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

Михайлов Дмитрий, Смирнов Иван

Санкт-Петербургский государственный университет
Кафедра статистического моделирования

Санкт-Петербург, 2023

Обучение без учителя

Обучение без учителя - это один из способов машинного обучения, при котором испытуемая 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 $\boldsymbol{X} \subset \mathbb{R}^p$, которое будем называть пространством объектов, выборка $\boldsymbol{X}^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, где $\boldsymbol{x}_i$ — индивиды, определяемые вектором признаков, C — множество кластеров. Задача состоит в том чтобы найти такую функцию $a : X \to Y$, которая разбила бы выборку на непересекающиеся кластеры $\boldsymbol{X}^n = \bigcup_{j=1}^{k} C_j,\ C_i \bigcap C_j = \emptyset$, таким образом, чтобы объекты одного кластера были близки по функции расстояния между объектами $\rho : \mathbb{R}^p \times \mathbb{R}^p \to [0, \infty)$ 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 ρ, 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

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

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

E - шаг.
В начале работы алгоритма задаём значения параметров $\omega, \theta = (\omega_1 \ldots, \omega_k; \theta_1 \ldots \theta_k)$, 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:

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

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

Кластеризация. EM - алгоритм для model-based подхода

M - шаг.

На этом шаге будут рассчитываться значения параметров, которые мы ищем, используя скрытые параметры, полученные на предыдущем шаге. Решение методом Лагрнажа 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. Алгоритм k-средних (k-means)

Метод k-means осуществляет декомпозицию набора данных, состоящего из n наблюдений, на k кластеров с заранее неизвестными параметрами. При этом выполняется поиск центроидов - максимально удаленных друг от друга центров сгущений точек C_k с минимальным разбросом внутри каждого кластера.

В качестве меры близости выбрано евклидово расстояние:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|^2.$$

Основная идея алгоритма заключается в минимизации меры близости между индивидами внутри одного кластера:

$$\min_{C_1,\ldots,C_k} \left\{ \sum_{l=1}^{k} \frac{1}{|C_l|} \sum_{i,i' \in C_l} \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|^2 \right\}.$$

Кластеризация. Алгоритм k-средних (k-means)

1. Выбираем начальное приближение центров кластеров $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ случайным образом;
2. Соотносим каждый объект к ближайшему центру (аналог E-шага)
$$C(i) = \underset{0 \le j \le k}{\text{argmin}} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2;$$
3. Для каждого кластера C_j пересчитываем центры $\boldsymbol{\mu}_j$ как выборочное среднее индивидов, которые были отнесены к этому кластеру (аналог M-шага);
4. Повторяем шаги 2 и 3 пока принадлежность кластерам не перестанет изменяться.

Иными словами, делаем следующее: инициализируем центры, затем разделяем индивиды по ближайшему центру кластера, перевычисляем каждый из центров, и если ничего не изменилось, останавливаемся, если изменилось, то повторяем.

Кластеризация. Алгоритм k-средних (k-means)

Достоинства:

- Простота реализации
- Алгоритм очень гибкий
- Существует множество различных модификаций этого алгоритма

Недостатки:

- Кластеризация очень сильно зависит от начального приближения
- Кластеризация может быть неадекватной, если изначально было выбрано неверное число кластеров
- Необходимость самостоятельно задавать число кластеров
- Форма кластеров только сферическая

Кластеризация. Иерархическая кластеризация

Методы иерархической кластеризации основываются на двух идеях:

- агломерации (AGNES) — последовательное объединение индивидуальных объектов или их групп во все более крупные подмножества
- разбиении (DIANA) — начинается с корня и на каждом шаге делит образующие группы по степени их гетерогенности

Более распространены агломеративные алгоритмы, общий вид которых приведён дальше.

Кластеризация. Иерархическая кластеризация

1. Одноэлементные кластеры:

$$C_1 = \{\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_n\}\};\ R_1 = 0$$

$$\forall\, i \neq j \text{ вычислить } R(\{\boldsymbol{x}_i\}, \{\boldsymbol{x}_j\})$$

2. для всех $t = 2, \ldots, n$ (t — номер итерации)
3. найти в C_{t-1} два ближайших кластера:

$$(U, V) = \arg\min_{U \neq V} R(U, V); R_t = R(U, V);$$

4. слить их в один кластер:

$$W = U \cup V;\ C_t = C_{t-1} \cup W \backslash \{U, V\}$$

5. для всех $S \in C_t \backslash W$
6. вычислить расстояние $R(W, S)$ по формуле Ланса-Уильямса.

Кластеризация. Иерархическая кластеризация

В начальный момент времени каждый объект содержится в собственном кластере. Далее происходит итеративный процесс слияния двух ближайших кластеров до тех пор, пока все кластеры не объединятся в один или не будет найдено необходимое число кластеров. На каждом шаге необходимо уметь вычислять расстояние между кластерами и пересчитывать расстояние между новыми кластерами.

Кластеризация. Иерархическая кластеризация

Расстояние между одноэлементными кластерами определяется через расстояние между объектами: $R(\{x\},\{y\}) = \rho(x,y)$. То есть сначала нужно задать, как мы будем измерять расстояние между точками. Например, это могут быть

- Евклидово расстояние: $\rho(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$.
- Расстояние городских кварталов (манхэттенское расстояние): $\rho(x,y) = \sum_i |x_i - y_i|$.
- Расстояние Чебышёва: $\rho(x,y) = \max_i |x_i - y_i|$.

Важно либо исходно стандартизовать признаки, либо измерять расстояние специальным образом (использовать расстояние Махаланобиса вместо обычного евклидового, если есть предположения о форме распределения точек внутри кластера).

Кластеризация. Иерархическая кластеризация

Для вычисления же расстояния $R(U, V)$ между кластерами U и V на практике используются различные функции в зависимости от специфики задачи. Изначально было придумано множество различных способов определить такие расстояния, но оказалось, что практически все разумные, являются частным случаем **формулы Ланса-Уильямса**, которая позволяет обобщить большинство способов определить расстояние между кластерами $R(W, S)$, $W = U \cup V$, U, V, $S \subset X$, зная расстояния $R(U, S)$, $R(V, S)$, $R(U, V)$:

$$R(W, S) = \alpha_U R(U, S) + \alpha_V R(V, S) + \beta R(U, V) + \gamma |R(U, S) - R(V, S)|,$$

где α_U, α_V, β, γ — числовые параметры.

Кластеризация. Иерархическая кластеризация

Ниже приведены некоторые способы определения расстояний явно и соответствующие им коэффициенты для формулы Ланса-Уильямса.

- Расстояние ближнего соседа (single linkage clustering) — расстояние между кластерами оценивается как минимальное из дистанций между парами объектов, один из которых входит в первый кластер, а другой - во второй:

$$R^{\text{б}}(W,S) = \min_{w\in W, s\in S} \rho(w,s); \quad \alpha_U = \alpha_V = 1/2, \ \beta = 0, \ \gamma = -1/2$$

- Расстояние дальнего соседа (complete linkage clustering) — вычисляется расстояние между наиболее удаленными объектами:

$$R^{\text{д}}(W,S) = \max_{w\in W, s\in S} \rho(w,s); \quad \alpha_U = \alpha_V = 1/2, \ \beta = 0, \ \gamma = 1/2;$$

.

Кластеризация. Иерархическая кластеризация

- Среднее расстояние (average linkage clustering) — на каждом следующем шаге объединяются два ближайших кластера, рассчитывая среднюю арифметическую дистанцию между всеми парами объектов:

$$R^{\text{с}}(W,S) = \frac{1}{|W||S|}\sum_{w\in W}\sum_{s\in S}\rho(w,s);$$

$$\alpha_U = \frac{|U|}{|W|},\ \ \alpha_V = \frac{|V|}{|W|},\ \ \beta = \gamma = 0;$$

- Расстояние между центрами:

$$R^{\text{ц}}(W,S) = \rho^2\left(\sum_{w\in W}\frac{w}{|W|}, \sum_{s\in S}\frac{s}{|S|}\right);$$

$$\alpha_U = \frac{|U|}{|W|},\ \ \alpha_V = \frac{|V|}{|W|},\ \ \beta = -\alpha_U\,\alpha_V,\ \ \gamma = 0;$$

.

Кластеризация. Иерархическая кластеризация

- Расстояние Уорда (метод минимума дисперсии Уорда):

$$R^{\text{ч}}(W,S) = \frac{|S||W|}{|S|+|W|}\rho^2\left(\sum_{w\in W}\frac{w}{|W|},\sum_{s\in S}\frac{s}{|S|}\right);$$

$$\alpha_U = \frac{|S|+|U|}{|S|+|W|},\ \ \alpha_V = \frac{|S|+|V|}{|S|+|W|},\ \ \beta = -\frac{|S|}{|S|+|W|},\ \ \gamma = 0;$$

- Гибкое расстояние:

$$\alpha_U = \alpha_V = \tfrac{1-\beta}{2},\ \ \beta < 1\ (-0.25),\ \ \gamma = 0.$$

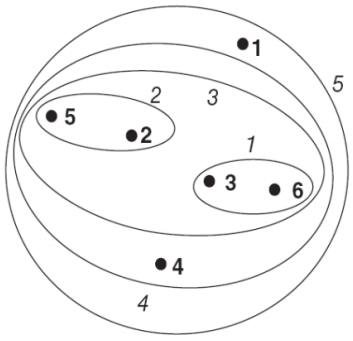
.

Кластеризация. Иерархическая кластеризация

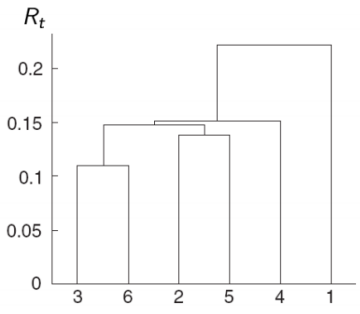
Результатом работы иерархического алгоритма является **дендрограмма** — древовидный график расстояний, при которых произошло слияние кластеров на каждом шаге.

1. Расстояние ближнего соседа:

Диаграмма вложения

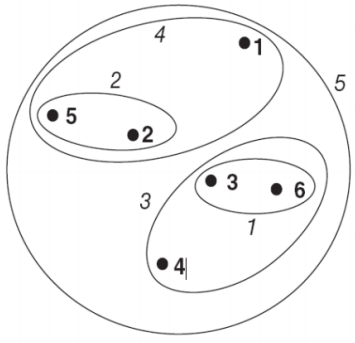


Дендрограмма

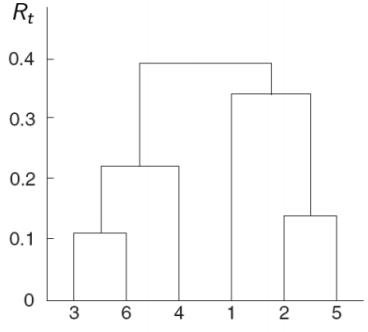


2. Расстояние дальнего соседа:

Диаграмма вложения

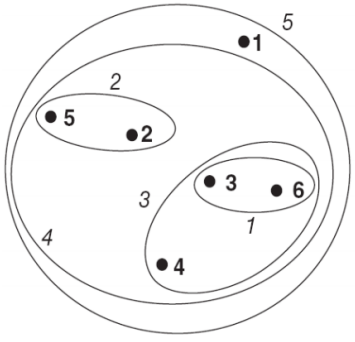


Дендрограмма

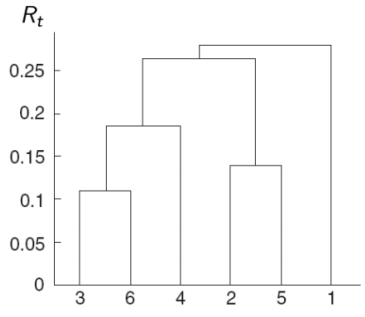


3. Групповое среднее расстояние:

Диаграмма вложения

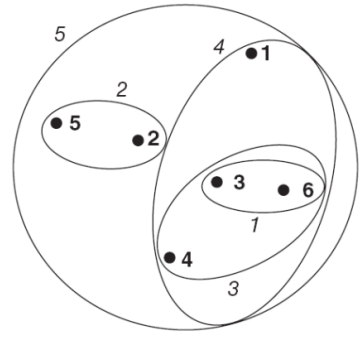


Дендрограмма

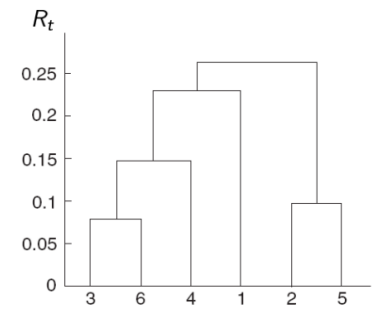


5. Расстояние Уорда:

Диаграмма вложения



Дендрограмма



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 $\boldsymbol{x} \in U$ его ε-окрестность $U_\varepsilon(\boldsymbol{x}) = \{\boldsymbol{u} \in U : \rho(\boldsymbol{x}, \boldsymbol{u}) \leq \varepsilon\}$.

Кластеризация. Алгоритм DBSCAN

В этом алгоритме рассматривается для каждого объекта $\boldsymbol{x} \in U$ его ε-окрестность $U_\varepsilon(\boldsymbol{x}) = \{\boldsymbol{u} \in U : \rho(\boldsymbol{x}, \boldsymbol{u}) \leq \varepsilon\}$.
Каждый объект может быть одного из трёх типов:

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

Кластеризация. Алгоритм DBSCAN

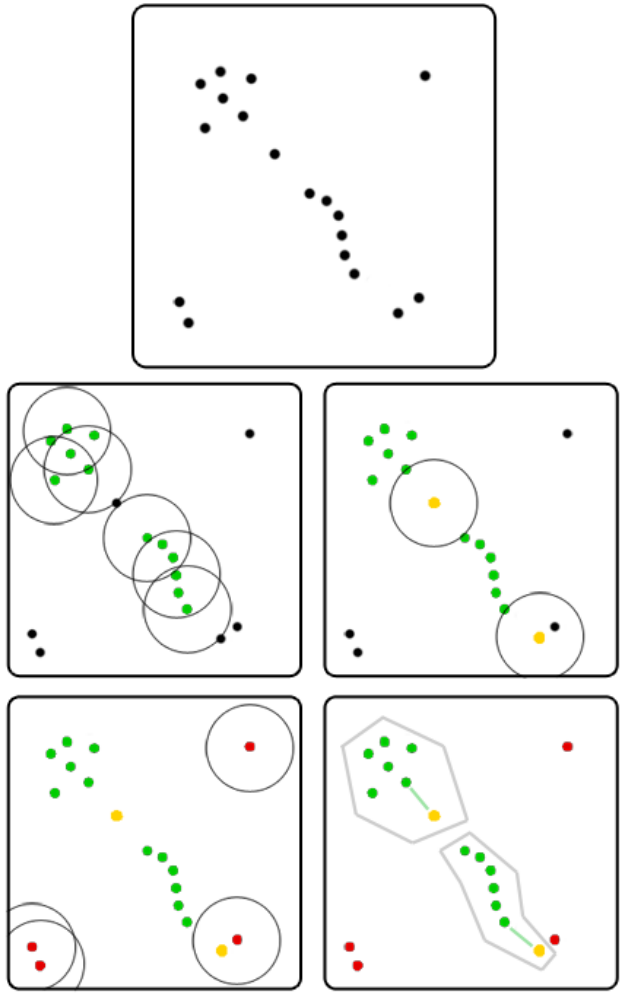


Figure: Иллюстрация к алгоритму DBSCAN. На рисунке зелёным отмечены корневые объекты, жёлтым — граничные и красным — шумовые.

Кластеризация. Алгоритм DBSCAN

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

Кластеризация. Алгоритм DBSCAN

Вход: выборка $\boldsymbol{X}^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, параметры ε и m;
Выход: разбиение выборки на кластеры и шумовые выбросы;

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

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

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

Среднее межкластерное расстояние

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

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: $F_0/F_1 \to min$.

Кластеризация

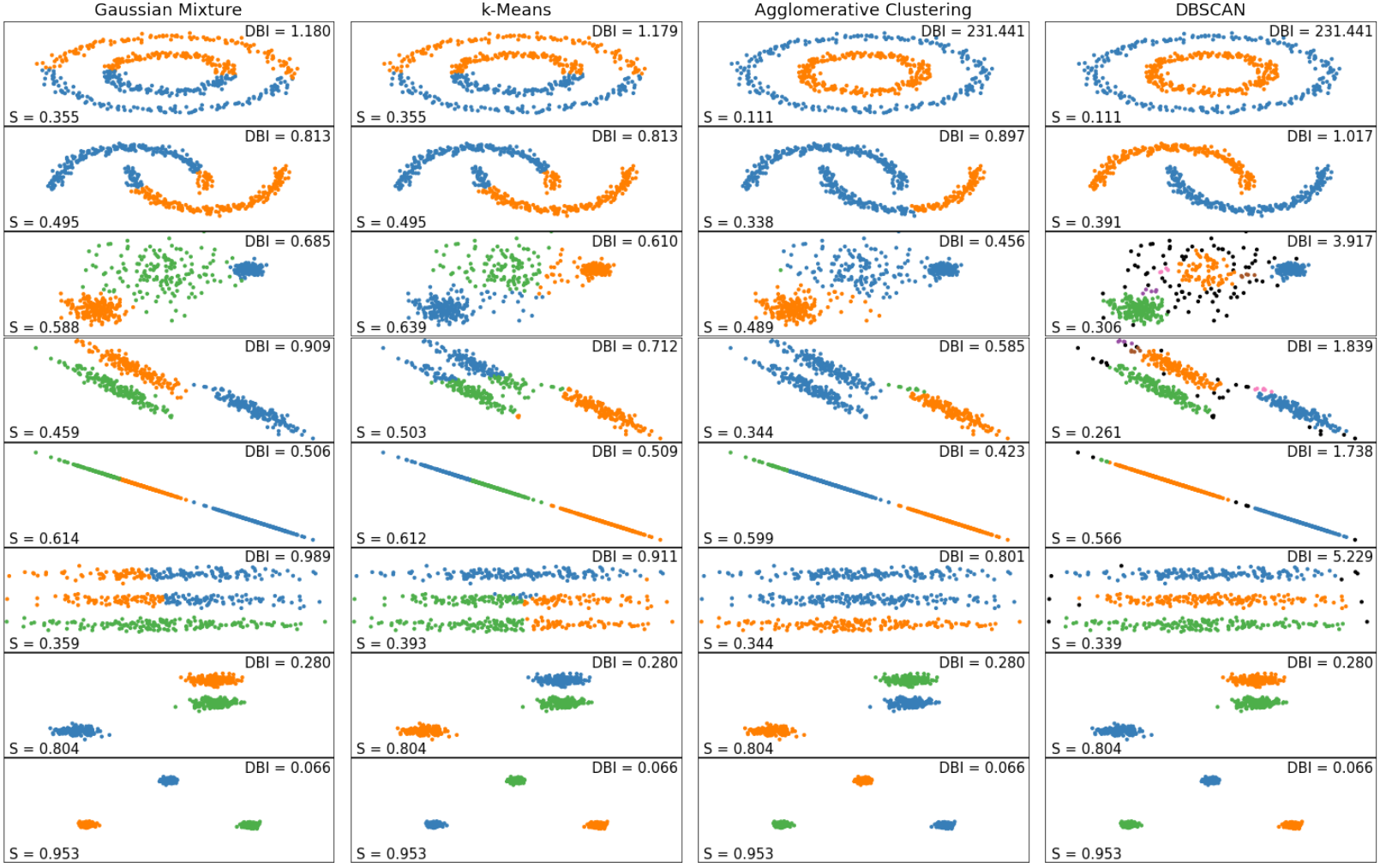


Figure: Сравнение результатов работы различных алгоритмов кластеризации

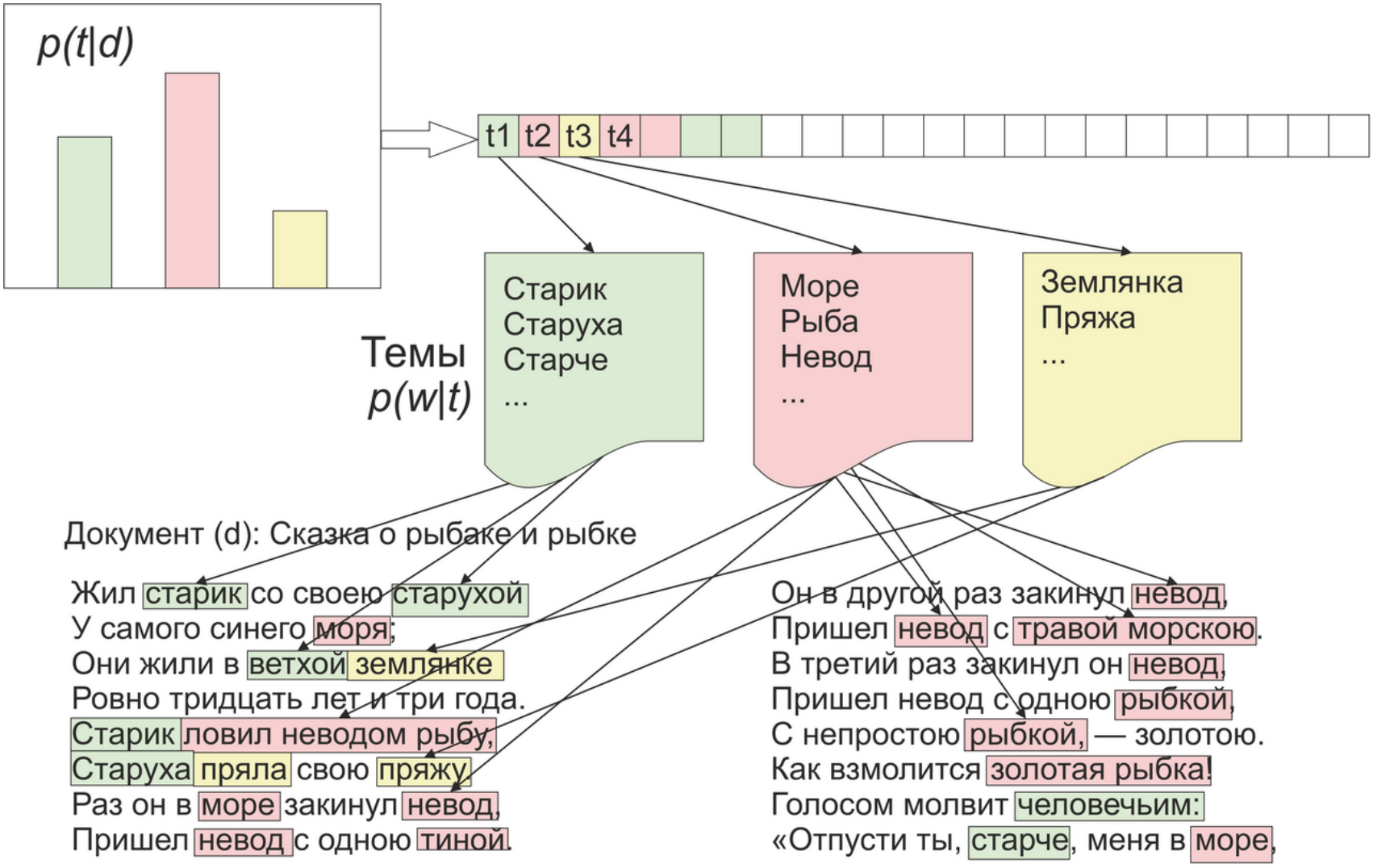
Тематическое обучение. Введение

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;
- ω — слово;
- d, ω — 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 — TruncaredSVD.

Тематическое обучение. LSA

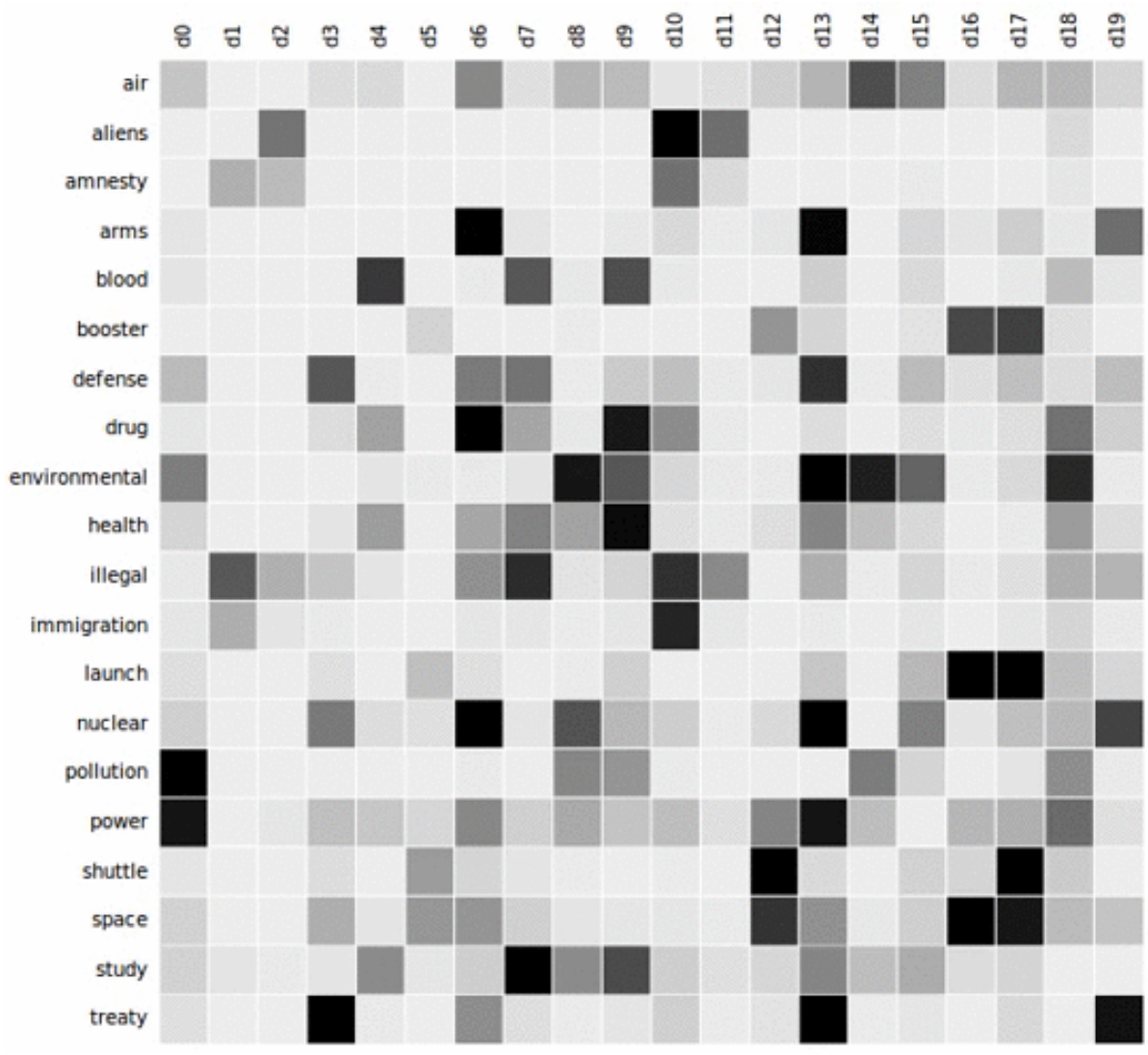


Figure: Иллюстрация SVD в методе LSA (исходная матрица).

Тематическое обучение. LSA

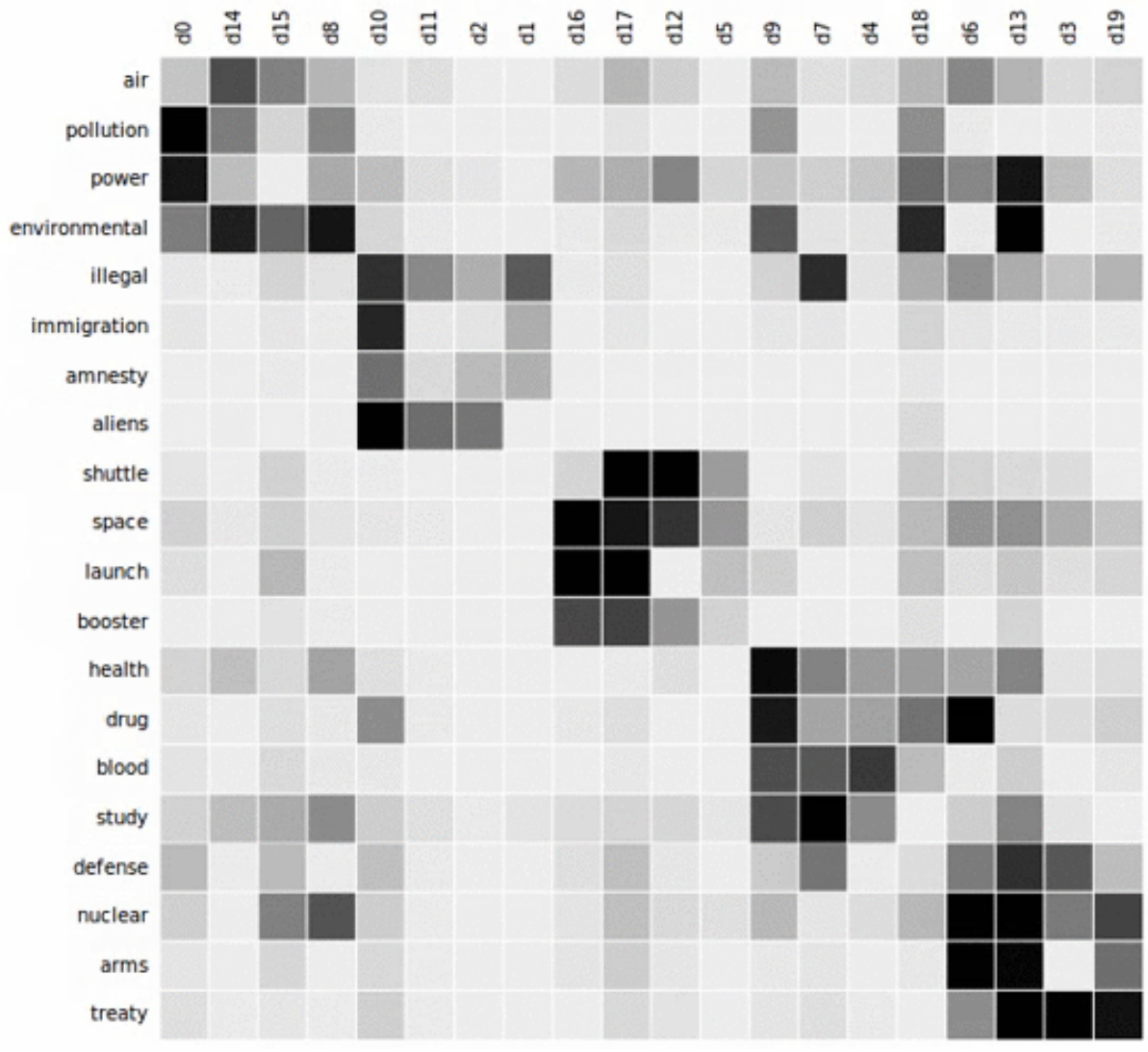


Figure: Иллюстрация SVD в методе LSA.

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 ω встречаетсяя в документе d.
Найти модель $p(\omega|d) = \sum_t \phi_{\omega t}\theta_{td}$ с параметрами ϕ, θ:

- $\phi_{\omega t} = p(\omega|t)$ — вероятности терминов ω 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:

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

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 ϕ, θ удовлетворяет системе уравнений со вспомогательными переменными $p_{td\omega}$:

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

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,ω,t) в коллекции

Тематическое обучение. pLSA

M-шаг — это частотные оценки условных вероятностей:

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

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

Тематическое обучение. pLSA

Вход: коллекция D, число тем $|T|$ и число иттераций i_{max};
Выход: матрица терминов тем Θ и тем документов Φ;

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