

2. ЛЕКЦИЯ. Модельные методы

Несмотря на легкость понимания и реализации анамнестических методов, у них есть ряд недостатков. Во-первых, возникают трудности при прогнозе предпочтений для новых пользователей или при появлении новых элементов, т.к. для них еще нет оценок («холодный старт»). Во-вторых, для расчета предпочтений необходимо хранить всю матрицу данных. В-третьих, ограничивается возможность методов при обработке больших объемов данных, т.к. выполнение большого количества операций для вычисления степени схожести затрудняет выдачу рекомендаций в реальном времени. Большой объем матрицы предпочтений затрагивает также проблему избыточности данных. Как правило, пользователи и элементы делятся на группы с аналогичными профилями предпочтений.

Вследствие этого, возникает задача в понижении размерности матрицы оценок. Такие задачи решают методы второй группы модельные методы (Model-based). Эти методы направлены на поиск закономерностей на основе обучающих данных. Такой подход является комплексным и даёт более точные прогнозы, так как помогает раскрыть скрытые факторы, объясняющие наблюдаемые оценки

В этом случае возможен вариант объединения пользователей (элементов) в кластеры (профили) с помощью некоторого индекса сходства. Элементы и оценки, выставленные пользователями из одного кластера, будут использоваться для вычисления рекомендаций. Кластерные модели лучше масштабируются, т.к. сверяют профиль пользователя с относительно небольшим количеством сегментов, а не с целой пользовательской базой.

Эта задача может быть выполнена на основе разных математических подходов.

Сингулярное разложение

Для уменьшения сложности вычисления степени схожести векторов объектов используется подход понижения размерности матрицы, который основан на разложении матрицы по сингулярным значениям. Это один из методов матричной факторизации.

Суть метода сингулярного разложения матрицы (Singular Value Decomposition, SVD) $A \in M(n, m)$ с рангом $d = \text{rank}(M) \leq \min(n, m)$ в произведение матриц меньшего ранга:

$$A = UDV^T \quad (2.1)$$

где матрицы $U \in M(n, n)$ и $V \in M(m, m)$ состоят из ортонормальных столбцов, являющихся собственными векторами при ненулевых собственных

значениях матрицы AA^T и $A^T A$.

Напомним:

Ортогональная матрица – квадратная матрица A с вещественными элементами, результат умножения которой на транспонированную матрицу A^T равен единичной матрице:

$$AA^T = A^T A = I$$

или, что эквивалентно, её обратная матрица (которая обязательно существует) равна транспонированной матрице:

$$A^{-1} = A^T$$

Формулу сингулярного разложения $A = UDV^T$ можно переформулировать в геометрических терминах (рис.2.1). Линейный оператор, отображающий элементы пространства \mathbb{R}^m в элементы пространства \mathbb{R}^n представим в виде последовательно выполняемых линейных операций вращения, масштабирования (растяжения) и вращения. Число ненулевых элементов на диагонали матрицы D есть фактическая размерность матрицы A .

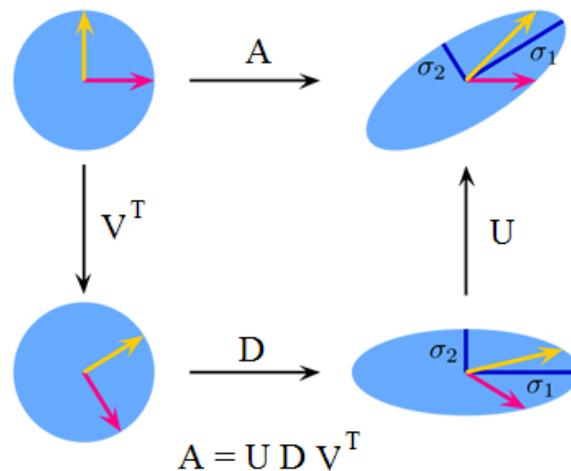


Рис. 2.1. Геометрический смысл сингулярного разложения

Вверху слева (рис. 2.1) видно, единичный диск (синий цвет) с двумя каноническими единичными векторами. Действие V^T (внизу слева) на единичном диске - это просто вращение (против часовой стрелки) на угол α . Действие D (справа внизу) масштабирует единичный круг по вертикали и горизонтали (вдоль осей координат). Действие U приводит к окончательному повороту с указанными сингулярными значениями σ_1 и σ_2 . Длины σ_1 и σ_2 из полуосей эллипса являются сингулярные значения из A .

Соответственно:

$$U \times U^T = I_n \text{ и } V \times V^T = I_m \quad (2.2)$$

Столбцы матрицы U представляют собой, ортонормальный базис

пространства столбцов матрицы A , а столбцы матрицы V – ортонормальный базис пространства строк матрицы A . I_n и I_m – единичные вектора. $D \in M(n, m)$ – диагональная матрица с положительными диагональными элементами $\sigma_1 \geq \sigma_2 \dots \geq \sigma_d \geq 0$, отсортированного в порядке убывания (рис. 2.2).

$$D = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \sigma_d \end{pmatrix} \quad (2.3)$$

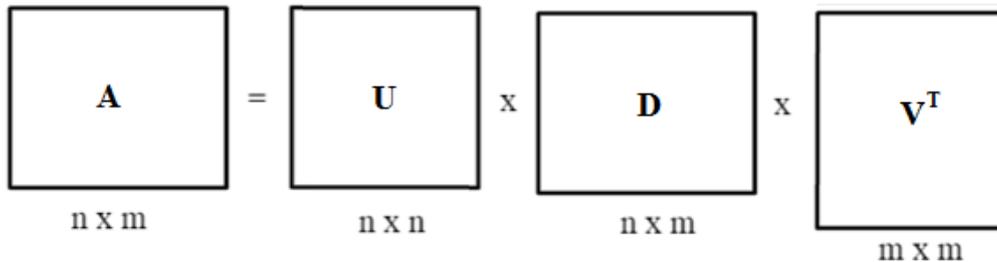


Рис. 2.2. SVD – сингулярное разложение

С использованием свойств (2), можно получить следующие соотношения:

$$AA^T U_i = \sigma_i^2 U_i; i = 1, \dots, m \quad (2.4)$$

где U_i – столбцы матрицы U , σ_i^2 – квадраты соответствующих значений на диагонали матрицы D .

Векторно-матричное выражение (2.3) представляет собой решение задачи нахождения собственных чисел σ_i^2 и собственных векторов U_i матрицы AA^T . В результате получаем, что сингулярные числа $\sigma_i^2 (i = 1, \dots, m)$ представляют собой решения характеристического уравнения

$$\det(\sigma_i^2 I - AA^T) = 0 \quad (2.5)$$

Также можно получить зеркальные соотношения:

$$A^T A V_i = \sigma_i^2 V_i; i = 1, \dots, m \quad (2.6)$$

где V_i – столбцы матрицы V , σ_i^2 – квадраты соответствующих значений на диагонали матрицы D .

В свою очередь, по аналогии с (4), векторно-матричное выражение (2.6) представляет собой полное решение задачи нахождения собственных чисел σ_i^2 и собственных векторов V_i матрицы $A^T A$. В результате получаем, что $\sigma_i^2 (i = 1, \dots, m)$ представляют собой решения характеристического уравнения

$$\det(\sigma_i^2 I - A^T A) = 0 \quad (2.7)$$

При конструировании сингулярного разложения произвольной (n, m) -матрицы A ранга d следует придерживаться следующего порядка действий.

1. Составить матрицу $A^T A$ и характеристический полином этой матрицы. Найти сингулярные числа матрицы A и составить (n, m) -матрицу D .

Рассмотрим следующий пример:

Пусть имеется таблица 2.1 с тестом мужчин о предпочтении к музыкальным направлениям. Индексы строк – это имена меломанов, а индексы столбцов – виды направлений в музыке.

Таблица 2.1

	классика	джаз	регги	рок
Рома	1	1	0	1
Витя	0	1	1	1
Вова	0	0	0	1

Задача состоит в том, чтобы аппроксимировать значения ячеек, где стоят нули. Иными словами необходимо получить новые знания о данных по предложенному тесту.

Составим сингулярное разложение для матрицы:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Решение. Для матрицы A имеем:

$$A^T A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}$$

Составим характеристическое уравнение. Записываем определитель матрицы, вычитая при этом « λ » из чисел главной диагонали:

$$\begin{vmatrix} 1 - \lambda & 0 & 0 & 1 \\ 0 & 1 - \lambda & 0 & 1 \\ 0 & 0 & 1 - \lambda & 1 \\ 1 & 1 & 1 & 3 - \lambda \end{vmatrix} = 0$$

Определитель представляет собой характеристический многочлен. В раскрытом виде он записывается так:

$$\lambda^4 - 6\lambda^3 + 9\lambda^2 - 4\lambda = 0$$

Представим альтернативную запись уравнения, разбив на множители.

$$\lambda(\lambda - 4)(\lambda - 1)^2 = 0$$

Тогда корни уравнения:

$$\lambda_1 = 4; \lambda_2 = \lambda_3 = 1; \lambda_4 = 0$$

Исходя из уравнения 2.7

$$\sigma_1 = \sqrt{\lambda_1} = 2; \sigma_2 = \sigma_3 = \sqrt{\lambda_2} = \sqrt{\lambda_3} = 1; \sigma_4 = 0.$$

Тогда

$$D = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

2. Построить ортонормированную систему собственных векторов e_1, e_2, \dots, e_n оператора с матрицей $A^T A$, принадлежащих соответственно собственным значениям $\lambda_1, \lambda_2, \lambda_3, \lambda_4$. Из столбцов координат этих векторов составить матрицу V .

Составим систему уравнений из строк характеристического уравнения.

$$(A^T A - \lambda I)x_n = 0$$

Тогда

$$\begin{cases} (1 - \lambda)x_1 + x_4 = 0 \\ (1 - \lambda)x_2 + x_4 = 0 \\ (1 - \lambda)x_3 + x_4 = 0 \\ x_1 + x_2 + x_3 + (3 - \lambda)x_4 = 0 \end{cases}$$

При $\lambda_1 = 4$ получаем

$$\begin{cases} -3x_1 + x_4 = 0 \\ -3x_2 + x_4 = 0 \\ -3x_3 + x_4 = 0 \\ x_1 + x_2 + x_3 - x_4 = 0 \end{cases}$$

Придавая переменным « x_n » произвольные значения, получаем бесконечно много собственных векторов. Все они будут коллинеарны друг другу, и поэтому нам достаточно указать один из них.

$$\bar{u} = (x_1, x_2, \dots, x_n)^T$$

Пусть $x_1 = 1, x_2 = 1, x_3 = 1$, тогда $x_4 = 3$. Таким образом, находим первый собственный вектор:

$$\bar{u}_1 = (1, 1, 1, 3)^T$$

При $\lambda_2 = \lambda_3 = 1$ получаем

$$\begin{cases} x_4 = 0 \\ x_4 = 0 \\ x_4 = 0 \\ x_1 + x_2 + x_3 + 2x_4 = 0 \end{cases}$$

Здесь $x_4 = 0$. Тогда возможны следующие варианты собственных векторов:

$$\bar{u}_2 = (1, -1, 0, 0)^T; \bar{u}_3 = (1, 1, -2, 0)^T$$

При $\lambda_4 = 0$ получаем

$$\begin{cases} x_1 + x_4 = 0 \\ x_2 + x_4 = 0 \\ x_3 + x_4 = 0 \\ x_1 + x_2 + x_3 + 3x_4 = 0 \end{cases}$$

Пусть $x_1 = 1, x_2 = 1, x_3 = 1$, тогда $x_4 = -1$. Таким образом, находим последний собственный вектор:

$$\bar{u}_4 = (1, 1, 1, -1)^T$$

Нормируем все полученные вектора, согласно формуле:

$$e_n = \frac{u_n}{\|u_n\|}$$

Тогда используя норма Фробениуса (норма Фробениуса, или евклидова норма представляет собой частный случай нормы для $p = 2$):

$$e_1 = \frac{1}{\sqrt{1^2 + 1^2 + 1^2 + 3^2}} (1, 1, 1, 3)^T = \frac{1}{2\sqrt{3}} (1, 1, 1, 3)^T$$

$$e_2 = \frac{1}{\sqrt{1^2 + (-1)^2 + 0^2 + 0^2}} (1, -1, 0, 0)^T = \frac{1}{\sqrt{2}} (1, -1, 0, 0)^T$$

$$e_3 = \frac{1}{\sqrt{1^2 + 1^2 + (-2)^2 + 0^2}} (1, 1, -2, 0)^T = \frac{1}{\sqrt{6}} (1, 1, -2, 0)^T$$

$$e_4 = \frac{1}{\sqrt{1^2 + 1^2 + 1^2 + (-1)^2}} (1, 1, 1, -1)^T = \frac{1}{2} (1, 1, 1, -1)^T$$

Из столбцов координат векторов e составим матрицу

$$V = \begin{pmatrix} \frac{1}{2\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \frac{1}{2} \\ \frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \frac{1}{2} \\ \frac{1}{2\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} & \frac{1}{2} \\ \frac{3}{2\sqrt{3}} & 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

3. Построить ортонормированную систему векторов

$$f_1 = \frac{Ae_1}{\sigma_1}, \dots, f_r = \frac{Ae_r}{\sigma_r}$$

и дополнить ее любыми векторами до ортонормированного базиса пространства. Из столбцов координат векторов составить матрицу U .

Далее строим векторы

$$f_1 = \frac{Ae_1}{\sigma_1} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \frac{1}{2\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 3 \end{pmatrix} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$f_2 = \frac{Ae_2}{\sigma_2} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

$$f_3 = \frac{Ae_3}{\sigma_3} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ -2 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}$$

Из столбцов координат векторов f_1, f_2, f_3 , составим матрицу.

$$U = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \end{pmatrix}$$

4. Записать искомое сингулярное разложение. $A = UDV^T$

$$A = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{3}{2\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

Важным свойством SVD-разложения является тот факт, что если для $k < d$ преобразовать матрицу D в матрицу

$$D_k = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \sigma_k \end{pmatrix} \in M(k, k)$$

состоящую только из k наибольших диагональных элементов, а также оставить в матрице U и V только k первых столбцов, т.е. преобразовать их в $U_k \in M(n, k)$ и $V_k \in M(m, k)$, то матрица (рис. 2.3)

$$A_k = U_k D_k V_k^T$$

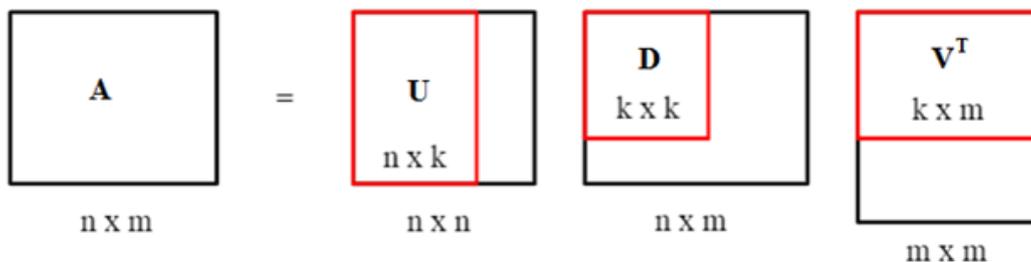


Рис. 2.3. Аппроксимация сингулярного разложения

будет являться лучшей аппроксимации матрицы A относительно нормы Фробениуса среди всех матриц с рангом k , т.е. $\|A - A_k\| \leq \|A - A'\| \forall A' \in M(n, m) \text{ rank}(A') = k$

Данные факторы являются как раз характеристикой вкусов и предпочтений пользователя – для матрицы U и характеристикой самих объектов – для матрицы V .

Это усечение одновременно достигает двух целей. Во-первых, оно уменьшает размерность векторного пространства, снижает требования хранения и вычислительные требования к модели. Во-вторых, отбрасывая малые сингулярные числа, малые искажения в результате шума в данных удаляются, оставляя только самые сильные эффекты и тенденции в этой модели. Снижение воздействия шума улучшает способность предоставлять высококачественные рекомендации.

Тогда зафиксируем некоторое число скрытых факторов $k = 2$ (для нашего примера), которое, так или иначе, описывает каждый элемент и предпочтения каждого меломана относительно этих факторов. Можно подбирать k , исходя из размера сингулярных значений матрицы, т.е. тех самых диагональных элементов матрицы D_k : желательно отбрасывать как можно более маленькие значения элементов.

$$D_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

Сингулярные значения показывают значимость факторов (табл. 2.1)

	f_1	f_2
f_1	2	0
f_2	0	1

Теперь рассмотрим матрицу V_2^T .

$$V_3^T = \begin{pmatrix} \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{3}{2\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \end{pmatrix} \approx \begin{pmatrix} 0,288 & 0,288 & 0,288 & 0,866 \\ 0,707 & -0,707 & 0 & 0 \end{pmatrix}$$

Так как определено 2 фактора – это будут индексы строк, а индексы столбцов относятся к направлениям в музыке (табл.).

Таблица 2.2

	классика	джаз	регги	рок
f_1	0,288	0,288	0,288	0,866
f_2	0,707	-0,707	0	0

Каждый фактор (строка) состоит из 2 значений, каждое из которых определено принадлежностью (весом) к каждой из 4 музыкальных направлений. Логично будет судить фактор за максимальный вес по направлениям. Таким образом, f_1 имеет максимальное значение 0,866 в секторе

«рок», $f_2 = 0,707$ в секторе «классика».

Тогда для понимания можно факторам дать название, которое будет обусловлено психологией музыкальных предпочтений. Многие психологи утверждают, что предпочтения в направлении музыкальных композиций определяют характер человека. Тогда исходя из данной информации можно охарактеризовать человека с иных позиций:

f_1 – склонность к творчеству (Любители рока вопреки бытующему мнению, они обладают чрезвычайно мягким и даже утонченным характером. Это творческие люди, но, как правило, с довольно низкой самооценкой);

f_2 –целеустремленность (Люди, которые предпочитают слушать классическую музыку постоянно стремятся вырваться вперед, любят работать над собой в различных сферах жизни);

Таким образом, помимо рекомендаций мы «бонусом» получили выполненную задачу классификации без учителя.

Теперь рассмотрим матрицу U_2 :

$$U_3 = \begin{pmatrix} 1 & 1 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & 0 \end{pmatrix} \approx \begin{pmatrix} 0,577 & 0,707 \\ 0,577 & -0,707 \\ 0,577 & 0 \end{pmatrix}$$

На основании матрицы составим таблицу 2.3

Таблица 2.3

	f_1	f_2
Рома	0,577	0,707
Витя	0,577	-0,707
Вова	0,577	0

В общем случае рекомендательных систем получается, что мы представляем каждого пользователя вектором из f факторов U_i , а каждый продукт – вектором из f факторов V_j . Представляем каждого пользователя вектором из k факторов r_u и каждый продукт вектором из k факторов r_i , чтобы предсказать рейтинг пользователя u товару i , вычисляем их скалярное произведение:

$$\hat{r}_{u,i} = r_i \times r_u = r_i^T \times r_u$$

Вектор факторов пользователя показывает, насколько пользователю нравится или не нравится тот или иной фактор, а вектор факторов продукта показывает, насколько тот или иной фактор в продукте выражен.

Тогда в нашей задаче рассмотрим, как каждый исследуемый относится к классической музыке. Узнаем, насколько выражена целеустремленность в каждом рассматриваемом человеке:

Рассчитаем скалярное произведение векторов Классика/Рома

$$\begin{aligned} \dot{r}_{1,1} &= (0,288 \quad 0,707)^T (0,577 \quad 0,707) \\ &= 0,288 \times 0,577 + 0,707 \times 0,707 = 0,166 + 0,5 = 0,666 \end{aligned}$$

$$\begin{aligned} \dot{r}_{2,1} &= (0,288 \quad -0,707)^T (0,577 \quad 0,707) \\ &= 0,288 \times 0,577 - 0,707 \times 0,707 = 0,166 - 0,5 = -0,334 \end{aligned}$$

$$\dot{r}_{3,1} = (0,288 \quad 0)^T (0,577 \quad 0,707) = 0,288 \times 0,577 = 0,166$$

$$\dot{r}_{4,1} = (0,866 \quad 0)^T (0,577 \quad 0,707) = 0,866 \times 0,577 = 0,511$$

Роман явно является приверженцем к классической музыке и определен фактором f_2 –целеустремленность, но ему, возможно, нравится направление рок – f_1 (склонность к творчеству).

Аналогично, рассматриваются остальные скалярные произведения векторов. При этом некоторые факторы легко будет понять человеческим разумом, которые будут показывать, насколько соответствующие характеристики объекта по вкусу пользователей. Но может и не выделиться ничего содержательного – гарантий нет, формально просто используется линейная алгебра.

Составим матрицу:

$$A_2 \approx \begin{pmatrix} 0,666 & -0,334 & 0,166 & 0,511 \\ -0,334 & 0,666 & 0,166 & 0,511 \\ 0,166 & 0,166 & 0,166 & 0,511 \end{pmatrix}$$

Тогда составим табл.2.4

Таблица 2.4

	классика	джаз	регги	рок
Рома	0,666	-0,334	0,166	0,511
Витя	-0,334	0,666	0,166	0,511
Вова	0,166	0,166	0,166	0,511

Таким образом, классическая музыка может быть предложена Роману с долей успеха 0,666 и Владимиру джаз с долей успеха 0,666. Аналогичные рассуждения можно привести по другим направлениям музыки.

Однако, в реальной системе подобная матрица будет иметь относительно небольшое число ненулевых элементов (сильноразреженная) – это значит, что далеко не все пользователи оценили объекты, и, соответственно, не все объекты будут оценены, т.е. большая часть оценок заранее неизвестна. Кроме того,

алгоритм SVD-разложения является довольно трудоёмким, поэтому невозможно производить такие вычисления на больших объёмах.

Измерение качества рекомендаций

Существует достаточное количество метрик (оценка качества) для тестирования рекомендательных систем. На данный момент не существует единой метрики. Каждый тестировщик подбирает метрику под свои цели.

Для тестирования качества прогнозирования оценок в рекомендательных системах используются метрики:

MAE – Mean Absolute Error – средняя абсолютная ошибка (2.8)

$$MAE = \frac{1}{|D|} \sum_{(u,i) \in D} |\hat{r}_{ui} - r_{ui}| \quad (2.8)$$

MSE – Mean Squared Error – среднеквадратичная ошибка (2.9)

$$MSE = \frac{1}{|D|} \sum_{(u,i) \in D} (\hat{r}_{ui} - r_{ui})^2 \quad (2.9)$$

RMSE – Root Mean Squared Error – корень среднеквадратичной ошибки (2.10)

$$RMSE = \sqrt{\frac{1}{|D|} \sum_{(u,i) \in D} (\hat{r}_{ui} - r_{ui})^2} \quad (2.10)$$

где для всех формул:

$|D|$ – размер набора данных;

\hat{r}_{ui} – предсказанная оценка пользователя u фильму i ;

r_{ui} – действительная оценка, поставленная пользователем u фильму i .

Эти метрики показывают, насколько ошибается модель при прогнозировании оценок.

Алгоритм Funk SVD

Множество записей в матрице предпочтений могут отсутствовать, и обычный SVD имеет следующие проблемы:

а) пропущенные значения могут оказать нежелательное влияние на результат.

б) вычислительная сложность для обучения может быть высокой со всеми рассмотренными записями.

Поскольку SVD может достичь очень хорошего результата только на разреженных данных (имеются в виду данные без пропущенных значений). Фактически, реальный набор данных для механизма рекомендаций может быть очень разреженным, можно найти матрицы пользовательских элементов со

значениями NaN (*англ. Not-a-Number*, «не число» — одно из особых состояний числа с плавающей запятой) более 95%.

Для этих задач Саймон Фанк придумал регуляризованный SVD алгоритм, который на конкурсе Netflix Prize занял одно из призовых мест. В алгоритме Funk SVD (Funk MF – Matrix Factorization, Факторизация матрицы) не применяется разложение по сингулярным значениям, это подобие SVD-модели машинного обучения. Прогнозируемые рейтинги могут быть вычислены как

$$\tilde{R} = HW \quad \tilde{R} \in \mathbb{R}^{users \times items}$$

где $H \in \mathbb{R}^{users \times latent\ factors}$ – матрица рейтингов пользовательских элементов, содержащая скрытые факторы пользователя и $W \in \mathbb{R}^{latent\ factors \times items}$ – скрытые факторы элемента.

Прогнозируемая оценка, которую пользователь u поставит элементу i , вычисляется как:

$$\tilde{r}_{ui} = \sum_{f=0}^{n\ factors} H_{u,f} W_{f,i}$$

Необходимо учитывать, что увеличение количества скрытых факторов улучшит персонализацию, следовательно, качество рекомендаций, до тех пор, пока количество факторов не станет слишком большим, после чего модель начнет переобучаться, и качество рекомендаций снизится. Распространенной стратегией, позволяющей избежать переобучения, является добавление членов регуляризации к целевой функции. Алгоритм Саймона Фанка был разработан как задача прогнозирования рейтингов, поэтому он использует явные числовые рейтинги в качестве взаимодействий между пользователем и элементом.

Учитывая все обстоятельства, алгоритм минимизирует следующую целевую функцию:

$$\arg \min_{H,W} \|R - \tilde{R}\|_F + \alpha \|H\| + \beta \|W\|$$

где $\|R - \tilde{R}\|_F$ определяется как норма Фробениуса (евклидова норма представляет собой частный случай нормы для $p = 2$). Но могут использоваться другие нормы в зависимости от конкретной рекомендуемой задачи.

Рассмотрим пример реализации алгоритма Саймона Фанка, который должен проигнорировать пропущенные значения и найти способ вычисления скрытых факторов только с использованием известных нам значений.

Пусть имеется матрица User-Item (Пользователь-Объект) табл. 2.5.

Таблица 2.5

	O1	O2	O3	O4
П1	NaN	NaN	9	1
П2	3	NaN	7	NaN
П3	5	NaN	NaN	10
П4	NaN	2	NaN	NaN

Чтобы достичь результата в подходе матричной факторизации с Funk SVD, необходимо выполнить следующие шаги:

1. Построить матрицы U и V^T , соответственно матрицу пользователей по количеству выбранных скрытых факторов (F) и матрицу этих же скрытых факторов по объекту. Заполните эти матрицы случайными числами.

Таблица 2.6. Матрица U

(Пользователь по скрытым факторам, заполненный случайными значениями)

	F1	F2	F3
П1	0,8	1,2	-0,2
П2	2	1,8	0,4
П3	0,8	3	0,1
П4	1	0,8	2,4

Таблица 2.7. Матрица V^T

(Скрытые факторы по объектам, заполненным случайными значениями)

	O1	O2	O3	O4
F1	-1,8	1	-0,2	1
F2	0,5	1,2	0,1	5
F3	1,4	4	0,14	2

2. Найти в матрице «Пользователь-Объект» существующую ставку для одной пары «пользователь-объект».

Таблица 2.8

	O1	O2	O3	O4
П1	NaN	NaN	9	1
П2	3	NaN	7	NaN
П3	5	NaN	NaN	10
П4	NaN	2	NaN	NaN

Первая найденная оценка – 9, присвоенная пользователем 1 пункту 3. Итак, 9 – это наше фактическое значение, истинное значение.

3. В матрице U берутся все случайные значения, связанные с

пользователем П1 (первая строка, табл. 2.9). У нас есть для этого пользователя [0.8, 1.2, -0.2]

Таблица 2.9

	F1	F2	F3
П1	0,8	1,2	-0,2
П2	2	1,8	0,4
П3	0,8	3	0,1
П4	1	0,8	2,4

Помните, что 9 было присвоено пользователем 1 элементу 3. В матрице V^T берутся все случайные значения, связанные с элементом 3. У нас есть для этого элемента $[-0,2, 0,1, 0,14]^T$ (табл. 2.10).

Таблица 2.10

	O1	O2	O3	O4
F1	-1,8	1	-0,2	1
F2	0,5	1,2	0,1	5
F3	1,4	4	0,14	2

4. Вычисляется скалярное произведение между найденной строкой и столбцом, чтобы сделать наш прогноз.

$$r_{13} = u_1 v_3 = (0,8 \times -0,2) + (1,2 \times 0,1) + (-0,2 \times 0,14) = -0,07$$

Имеем: Фактическое $\hat{r}_{13} = 9$; Прогнозируемый $r_{13} = -0,07$

Таким образом, среднеквадратичная ошибка составляет:

$$e_{13} = (\hat{r}_{13} - r_{13})^2 = 0,5 \times (9 - (-0,07))^2 = 41,13$$

5. Необходимо минимизировать ошибку, используя градиентный спуск.

Для пользователя:

$$U_i^* = U_i + \alpha \nabla e_i V_i = U_i + \alpha \nabla (\hat{r}_i - r_i)^2 V_i = U_i + \alpha 2(\hat{r}_i - r_i) V_i$$

где

U_i^* – обновленное значение пользователя по скрытым факторам;

U_i – случайное значение пользователя по скрытым факторам;

∇e_i – градиент функции ошибки;

V_i – случайное значение скрытых факторов по объектам.

Для объектов:

$$V_i^* = V_i + \alpha 2(\hat{r}_i - r_i) U_i$$

где

V_i^* – обновленное значение скрытых факторов по объектам;

V_i – случайное значение скрытых факторов по объектам;

∇e_i – градиент функции ошибки;

U_i – случайное значение пользователя по скрытым факторам/

Составим таблицу 2.11 для каких значений необходимо сделать обновления.

Таблица 2.11

	F1	F2	F3
для U	0,8	1,2	-0,2
для V	-0,2	0,1	0,14

Обновим значение пользователя по скрытым факторам при $\alpha = 0,1$:

$$U_1^* = U_1 + \alpha 2(\hat{r}_{13} - r_{13})V_1 = 0,8 + 0,1 \times 2 (9 + 0,07) \times -0,2 = 0,44$$

$$U_2^* = U_2 + \alpha 2(\hat{r}_{13} - r_{13})V_2 = 1,2 + 0,1 \times 2 (9 + 0,07) \times 0,1 = 1,38$$

$$U_3^* = U_3 + \alpha 2(\hat{r}_{13} - r_{13})V_3 = -0,2 + 0,1 \times 2 (9 + 0,07) \times 0,14 = 0,053$$

Обновим значение скрытых факторов по объектам

$$V_1^* = V_1 + \alpha 2(\hat{r}_{13} - r_{13})U_1 = -0,2 + 0,1 \times 2 (9 + 0,07) \times 0,8 = 1,25$$

$$V_2^* = V_2 + \alpha 2(\hat{r}_{13} - r_{13})U_2 = 0,1 + 0,1 \times 2 (9 + 0,07) \times 1,2 = 2,28$$

$$V_3^* = V_3 + \alpha 2(\hat{r}_{13} - r_{13})U_3 = 0,14 + 0,1 \times 2 (9 + 0,07) \times -0,2 = -0,22$$

Обновляются все значения из U и V таблицы 2.11.

Таблица 2.12

	F1	F2	F3
для U	0,44	1,38	0,053
для V	1,25	2,28	-0,22

Соответственно, исправляются значения для матрицы пользователя U табл. 2.13.

Таблица 2.13

	F1	F2	F3
П1	0,44	1,38	0,053
П2	2	1,8	0,4
П3	0,8	3	0,1
П4	1	0,8	2,4

И заменяются значения для матрицы объектов V^T табл. 2.14.

Таблица 2.14

	O1	O2	O3	O4
F1	-1,8	1	1,25	1
F2	0,5	1,2	2,28	5
F3	1,4	4	-0,22	2

6. Снова вычисляем скалярное произведение между новой строкой и

столбцом, чтобы сделать обновленный прогноз.

$$r_{13}' = u_1'v_3' = (0,44 \times 1,25) + (1,38 \times 2,28) + (0,053 \times -0,22) \\ = 3,58$$

Тогда имеем: Фактическое $\hat{r}_{13} = 9$; Прогнозируемое $r_{13}' = 3,58$

Таким образом, среднеквадратичная ошибка составляет:

$$e_{13} = (\hat{r}_{13} - r_{13}')^2 = 0,5 \times (9 - 3,58)^2 = 14,68$$

Таким образом, за одну итерацию величина ошибки уменьшилась на $41,13 - 14,68 = 26,45$.

Таким образом, алгоритм Funk SVD легко позволяет найти способ оценить механизм рекомендаций и создать хорошие матрицы U и V^T , чтобы давать рекомендации, даже если у нас очень разреженная матрица. Но алгоритм Funk SVD не следует использовать в одиночку, потому что можно столкнуться с общей проблемой в системе рекомендаций, которая называется «Проблема холодного запуска». Эта проблема означает, что нельзя давать рекомендации новым пользователям или новым объектам.

Алгоритм SVD++

Funk SVD может обеспечить очень хорошее качество рекомендаций, его способность использовать только явные числовые рейтинги в качестве взаимодействий между пользователями и элементами представляет собой ограничение. Современные рекомендательные системы должны использовать все доступные взаимодействия, как явные (например, числовые рейтинги), так и неявные (например, лайки, покупки, пропущенные, добавленные в закладки). С этой целью SVD ++ также был разработан с учетом неявных взаимодействий, учитывает предвзятость пользователя и элемента.

Пусть стоит задача: по известным оценкам известных продуктов предсказать, насколько хорошо будет оценен новым пользователем каждый продукт.

Введем так называемые базовые предикторы $b_{i,a}$, которые складываются из базовых предикторов отдельных пользователей b_i , и отдельных продуктов b_a , а также просто общего среднего рейтинга по базе μ :

$$b_{i,a} = \mu + b_i + b_a,$$

где μ – средний рейтинг по базе; b_i – средний рейтинг каждого i пользователя; b_a – средний рейтинг каждого a продукта.

Для определения только базовых предикторов необходимо найти такие μ , b_i , b_a , для которых $b_{i,a}$ лучше всего приближают имеющиеся рейтинги. Затем можно будет добавить собственно факторы. Поскольку теперь, когда сделана

поправка на базовые предикторы, остатки будут сравнимы между собой, вполне возможно будет получить разумные значения для факторов:

$$\hat{r}_{ui} = \mu + b_i + b_a + v_a^T u_i \quad (2.11)$$

где v_a – вектор факторов, представляющий продукт a ; u_i – вектор факторов, представляющий пользователя i .

Теперь можно вернуться к исходной задаче и сформулировать ее точно: нужно найти наилучшие предикторы, которые приближают величину \hat{r}_{ui} . Лучшими будут те предикторы, которые дают минимальную среднеквадратичную ошибку, определяемую следующим образом:

$$L(\mu, b_i, b_a, v_a, u_i) = \sum_{(i,a) \in D} (r_{i,a} - \hat{r}_{i,a})^2 = \sum_{(i,a) \in D} (r_{i,a} - \mu - b_i - b_a - v_a^T u_i)^2$$

Функцию $L(\mu, b_i, b_a, v_a, u_i)$ минимизируем градиентным спуском: берем частные производные по каждому аргументу и двигаемся в сторону, обратную направлению этих частных производных.

Для компенсации эффекта переобучения добавляется параметр регуляризации. Иными словами, накладывается штраф за слишком большие значения обучаемых переменных. Например, можно просто добавить в функцию ошибки сумму квадратов всех факторов и предикторов. В результате функция ошибки выглядит как

$$b_*, v_*, u_* = \arg \min_{b,p,g} \sum_{(i,a)} (r_{i,a} - \mu - b_i - b_a - v_a^T u_i)^2 + \lambda (\sum_i b_i^2 + \sum_a b_a^2 + \|v_a\|^2 + \|u_i\|^2), \quad (2.12)$$

Если взять у функции ошибки в формуле (2) частные производные по каждой из оптимизируемых переменных, получим простые правила для градиентного (стохастического) спуска:

$$\begin{aligned} b_i &= b_i + \gamma(e_{i,a} - \lambda b_i), \\ b_a &= b_a + \gamma(e_{i,a} - \lambda b_a), \\ v_{a,j} &= v_{a,j} + \gamma(e_{i,a} u_{i,j} - \lambda v_{a,j}), \\ u_{i,j} &= u_{i,j} + \gamma(e_{i,a} v_{a,j} - \lambda u_{i,j}) \end{aligned}$$

для всех j , где $e_{i,a} = r_{i,a} - \hat{r}_{i,a}$, a — ошибка на данном тестовом примере, а γ — скорость обучения.