

Современные методы построения рекомендательных систем

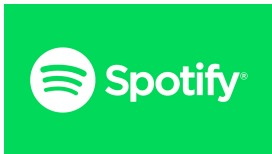
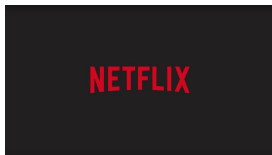
Ефимов В. И.

Введение

- ▶ Рекомендательные системы призваны помочь пользователю с поиском релевантных товаров или других сущностей.
- ▶ Их актуальность обусловлена разнообразием товаров и услуг таким, что пользователь не в состоянии ознакомиться со всем ассортиментом за адекватное время.

Введение

Примерами рекомендательных систем являются предложения товаров в онлайн-магазине, новой музыки в стриминговых сервисах, интересных новостей на различных порталах.



Обзор

Обычно выделяют три вида РС:

1. Content-based systems (CB).
2. Collaborative filtering systems (CF).
3. Hybrid systems.

Подходы:

1. Эвристические методы.
2. Модельные, основанные на матричных разложениях.

Эвристические методы

Матрица взаимодействий. Строки отвечают отдельным пользователям, а столбцы продуктам. На пересечении находятся сами взаимодействия (оценки, просмотры, клики, лайки и тп). Пропуски обозначают отсутствие действия.

1	-1			1	1	-1
1		1		1		-1
-1	1	1				1
-1		-1	1	1		
		-1	-1	-1	-1	

- ▶ Какие особенности у таких матриц?

Эвристические методы

Меры схожести:

1. Косинусное расстояние: $d(x, y) = \frac{(x \cdot y)}{\|x\| \|y\|}$.
2. Корреляция Пирсона: $d(x, y) = \frac{\Sigma(x, y)}{\sigma_x \sigma_y}$.

Обе меры в шкале от -1 до 1, чем ближе к единице, тем более похожими считаются объекты.

Нормализация:

1. Вычитание среднего рейтинга: $h_u(r) = r - \bar{r}_u$.
2. Z-score: $h_u(r) = \frac{r - \bar{r}_u}{\sigma_u}$.

Эвристические методы

Пример расчета мер между первыми двумя строками матрицы:

x	1	-1	0	0	1	1	-1
y	1	0	1	0	1	0	-1

1. Косинусное расстояние:

$$d(x, y) = \frac{(x \cdot y)}{\|x\| \|y\|} = \frac{3}{2\sqrt{5}} \approx 0.671.$$

2. Корреляция Пирсона: $d(x, y) = \frac{\Sigma(x, y)}{\sigma_x \sigma_y} =$

$$= \frac{\sum_{i=1}^7 (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^7 (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^7 (y_i - \bar{y})^2}} = \frac{\frac{19}{7}}{\sqrt{\frac{34}{7}} \sqrt{\frac{24}{7}}} \approx 0.665$$

Эвристические методы

User-based подход:

$$\hat{r}_{ui} = h_u^{-1} \left(\frac{\sum_{v \in \text{NN}_u(\mathcal{U}_i)} \text{sim}(u, v) \cdot h_v(r_{vi})}{\sum_{v \in \text{NN}_u(\mathcal{U}_i)} |\text{sim}(u, v)|} \right)$$

Item-based подход:

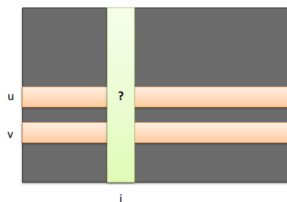
$$\hat{r}_{ui} = h_i^{-1} \left(\frac{\sum_{j \in \text{NN}_i(\mathcal{I}_u)} \text{sim}(i, j) \cdot h_j(r_{uj})}{\sum_{j \in \text{NN}_i(\mathcal{I}_u)} |\text{sim}(i, j)|} \right)$$

sim – мера схожести между объектами, *h* – нормализация, приведение рейтингов к единой шкале.

Классические методы

User-based подход:

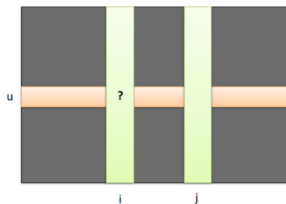
$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in S_{ui}} (r_{vi} - \bar{r}_v) \text{sim}(u, v)}{\sum_{v \in S_{ui}} |\text{sim}(u, v)|}$$



Классические методы

Item-based подход:

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathcal{S}_{ui}} r_{uj} \cdot \text{sim}(i, j)}{\sum_{j \in \mathcal{S}_{ui}} |\text{sim}(i, j)|}$$



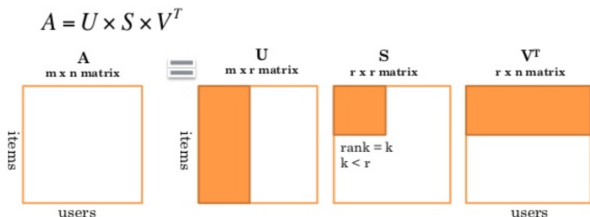
Иногда даже в item-based подходе вычитают среднее именно по пользователю, это называется Adjusted Cosine:

$$\text{sim}(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_u)^2}}$$

SVD модели

Теорема Эккарта-Янга. Для матрицы M матрица M_k с рангом k является наилучшим приближением в смысле нормы Фробениуса разницы этих матриц и определяется как:

$$M_k = US_k V^T.$$



$$A_k = U_k \times S_k \times V_k^T$$

SVD модели

Ищем решение в виде:

$$\hat{r}_{ui} = p_u^T q_i$$

		Item			
		W	X	Y	Z
User	A		4.5	2.0	
	B	4.0		3.5	
	C		5.0		2.0
	D		3.5	4.0	1.0

Rating Matrix

=

A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

User Matrix

X

		W	X	Y	Z
		1.5	1.2	1.0	0.8
1.7	0.6	1.1	0.4		

Item Matrix

SVD модели

Ищем решение в виде:

$$\hat{r}_{ui} = p_u^T q_i$$

		Item			
		W	X	Y	Z
User	A		4.5	2.0	
	B	4.0		3.5	
	C		5.0		2.0
	D		3.5	4.0	1.0

Rating Matrix

=

A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

User Matrix

X

W	X	Y	Z
1.5	1.2	1.0	0.8
1.7	0.6	1.1	0.4

Item Matrix

Оптимизируем целевой функционал:

$$J(P, Q) = \sum_{(u,i) \in \mathcal{K}} (r_{ui} - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2) \rightarrow \min_{P, Q}.$$

SVD модели

SGD ALS

Algorithm 1 SGD ALS для SVD

- 1: Инициализировать p_u и q_i для всех u, i
 - 2: **repeat**
 - 3: **for all** $(u, i) \in \mathcal{K}$ **do**
 - 4: $e_{ui} = r_{ui} - p_u^T q_i$
 - 5: $p_u \leftarrow p_u + 2\eta(e_{ui}q_i - \lambda p_u)$
 - 6: $q_i \leftarrow q_i + 2\eta(e_{ui}p_u - \lambda q_i)$
 - 7: **end for**
 - 8: **until** p_u и q_i меняются больше чем на ϵ или другое условие остановки
-

SVD модели

ALS

Algorithm 2 ALS для SVD

- 1: Инициализировать p_u и q_i для всех u, i
 - 2: **repeat**
 - 3: **for all** $u = 1 \dots m$ **do**
 - 4: $p_u = (\sum_{i:(u,i) \in \mathcal{K}} q_i q_i^T + \lambda I_k)^{-1} (\sum_{i:(u,i) \in \mathcal{K}} r_{ui} q_i)$
 - 5: **end for**
 - 6: **for all** $i = 1 \dots n$ **do**
 - 7: $q_i = (\sum_{u:(u,i) \in \mathcal{K}} p_u p_u^T + \lambda I_k)^{-1} (\sum_{u:(u,i) \in \mathcal{K}} r_{ui} p_u)$
 - 8: **end for**
 - 9: **until** p_u и q_i меняются больше чем на ϵ или другое условие остановки
-

SVD модели

Улучшения модели:

- ▶ μ – среднее по всем парам в данных;
- ▶ b_u – отклонение от среднего для пользователя u ;
- ▶ b_i – отклонение от среднего для объекта i .

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u.$$

SVD модели

Еще улучшения:

- ▶ Неявные предпочтения пользователя:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j).$$

- ▶ Доп информация о пользователе.
- ▶ Изменения предпочтений пользователя со временем.
- ▶ Изменения популярности товаров со временем.

Линейная модель

Признаки x для предсказания r_{ui} :

- ▶ One-hot encoding id пользователя
- ▶ One-hot encoding id объекта
- ▶ Дополнительная информация про пользователя или объект

Feature vector x										Target y	
$x^{(1)}$	1	0	0	...	1	0	0	0	...	5	$y^{(1)}$
$x^{(2)}$	1	0	0	...	0	1	0	0	...	3	$y^{(2)}$
$x^{(3)}$	1	0	0	...	0	0	1	0	...	1	$y^{(3)}$
$x^{(4)}$	0	1	0	...	0	0	1	0	...	4	$y^{(4)}$
$x^{(5)}$	0	1	0	...	0	0	0	1	...	5	$y^{(5)}$
$x^{(6)}$	0	0	1	...	1	0	0	0	...	1	$y^{(6)}$
$x^{(7)}$	0	0	1	...	0	0	1	0	...	5	$y^{(7)}$
	A	B	C	...	TI	NH	SW	ST	...		
	User				Movie						

Линейная модель

Линейная модель:

$$h_{lin}(x) = \omega_0 + \sum_{j=1}^d \omega_j x_j$$

Poly2:

$$h_{poly2}(x) = h_{lin}(x) + \sum_{j_1 < j_2} \omega_{j_1 j_2} x_{j_1} x_{j_2}$$

Factorization Machines

Обобщение прошлых моделей, универсальная модель коллаборативной фильтрации:

$$h(x) = \omega_0 + \sum_{j=1}^d \omega_j x_j + \sum_{j_1=1}^d \sum_{j_2=j_1+1}^d x_{j_1} x_{j_2} v_{j_1}^T v_{j_2}$$

Признаковое описание для обучающих примеров:

Feature vector x														Target y											
x_1	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	y_1			
x_2	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	y_2			
x_3	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	y_3			
x_4	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	y_4			
x_5	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	y_5			
x_6	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	y_6			
x_7	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	y_7			
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...					
	User				Movie				Other Movies rated					Last Movie rated											

Factorization Machines

Важно понимать, что представление (вектор в новом пространстве) строится для каждого признака. Как это работает можно посмотреть на примере LightFM.



<https://github.com/lyst/lightfm>

<https://arxiv.org/abs/1507.08439>

Factorization Machines

Расчет представления:

$$\underbrace{(f_1 \quad f_2 \quad \dots \quad f_n)}_{(1 \times n)} \cdot \underbrace{\begin{pmatrix} e_{1,1} & e_{1,2} & \dots & e_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n,1} & e_{n,2} & \dots & e_{n,k} \end{pmatrix}}_{(n \times k)} = \underbrace{(c_1 \quad c_2 \quad \dots \quad c_k)}_{(1 \times k)}$$

Factorization Machines

Расчет представления:

$$\underbrace{(f_1 \quad f_2 \quad \dots \quad f_n)}_{(1 \times n)} \cdot \underbrace{\begin{pmatrix} e_{1,1} & e_{1,2} & \dots & e_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n,1} & e_{n,2} & \dots & e_{n,k} \end{pmatrix}}_{(n \times k)} = \underbrace{(c_1 \quad c_2 \quad \dots \quad c_k)}_{(1 \times k)}$$

Расчет смещения:

$$\underbrace{(f_1 \quad f_2 \quad \dots \quad f_n)}_{(1 \times n)} \cdot \underbrace{(b_1 \quad b_2 \quad \dots \quad b_n)}_{(1 \times n)} = d$$

Factorization Machines

Расчет представления:

$$\underbrace{(f_1 \quad f_2 \quad \dots \quad f_n)}_{(1 \times n)} \cdot \underbrace{\begin{pmatrix} e_{1,1} & e_{1,2} & \dots & e_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n,1} & e_{n,2} & \dots & e_{n,k} \end{pmatrix}}_{(n \times k)} = \underbrace{(c_1 \quad c_2 \quad \dots \quad c_k)}_{(1 \times k)}$$

Расчет смещения:

$$\underbrace{(f_1 \quad f_2 \quad \dots \quad f_n)}_{(1 \times n)} \cdot \underbrace{(b_1 \quad b_2 \quad \dots \quad b_n)}_{(1 \times n)} = d$$

Расчет оценки для данной пары пользователя и товара:

$$raw_score = emb_{user} \cdot emb_{item} + bias_{user} + bias_{item}$$

Полезные ссылки

- ▶ Ricci F., Rokach L., Shapira B. Introduction to recommender systems handbook //Recommender systems handbook. – Springer, Boston, MA, 2011. – С. 1-35.
- ▶ Sarwar B. et al. Item-based collaborative filtering recommendation algorithms //Proceedings of the 10th international conference on World Wide Web. – 2001. – С. 285-295.
- ▶ Rendle S. Factorization machines //2010 IEEE International Conference on Data Mining. – IEEE, 2010. – С. 995-1000.
- ▶ Rendle S. et al. BPR: Bayesian personalized ranking from implicit feedback //arXiv preprint arXiv:1205.2618. – 2012.
- ▶ Kula M. Metadata embeddings for user and item cold-start recommendations //arXiv preprint arXiv:1507.08439. – 2015.

Спасибо за внимание!