{i \in I(u, v)} r_{vi}^2}};$$

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

- статистические критерии:

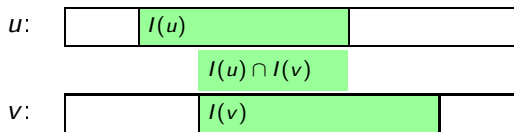
χ^2 , точный тест Фишера (для бинарных данных).

Функции близости на основе точного теста Фишера (FET)

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

Нулевая гипотеза:

клиенты u и v совершают свой выбор независимо.



Вероятность случайной реализации i совместных выборов

$$p(i) = P\{|I(u) \cap I(v)| = i\} = \frac{C_{|I(u)|}^i C_{|I|-|I(u)|}^{|I(v)|-i}}{C_{|I|}^{|I(v)|}}.$$

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

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:

$$\begin{aligned} (p_{tu})_{t \in G} & \text{ — профили клиентов } u \in U, \quad |G| \ll |I|; \\ (q_{ti})_{t \in H} & \text{ — профили объектов } i \in I, \quad |H| \ll |U|. \end{aligned}$$

Типы латентных моделей (основные идеи):

1 Ко-кластеризация:

$$\text{— жёсткая: } \begin{cases} p_{tu} = [\text{клиент } u \text{ принадлежит кластеру } t \in G]; \\ q_{ti} = [\text{объект } i \text{ принадлежит кластеру } t \in H]; \end{cases}$$

— 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;
 $p_{tu} = p(t|u)$, $q_{ti} = q(t|i)$.

Матричные разложения (matrix factorization)

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

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

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

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

Матричная запись: $R = P^T \Delta Q$, $\Delta = \text{diag}(\pi_1, \dots, \pi_{|T|})$;

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

Методы решения:

SVD — сингулярное разложение;

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

PLSA — вероятностный латентный семантический анализ.

Сингулярное разложение (SVD, singular value decomposition)

Постановка задачи SVD: $\|R - P^T \Delta Q\|^2 \rightarrow \min_{P, Q, \Delta}$.

Недостатки:

- если r_{ui} не известно, то полагаем $r_{ui} = 0$
- ортогональность вектор-строк p_t, q_t
- неинтерпретируемость компонент вектор-строк

Достоинства:

- обоснование $r_{ui} = 0$: если клиент u никогда не выбирал объект i , то он ему, скорее всего, не интересен
- высокую оценку \hat{r}_{ui} получают лишь самые интересные
- можно применять готовые библиотеки линейной алгебры
- хорошее ранжирование предложений на некоторых данных

Cremonesi P., Koren Y., Turrin R. Performance of Recommender Algorithms on Top-N Recommendation Tasks. RecSys 2010.

Модель латентных факторов (LFM, Latent Factor Model)

Модификация задачи SVD для случая разреженных данных:

$$\sum_{(u,i) \in D} \underbrace{\left(r_{ui} - \bar{r}_u - \bar{r}_i - \sum_{t \in T} p_{tu} q_{ti} \right)^2}_{\varepsilon_{ui}} \rightarrow \min_{P, Q}$$

Метод стохастического градиента:

перебираем все $(u, i) \in D$ многократно в случайном порядке и делаем каждый раз градиентный шаг для задачи $\varepsilon_{ui}^2 \rightarrow \min_{P_u, Q_i}$

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

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

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

Модель латентных факторов (LFM, Latent Factor Model)

Преимущества метода стохастического градиента:

- легко вводится регуляризация:

$$\varepsilon_{ui}^2 + \lambda \|p_u\|^2 + \mu \|q_i\|^2 \rightarrow \min_{p_u, q_i};$$

- легко вводятся ограничения неотрицательности:

$$p_{tu} \geq 0, \quad q_{ti} \geq 0 \quad (\text{метод проекции градиента});$$

- легко вводится обобщение для ранговых данных:

$$\sum_{(u,i) \in D} \left(r_{ui} - \bar{r}_u - \bar{r}_i - \beta \left(\sum_{t \in T} p_{tu} q_{ti} \right) \right)^2 \rightarrow \min_{P, Q, \beta}.$$

- 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):

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

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

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

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

Cichocki A., Zdunek R., Amari S., Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. 2007

Модель с учётом неявной информации (implicit feedback)

Явные (explicit) предпочтения r_{ui} , более качественные данные:

- покупки товаров в интернет-магазине
- оценки, рейтинги, лайки/дизлайки

Неявные (implicit) предпочтения s_{ui} , большой объём данных:

- посещение страницы товара
- просмотр фильма

Идея: предсказываем s_{ui} с весом $c_{ui} = 1 + \alpha r_{ui}$:

$$\sum_{(u,i) \in D} c_{ui} \left(s_{ui} - \bar{s}_u - \bar{s}_i - \sum_{t \in T} p_{tu} q_{ti} \right)^2 + \lambda \sum_{u \in U} \|p_u\|^2 + \mu \sum_{i \in I} \|q_i\|^2 \rightarrow \min_{P, Q}$$

Модель с неявными предпочтениями победила в Netflix Prize.

Bell R. M., Koren Y., Volinsky C. The BellKor 2008 solution to the Netflix Prize.

Линейная регрессионная модель

$x_{ui} = (x_{ui1}, \dots, x_{uin})$ — вектор признакового описания (u, i)

Примеры признаков:

- для фильмов: текст описания, теги, артисты
- для музыки: жанр, исполнитель, теги
- для событий: текст описания, геолокация, отзывы
- унитарный код (one-hot encoding) клиента u
- унитарный код (one-hot encoding) объекта i

Линейная регрессионная модель для r_{ui} :

$$\hat{r}_{ui} = w_0 + \sum_{j=1}^n w_j x_{uij}.$$

Она не описывает связи между пользователями и объектами.

Квадратичная регрессионная модель

Добавим взаимодействия между признаками:

$$\hat{r}_{ui} = w_0 + \sum_{j=1}^n w_j x_{uij} + \sum_{j=1}^n \sum_{k=1}^n w_{jk} x_{uij} x_{uik}$$

Представим веса w_{jk} низкоранговым матричным разложением:

$$w_{jk} = v_j^T v_k, \quad v_i \in \mathbb{R}^{|T|}.$$

- регулируемое число параметров
- если нет дополнительных признаков, то получаем LFM
- настраивается с помощью SGD или ALS
- наиболее мощный инструмент — библиотека libFM

Steffen Rendle. Factorization machines with libFM. 2012.

Измерение качества рекомендаций

RMSE — точность предсказания рейтингов:

$$\text{RMSE}^2 = \sum_{(u,i) \in D} (r_{ui} - \hat{r}_{ui})^2$$

Точность предсказаний не гарантирует хороших рекомендаций.

$R_u(k) \subset I$ — первые k рекомендаций для u ;

$L_u \subset I$ — истинные предпочтения u .

Более адекватные метрики качества рекомендаций:

- $\text{precision}@k = \frac{|R_u(k) \cap L_u|}{|R_u(k)|}$ — точность
- $\text{recall}@k = \frac{|R_u(k) \cap L_u|}{|L_u|}$ — полнота
- меры качества ранжирования: MAP, NDCG и др.

Измерение качества рекомендаций

Рекомендательные системы отличаются многокритериальностью:

- *Разнообразие* (diversity): например, число рекомендаций из разных категорий или степень различия рекомендаций между сессиями пользователя
- *Новизна* (novelty): сколько среди рекомендаций объектов, новых для пользователя
- *Покрытие* (coverage): доля объектов, которые хотя бы раз побывали среди рекомендованных
- *Догадливость* (serendipity): способность угадывать неожиданные нетривиальные предпочтения пользователей

Можно оптимизировать линейную комбинацию критериев, либо оптимизировать один при ограничениях на остальные.

Оффлайн- и онлайн- измерения качества рекомендаций

Типичная схема эксперимента:

- разбиваем выборку сессий на обучение и тест;
- оптимизируем оффлайн-метрику качества на обучении;
- оцениваем качество на тесте;
- внедряем модель в рекомендательный сервис;
- проводим АВ-тестирование, измеряем онлайн-метрику (деньги или число кликов).

Онлайн- и оффлайн-метрики могут быть слабо связаны:

- в оффлайне не известно, что пользователь мог бы купить
- в оффлайне не известно, что он купил бы без рекомендаций.

Выводы из экспериментов: для улучшения онлайн-точности нужно оптимизировать разные аспекты качества в оффлайне.

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

Корреляционные модели — простые, но устаревшие

Модели латентной семантики обладают рядом преимуществ:

- тематические профили содержательно интерпретируемы,
- оцениваются по внешним данным для «холодного старта»,
- дают адекватные оценки сходства клиентов и объектов,
- позволяют ранжировать рекомендации,
- резко сокращают объём хранимых данных.