

# Обучение без учителя. Разделение смеси распределений. Кластеризация. Тематическое обучение (Probabilistic LSA)

Оленев Роман, Самарин Игорь

Санкт-Петербургский государственный университет  
Кафедра статистического моделирования

Санкт-Петербург, 2024

# Обучение без учителя

Обучение без учителя - это один из способов машинного обучения, при котором испытываемая система обучается выполнять поставленную задачу без вмешательства со стороны экспериментатора. Как правило, это пригодно только для задач, в которых известны описания множества объектов (обучающей выборки), и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами.

Обучение без учителя становится формально поставленной задачей только в случае, если можно написать функцию правдоподобия.

# Кластеризация. Введение

Задача кластеризации заключается в том чтобы выполнить разбиение индивидов на кластеры на основе их сходства друг с другом (близость относительно выбранной метрики), при этом сами кластеры или их количество, как правило, заранее не известны. Кластеры строятся так, что характеристики для объектов внутри одного кластера близки, а характеристик объектов из разных кластеров сильно отличаются.

## Кластеризация. Формальное описание задачи

Пусть имеется подмножество  $\mathbf{X} \subset \mathbb{R}^p$ , которое будем называть пространством объектов, выборка  $\mathbf{X}^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , где  $\mathbf{x}_i$  — индивиды, определяемые вектором признаков,  $C$  — множество кластеров. Задача состоит в том чтобы найти такую функцию  $a : X \rightarrow Y$ , которая разбила бы выборку на непересекающиеся кластеры  $\mathbf{X}^n = \bigcup_{j=1}^k C_j$ ,  $C_i \cap C_j = \emptyset$ , таким образом, чтобы объекты одного кластера были близки по функции расстояния между объектами  $\rho : \mathbb{R}^p \times \mathbb{R}^p \rightarrow [0, \infty)$  и существенно отличались для объектов разных кластеров.

## Кластеризация. Формальное описание задачи

Не существует «истинных» или «лучших» определений для кластера. Что понимать под кластером должно быть определено исследователем, который применяет методы кластеризации. Как правило, для этого нужно определить характеристики кластера в отношении размера и формы, а также предполагаемых различий между кластерами.

Общая схема процесса кластеризации данных включает в себя:

- ▶ определение меры сходства;
- ▶ разбиение множества объектов на кластеры;
- ▶ оценку качества кластеризации;
- ▶ интерпретацию результатов.

Решение задачи кластеризации принципиально неоднозначно, так как число кластеров, как правило, не известно заранее, к тому же результат кластеризации сильно зависит от метрики  $\rho$ , выбор которой также не однозначен.

## Кластеризация. EM - алгоритм для model-based подхода

Один из вариантов формализовать задачу кластеризации это сделать предположение о статистическом распределении данных. Затем задача будет состоять в поиске параметров этого распределения. Предположим, что модель данных состоит из  $k$  смеси распределений. Пусть  $\omega_1 \dots \omega_k$  — априорные вероятности появления объектов из соответствующих кластеров,  $p_1(x) \dots p_k(x)$  — плотности распределения признаков внутри кластеров. Тогда плотность распределения сразу для всех кластеров равна взвешенной сумме плотностей по каждому кластеру:

$$p(x) = \sum_{i=1}^k \omega_i p_i(x).$$

## Кластеризация. EM - алгоритм для model-based подхода

Поставим задачу разделения смеси распределений, оценим по выборке  $\omega_1 \dots \omega_k$  и  $p_1(x) \dots p_k(x)$ . Это позволит оценить вероятность принадлежности индивида к разным кластерам и решить к какому кластеру его отнести. Часто рассматриваются случаи когда распределение смеси принадлежат одному семейству распределений, например нормальному, но с разным набором параметров для каждого из кластеров.

$$p_i(x) = \varphi(\theta_i; x)$$

Согласно методу максимального правдоподобия

$$\omega, \theta = \operatorname{argmax}_{\omega, \theta} \sum_{i=1}^n \ln p(x_i) = \operatorname{argmax}_{\omega, \theta} \sum_{i=1}^n \ln \sum_{j=1}^k \omega_j \varphi(\theta_j; x_i)$$

Максимизация логарифма суммы достаточно сложна, поэтому задача не решается напрямую с помощью метода максимума правдоподобия. Для максимизации логарифма функции правдоподобия применяется EM-алгоритм.

# Кластеризация. EM - алгоритм для model-based подхода

## Е - шаг.

В начале работы алгоритма задаём значения параметров  $\omega, \theta = (\omega_1 \dots, \omega_k; \theta_1 \dots \theta_k)$ , и подставляя их рассчитываем скрытые переменные. Скрытые переменные  $h_{ij} = P(\theta_j | x_i)$  — это вероятность того, что индивид  $x_i$  принадлежит  $j$  смеси. Найдем скрытые переменные по формуле Байеса:

$$h_{ij} = \frac{\omega_j \varphi(\theta_j; x_i)}{\sum_{s=1}^k \omega_s \varphi(\theta_s; x_i)}.$$

Для любого индивида  $\sum_{j=1}^k h_{ij} = 1$ .



# Кластеризация. EM - алгоритм для model-based подхода

## М - шаг.

На этом шаге будут рассчитываться значения параметров, которые мы ищем, используя скрытые параметры, полученные на предыдущем шаге. Решение методом Лагранжа для максимизации логарифма правдоподобия (с ограничением

$\sum_{j=1}^k \omega_j = 1$ ) даёт оценку для параметров:

$$\omega_j = \frac{1}{n} \sum_{i=1}^n h_{ij}$$

$$\theta_j = \operatorname{argmax}_{\theta} \sum_{i=1}^n h_{ij} \ln \varphi(\theta; x_i)$$

Таким образом, параметры будут уточняться на каждом шаге.

## Кластеризация. EM - алгоритм для model-based подхода

Если сделать предположение о том что классы принадлежат семейству нормальных распределений, то параметрами модели являются математическое ожидание и ковариационная матрица. Если не делать никаких предположений о ковариациях использование общей модели может быть весьма затруднительно, проблема заключается в большом количестве параметров, которые необходимо оценить. Ковариационные матрицы описывают геометрические характеристики кластеров, а именно объем, форму и ориентацию кластера. Общая модель предполагает, что все эти геометрические характеристики различны для каждого кластера. Однако, оценка плотности смеси, состоящей из кластеров одинаковой формы или ориентации, намного проще. Поэтому, сделав предположения о ковариационных матрицах, можно существенно облегчить задачу.

# Кластеризация. Байесовский подход для GMM

Суть подхода заключается в том, чтобы рассматривать параметры модели как случайные величины с некоторыми априорными распределениями, после чего вычислять апостериорное распределение параметров

$p(\pi, \gamma | X) \propto p(\pi, \gamma) p(X | \pi, \gamma)$  на основе данных.

Распределения параметров для конечного числа кластеров:

$$x_i | z_i, \{\mu_k, \sigma_k\} \sim \mathcal{N}(\mu_{z_i}, \sigma_{z_i}),$$

$$z_i | \pi \sim \text{Discrete}(\pi_1, \dots, \pi_k),$$

$$\{\mu_k, \sigma_k\} \sim \text{NIW}(\beta) \text{ (Normal-inverse-Wishart),}$$

$$\pi \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_k).$$

Сэмплирование параметров осуществляется с помощью вариационного вывода или методов Монте-Карло (Методы Монте-Карло для марковских цепей – например, метод сэмплирования по Гиббсу).

## Кластеризация. Байесовский подход для GMM

Существует обобщение данного подхода, при котором не нужно задавать число кластеров, а априорное распределение параметров задано в виде реализации процесса Дирихле с параметрами  $\alpha > 0$  (коэффициент концентрации) и  $H$  (базовое распределение). Процесс Дирихле является случайной вероятностной мерой, т.е. распределением над распределениями случайной величины. Таким образом, реализация случайного процесса  $\xi(\omega, \cdot)$  – распределение над множеством  $\mathbb{R}^d$ .

Индексирующий элемент – измеримое подмножество  $\mathbb{R}^d$ .

Будем говорить, что случайное распределение вероятностей (случайная вероятностная мера)  $G$  распределена согласно процессу Дирихле с параметрами  $H$  и  $\alpha$ , если для любого конечного измеримого разбиения

$(A_1, \dots, A_n), A_i \cap A_j = \emptyset, \bigcup_{i=1}^n A_i = \mathbb{R}^d$  случайный вектор

$(G(A_1), \dots, G(A_n))$  распределен согласно распределению Дирихле  $Dir(\alpha H(A_1), \dots, \alpha H(A_n))$ .

# Кластеризация. Байесовский подход для GMM

$$\begin{aligned} G &\sim \text{DP}(\mathcal{N}(\mu \mid 0, \sigma_\mu I), \alpha), \\ \mu_i &\sim G, \quad i = 1, \dots, n, \\ x_i &\sim \mathcal{N}(x_i \mid \mu_i, \sigma_x I), \quad i = 1, \dots, n \end{aligned}$$

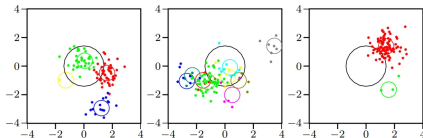


Figure: Примеры генерации выборки

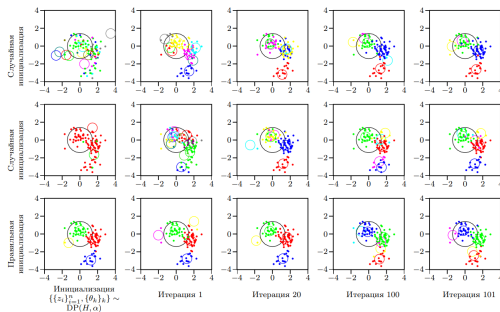


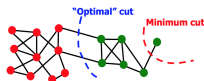
Figure: Кластеризация с помощью infinite GMM

# Кластеризация. Спектральная кластеризация

Спектральная кластеризация - алгоритм кластеризации, использующий спектр матрицы сходства данных для снижения размерности перед последующей кластеризацией в пространствах меньшей размерности.

Пусть дан граф  $G(V, E)$ . Требуется разбить его на 2 непересекающиеся группы  $(A, B)$  наилучшим в смысле  $\phi(A, B)$  образом.

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$$



$$\phi(A, B) = \frac{\text{cut}(A, B)}{\min(\text{vol}(A), \text{vol}(B))} \rightarrow \min$$

## Кластеризация. Спектральная кластеризация

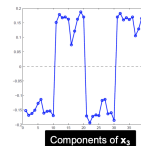
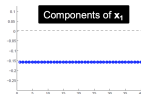
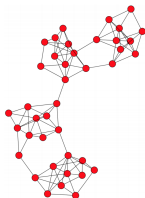
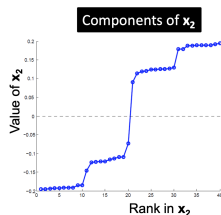
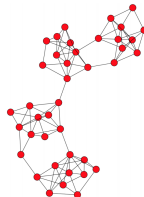
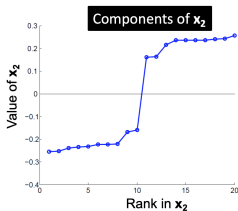
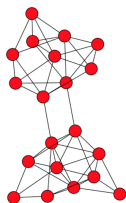
Матрица смежности  $A$  ( $n \times n$ ) для неориентированного графа симметрична, а значит обладает действительными собственными значениями и ортогональным базисом из собственных векторов. Матрица  $D$  – диагональная матрица, диагональные элементы которой являются степенями соответствующих вершин. Зададим Лапласиан  $L = D - A$ , все его собственные числа неотрицательны. Тривиальным собственным числом и собственным вектором Лапласиана являются  $0$  и  $\mathbf{1}_n$  соответственно.

Для симметричных матриц верно:  $\lambda_2 = \min_{x: x^T w_1 = 0} \frac{x^T M x}{x^T x}$ , тогда

$$\lambda_2 = \min_{x: \sum x_i = 0} \sum_{(i,j) \in E} (x_i - x_j)^2. \quad (1)$$

# Кластеризация. Спектральная кластеризация

(1) минимально, когда  $x$  – собственный вектор (вектор Фидлера), соответствующий  $\lambda_2$ . Если мы разделим граф по знаку компонент вектора Фидлера, то получим близкое к оптимальному решение в смысле  $\phi(A, B)$ .





## Кластеризация. Спектральная кластеризация

**Вход:** выборка  $X^n = \{x_1, \dots, x_n\}$ , способ задания матрицы смежности  $A$ , количество кластеров  $k$ , количество собственных векторов  $l$ , способ кластеризации в пространстве меньшей размерности;

**Выход:** разбиение выборки на кластеры;

1. Задаем матрицу  $A$ ;
2. Находим  $l$  собственных векторов  $L$  с наименьшими с.ч.;
3. Если последовательный метод;
4.     **Пока** число кластеров  $< k$ ;
5.         Делим на 2 кластера  $A, B$  по знакам элементов  $w_2$ ;
6.         Применяем алгоритм к одному из имеющихся кластеров с  $k$ , меньшим на 1.
7. **Иначе** применяем алгоритм нахождения  $k$  кластеров (например, k-means) для матрицы, столбцами которой являются  $l$  собственных векторов из шага 2.

# Кластеризация. Спектральная кластеризация

Способы задания матрицы  $A$ :

1. RBF:  $a_{i,j} = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ ,  $\sigma$  – параметр
2. Cosine Similarity:  $a_{i,j} = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$
3. Матрица смежности для графа  $k$  (или  $\varepsilon$ ) ближайших соседей ( $k$  – не число кластеров, а параметр)

Для определения числа кластеров можно использовать метод локтя для собственных чисел. Вместо  $L$  можно использовать нормированные версии:  $L_{\text{sym}} = E - D^{-1/2} A D^{-1/2}$ ,  
 $L_{\text{rw}} = E - D^{-1} A$ .

**Достоинства:** Хорошо работает для кластеров сложной формы, понижает размерность данных, много гиперпараметров.

**Недостатки:** Высокая вычислительная сложность и потребление памяти, чувствителен к выбросам, шуму и выбору гиперпараметров.

# Кластеризация. Алгоритм DBSCAN

**DBSCAN** (Density-based spatial clustering of applications with noise) — это эвристический алгоритм кластеризации, который предложили Маритин Эстер, Ганс-Петер Кригель, Ёрг Сандер и Сяовэй Су в 1996. Это алгоритм кластеризации, основанный на плотности — алгоритм группирует вместе те объекты, которые тесно расположены, помечая как выбросы объекты, которые находятся в областях с малой плотностью.

В этом алгоритме рассматривается для каждого объекта  $\mathbf{x} \in U$  его  $\varepsilon$ -окрестность  $U_\varepsilon(\mathbf{x}) = \{\mathbf{u} \in U : \rho(\mathbf{x}, \mathbf{u}) \leq \varepsilon\}$ .

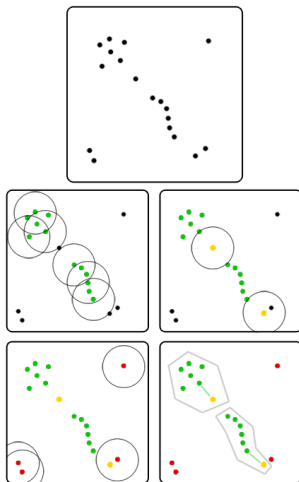
# Кластеризация. Алгоритм DBSCAN

В этом алгоритме рассматривается для каждого объекта  $\mathbf{x} \in U$  его  $\varepsilon$ -окрестность  $U_\varepsilon(\mathbf{x}) = \{\mathbf{u} \in U : \rho(\mathbf{x}, \mathbf{u}) \leq \varepsilon\}$ .

Каждый объект может быть одного из трёх типов:

- ▶ **корневой**: имеет плотную окрестность  $|U_\varepsilon(\mathbf{x})| \geq m$
- ▶ **граничный**: не корневой, но находится в окрестности корневого
- ▶ **выброс**: не корневой и не граничный.

## Кластеризация. Алгоритм DBSCAN



**Figure:** Иллюстрация к алгоритму DBSCAN. На рисунке зелёным отмечены корневые объекты, жёлтым — граничные и красным — шумовые.

## Кластеризация. Алгоритм DBSCAN

Корневые объекты находящиеся в  $\varepsilon$ -окрестности друг друга объединяются в один кластер. Граничные объекты относятся к тому кластеру, к какому относится корневой объект, в  $\varepsilon$ -окрестности которого лежит данный граничный объект. Таким образом, в итоге получается разделение всех объектов на кластеры и шумовые объекты.

## Кластеризация. Алгоритм DBSCAN

**Вход:** выборка  $X^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , параметры  $\varepsilon$  и  $m$ ;

**Выход:** разбиение выборки на кластеры и шумовые выбросы;

1.  $U = X^n$ ,  $a = 0$ ;
2. **Пока** есть некластеризованные точки, т.е.  $U \neq \emptyset$ ;
3.   взять случайную точку  $\mathbf{x} \in U$ ;
4.   **если**  $|U_\varepsilon(\mathbf{x})| < m$ , **то**
5.     пометить  $\mathbf{x}$  как шумовой;
6.   **иначе**
7.     создать новый кластер:  $K = U_\varepsilon(\mathbf{x})$ ;  $a = a + 1$ ;
8.     **для всех**  $\mathbf{x}' \in K$
9.       **если**  $|U_\varepsilon(\mathbf{x}')| \geq m$  **то**  $K = K \cup U_\varepsilon(\mathbf{x}')$ ;
10.      **иначе** пометить  $\mathbf{x}'$  как граничный элемент  $K$ ;
11.   соотнести объект классу  $a$  для всех  $\mathbf{x}' \in K$ ;
12.    $U = U \setminus K$

# Кластеризация. Алгоритм DBSCAN

## Достоинства:

- ▶ Относительно быстрая кластеризация больших данных (от  $O(n \ln n)$  до  $O(n^2)$  в зависимости от реализации);
- ▶ Позволяет обрабатывать кластеры произвольной формы (в том числе протяжённые ленты, концентрические гиперсферы);
- ▶ Помимо деления на кластеры выдаёт ещё и разметку шумовых объектов;
- ▶ Сам определяет количество кластеров (по модулю задания других гиперпараметров);
- ▶ Хорошо поддаётся модифицированию (существуют реализации, скрещенные с k-means, например).

## Недостатки:

Алгоритм может неадекватно обрабатывать сильные вариации плотности данных внутри кластера, проёмы и шумовые мосты между кластерами. То есть метод не способен соединять кластеры через проёмы, и, наоборот, связывает явно различные кластеры через плотно населённые перемычки.



# Кластеризация. Оценка результата

## Среднее внутрикластерное расстояние

$$F_0 = \frac{\sum_{i < j} \mathbf{I}_{\{y_i = y_j\}} \rho(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{i < j} \mathbf{I}_{\{y_i = y_j\}}},$$

Решая задачу кластеризации, мы хотим по возможности получать как можно более кучные кластеры, то есть минимизировать  $F_0$ .

## Среднее межкластерное расстояние

$$F_1 = \frac{\sum_{i < j} \mathbf{I}_{\{y_i \neq y_j\}} \rho(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{i < j} \mathbf{I}_{\{y_i \neq y_j\}}}.$$

Среднее межкластерное расстояние, напротив, нужно максимизировать, то есть целесообразно выделять в разные кластеры наиболее удалённые друг от друга объекты. Имеет смысл вычислять отношение пары функционалов, чтобы учесть как внутрикластерные, так и межкластерные расстояния:  $F_0/F_1 \rightarrow \min$ .

# Кластеризация. Оценка результата

## Коэффициент силуэта

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}.$$

$$\text{DBI} = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left( \frac{s_i + s_j}{d_{ij}} \right).$$

# Кластеризация

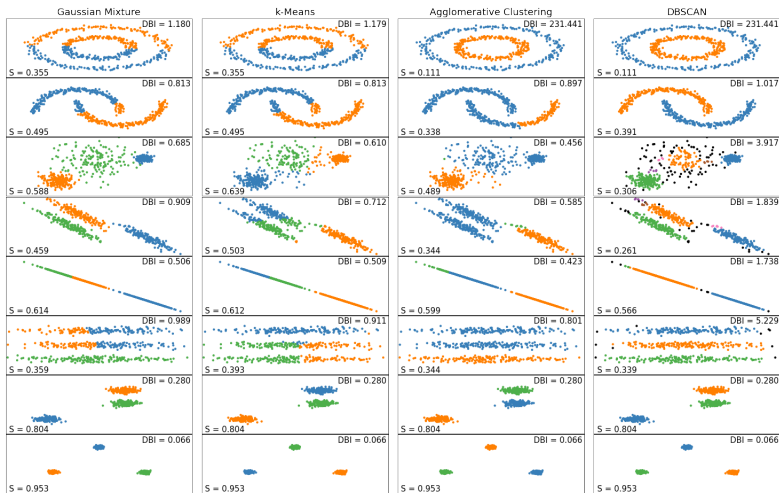


Figure: Сравнение результатов работы различных алгоритмов кластеризации

# Тематическое обучение. Введение

Тематическое моделирование — технология статистического анализа текстов для автоматического выявления тематики в больших коллекциях документов.

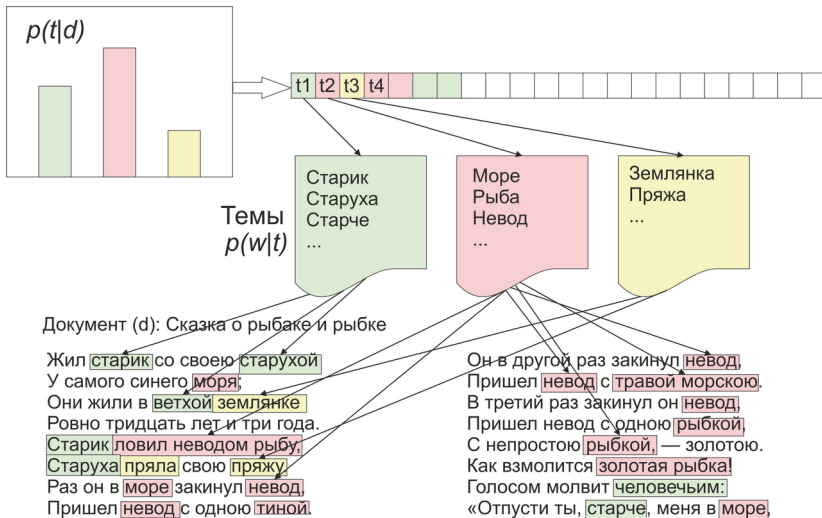
Чем-то похоже на кластеризацию, но тематическое моделирование в этом плане является «мягким» и допускает, чтобы документ относился к нескольким кластерам-темам.

Тематическое моделирование не претендует на понимание смысла текста, однако оно способно отвечать на вопросы «о чём этот текст» или «какие общие темы имеет эта пара текстов».

## Тематическое обучение. Параметры тематической модели

- ▶  $p(\omega|t)$  — матрица искомых условных распределений слов по темам;
- ▶  $p(t|d)$  — матрица искомых условных распределений тем по документам;
- ▶  $d$  — документ;
- ▶  $\omega$  — слово;
- ▶  $d, \omega$  — наблюдаемые переменные;
- ▶  $t$  — тема (скрытая переменная).

# Тематическое обучение. Параметры тематической модели



## Тематическое обучение. LSA

Латентно-семантический анализ, LSA (latent semantic analysis), он же LSI (latent semantic indexing) — самая ранняя модель, предложенная еще в конце 80-х гг. Модель называется латентной, т.к. предполагает введение скрытого (латентного) параметра — темы.

LSA основан на использовании сингулярного разложения матрицы. С помощью SVD-разложения любая матрица раскладывается во множество ортогональных матриц, линейная комбинация которых является достаточно точным приближением к исходной матрице. Этим и объясняется название этого алгоритма в sklearn — TruncatedSVD.

# Тематическое обучение. LSA

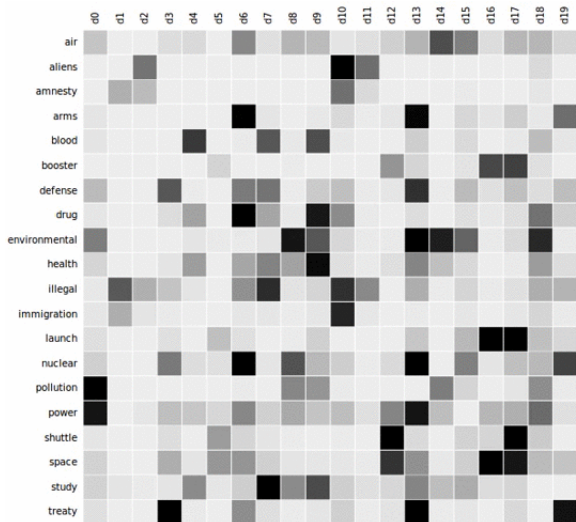


Figure: Иллюстрация SVD в методе LSA (исходная матрица).



# Тематическое обучение. LSA

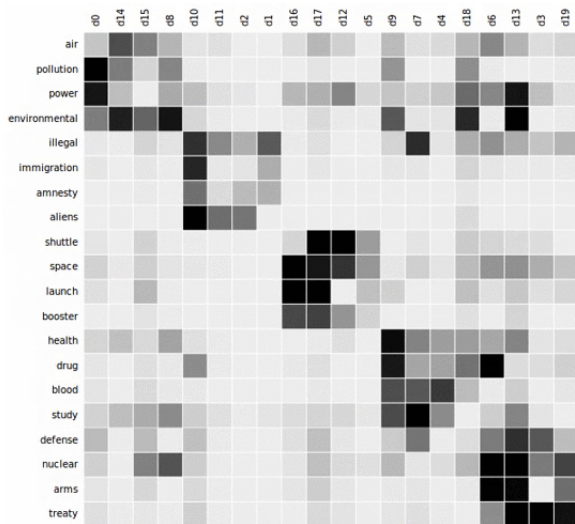


Figure: Иллюстрация SVD в методе LSA.

## Тематическое обучение. pLSA

pLSA (probabilistic latent semantic analysis), она же pLSI (probabilistic latent semantic indexing) — вероятностный латентно-семантический анализ (индексирование). Модель предложена в 1999 г. Томасом Хоффманом.

Зачем понадобилось модифицировать LSA? Проблема этого метода в том, что он предполагает, что слова и документы имеют нормальное распределение, но в реальности это не так. Поэтому на практике чаще используется pLSA, основанный на мультиномиальном распределении. Если LSA — это чистая линейная алгебра, то pLSA имеет еще и статистические основания.

## Тематическое обучение. pLSA

**Дана** коллекция текстовых документов (мешок слов):  $n_{d\omega}$  — сколько раз термин  $\omega$  встречается в документе  $d$ .

**Найти** модель  $p(\omega|d) = \sum_t \phi_{\omega t} \theta_{td}$  с параметрами  $\phi, \theta$ :

- ▶  $\phi_{\omega t} = p(\omega|t)$  — вероятности терминов  $\omega$  в каждой теме  $t$ ;
- ▶  $\theta_{td} = p(t|d)$  — вероятности тем  $t$  в каждом документе  $d$ .

## Тематическое обучение. pLSA

Неизвестная модель находится путем максимизации логарифма правдоподобия:

$$\sum_{d,\omega} n_{d,\omega} \ln \sum_t \phi_{\omega t} \theta_{td} \rightarrow \max_{\phi, \theta}$$

при ограничениях нормировки и неотрицательности:

$$\phi_{\omega t} \geq 0; \sum_{\omega} \phi_{\omega t} = 1; \theta_{td} \geq 0; \sum_t \theta_{td} = 1$$

## Тематическое обучение. pLSA

Точка максимума правдоподобия  $\phi, \theta$  удовлетворяет системе уравнений со вспомогательными переменными  $p_{td\omega}$ :

$$\begin{cases} p_{td\omega} = \frac{\phi_{\omega t} \theta_{td}}{\sum_{t'} \phi_{\omega t'} \theta_{t' d}} \\ \phi_{\omega t} = \frac{n_{\omega t}}{\sum_{\omega'} n_{\omega' t}}; n_{\omega t} = \sum_{d \in D} n_{d\omega} p_{td\omega} \\ \theta_{td} = \frac{n_{td}}{\sum_{t'} n_{t' d}}; n_{td} = \sum_{\omega \in d} n_{d\omega} p_{td\omega} \end{cases}$$

Где первое уравнение это  $E$ -шаг  $EM$  алгоритма, а второе и третье уравнение —  $M$ -шаг.

## Тематическое обучение. pLSA

*E*-шаг – это формула Байеса:

$$p_{td\omega} = p(t|d, \omega) = \frac{p(\omega, t|d)}{p(\omega|d)} = \frac{p(\omega|t)p(t|d)}{p(\omega|d)} = \frac{\phi_{\omega t}\theta_{td}}{\sum_{s \in T} \phi_{\omega t}\theta_{td}}$$

$n_{d\omega t} = n_{d\omega}p(t|d, \omega)$  — оценка числа троек  $(d, \omega, t)$  в коллекции

## Тематическое обучение. pLSA

***M*-шаг — это частотные оценки условных вероятностей:**

$$\phi_{\omega t} = \frac{n_{\omega t}}{n_t} \equiv \frac{\sum_{d \in D} n_{d\omega t}}{\sum_{d \in D} \sum_{\omega \in d} n_{d\omega t}}$$

$$\theta_{td} = \frac{n_{td}}{n_d} \equiv \frac{\sum_{\omega \in D} n_{d\omega t}}{\sum_{\omega \in \Omega} \sum_{t \in T} n_{d\omega t}}$$

## Тематическое обучение. pLSA

**Вход:** коллекция  $D$ , число тем  $|T|$  и число итераций  $i_{max}$ ;

**Выход:** матрица терминов тем  $\Theta$  и тем документов  $\Phi$ ;

1. инициализация  $\phi_{\omega t}, \theta_{td}$  для всех  $d \in D, \omega \in \Omega, t \in T$ ;
2. для всех итераций  $i = 1, \dots, i_{max}$
3.  $n_{\omega t}, n_{td}, n_t, n_d := 0$  для всех  $d \in D, \omega \in \Omega, t \in T$ ;
4. для всех документов  $d \in D$  и слов  $\omega \in d$
5. 
$$p_{td\omega} = \frac{\phi_{\omega t} \theta_{td}}{\sum_s \phi_{\omega s} \theta_{sd}};$$
6.  $n_{\omega t}, n_{td}, n_t, n_d += n_{d\omega} p_{td\omega}$  для всех  $t \in T$ ;
7.  $\phi_{\omega t} := n_{\omega t} / n_t$  для всех  $\omega \in \Omega, t \in T$ ;
8.  $\phi_{td} := n_{td} / n_d$  для всех  $d \in D, t \in T$ ;