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

Козак М., Мехнин П., Шкурат Д.

10.11.2022

1 Кластеризация

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.

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.

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.

2 EM - алгоритм

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 $\omega_1 \ldots \omega_k$ — априорные вероятности появления объектов из соответствующих кластеров, $p_1(x) \ldots p_k(x)$ — плотности распределения признаков внутри кластеров. Тогда плотность распределения сразу для всех кластеров равна взвешенной сумме плотностей по каждому кластеру:

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

Поставим задачу разделения смеси распределений, оценим по выборке $\omega_1 \ldots \omega_k$ и $p_1(x) \ldots p_k(x)$. 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.

$$p_i(x) = \varphi(\theta_i; x) \tag{2}$$

Согласно методу максимального правдоподобия

$$\omega, \theta = \underset{\omega,\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \ln p(x_i) = \underset{\omega,\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \ln \sum_{j=1}^{k} \omega_j \varphi(\theta_j; x_i) \tag{3}$$

Максимизация логарифма суммы достаточно сложна, поэтому задача не решается напрямую с помощью метода максимума правдоподобия. Для максимизации логарифма функции правдоподобия применяется EM-алгоритм. Это итеративный алгоритм, состоящий из двух шагов, в котором на первом шаге задаются какие-нибудь значения параметров, а затем с каждой итерацией эти параметры уточняются.

E - 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:

$$h_{ij} = \frac{\omega_j \varphi(\theta_j; x_i)}{\sum\limits_{s=1}^{k} \omega_s \varphi(\theta_s; x_i)}. \tag{4}$$

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

M - шаг.

На этом шаге будут рассчитываться значения параметров, которые мы ищем, используя скрытые параметры, полученные на предыдущем шаге. Решение методом Лагрнажа для максимизации (3) (с ограничением $\sum\limits_{j=1}^{k} \omega_j = 1$) даёт оценку для параметров:

$$\omega_j = \frac{1}{n} \sum_{i=1}^{n} h_{ij} \tag{5}$$

$$\theta_j = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} h_{ij} \ln \varphi(\theta; x_i) \tag{6}$$

Таким образом, параметры будут уточняться на каждом шаге.

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.

3 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:

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

3.1 Алгоритм

1. Выбираем начальное приближение центров кластеров $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ случайным образом;
2. Соотносим каждый объект к ближайшему центру (аналог Е-шага)

$$C(i) = \underset{0 \leqslant j \leqslant k}{\operatorname{argmin}} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2;$$

3. Для каждого кластера C_j пересчитываем центры $\boldsymbol{\mu}_j$ как выборочное среднее индивидов, которые были отнесены к этому кластеру (аналог М-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.

3.2 Сложность и возможные оптимизации

Средняя сложность определяется как $O(knT)$, где k — количество кластеров, n — количество индивидов, а T — количество итераций до сходимости.

Сложность в наихудшем случае определяется как $O(n^{k+2/p})$, где n — количество индивидов, p — количество признаков.

На практике алгоритм k-средних очень быстр (один из самых быстрых доступных алгоритмов кластеризации), но он попадает в локальные минимумы. Вот почему может быть полезно перезапустить его несколько раз.

При больших n целесообразно использовать модификацию MiniBatchKMean, который выполняет обновления позиций центров с помощью мини-пакетов вместо всей выборки.

При больших p целесообразно уменьшить размерность простанства признаков с помощью АГК.

3.3 k-means и его связь с EM - алгоритмом

Алгоритм k-средних является частным случаем для гауссовой смеси распределения с диагональными ковариационными матрицами, у которых одинаковые значения на диагоналях.

В таком случае:

- На Е-шаге мы не считаем вероятности g_{ij} принадлежности i-го объекта j-ому кластеру, а приписываем каждый объект одному кластеру (вероятность принадлежности будет равна 0 или 1);
- Форма кластеров не настраивается: они все являются сферическими.

3.4 Достоинства и недостатки

Достоинства:

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

Недостатки:

- Кластеризация очень сильно зависит от начального приближения
  Выгодно брать максимально удаленные друг от друга центры. Неудачный выбор центров может привести к плохому результату кластеризации. Для решения этой проблемы можно провести кластеризацию с несколькими начальными приближениями и выбрать лучший вариант. Также на практике работает следующая эвристика (K-means++): первый центр выбираем случайно из равномерного распределения на точках выборки, а каждый следующий центр выбираем из случайного распределения на объектах выборки, в котором вероятность выбрать объект пропорциональна квадрату расстояния от него до ближайшего к нему центра кластера.
- Кластеризация может быть неадекватной, если изначально было выбрано неверное число кластеров

- Необходимость самостоятельно задавать число кластеров
  Подбирать число кластеров можно с помощью коэффицента силуэта, либо использовать метод "локтя" (elbow method), который рассматривает характер изменения внутригруппового разброса с увеличением числа групп k. На каком-то этапе снижение этой дисперсии замедляется — на графике это происходит в точке, называемой "локтем" (аналогично "каменистой осыпи" для анализа главных компонент).
- Форма кластеров только сферическая

4 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 ниже.

4.1 Алгоритм агломеративной иерархической кластеризации

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.

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.

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).

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.

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:
$$R^{\text{д}}(W,S) = \max_{w\in W, s\in S} \rho(w,s); \quad \alpha_U = \alpha_V = 1/2, \ \beta = 0, \ \gamma = 1/2;$$
- Среднее расстояние (average linkage clustering) — на каждом следующем шаге объединяются два ближайших кластера, рассчитывая среднюю арифметическую дистанцию между всеми парами объектов:
$$R^{\text{с}}(W,S) = \frac{1}{|W||S|}\sum_{w\in W}\sum_{s\in S}\rho(w,s); \quad \alpha_U = \frac{|U|}{|W|}, \ \alpha_V = \frac{|V|}{|W|}, \ \beta = \gamma = 0;$$
- Расстояние между центрами:
$$R^{\text{ц}}(W,S) = \rho^2\left(\sum_{w\in W}\frac{w}{|W|}, \sum_{s\in S}\frac{s}{|S|}\right); \quad \alpha_U = \frac{|U|}{|W|}, \ \alpha_V = \frac{|V|}{|W|}, \ \beta = -\alpha_U\,\alpha_V, \ \gamma = 0;$$
- Расстояние Уорда (метод минимума дисперсии Уорда):
$$R^{\text{ц}}(W,S) = \frac{|S||W|}{|S|+|W|}\rho^2\left(\sum_{w\in W}\frac{w}{|W|}, \sum_{s\in S}\frac{s}{|S|}\right);$$
$$\alpha_U = \frac{|S|+|U|}{|S|+|W|}, \ \alpha_V = \frac{|S|+|V|}{|S|+|W|}, \ \beta = -\frac{|S|}{|S|+|W|}, \ \gamma = 0;$$
- Гибкое расстояние:
$$\alpha_U = \alpha_V = \tfrac{1-\beta}{2}, \ \beta < 1 \ (-0.25), \ \gamma = 0.$$

.

4.2 Визуализация кластерной структуры

Результатом работы иерархического алгоритма является **дендрограмма** — древовидный график расстояний, при которых произошло слияние кластеров на каждом шаге. В узлах дерева находятся подмножества объектов. При этом на каждом ярусе дерева множество объектов из всех узлов составляет исходное множество объектов. Объединение узлов между ярусами соответствует слиянию двух кластеров. При этом длина ребра соответствует расстоянию между кластерами. Введем обозначение R_t — расстояние между кластерами, выбранными на шаге t для объединения. Дендрограмма позволяет представлять зависимости между множеством объектов с любым числом заданных характеристик на двумерном графике, где по одной из осей откладываются все объекты, а по другой — расстояние R_t. Если не накладывать на это расстояние никаких ограничений, то дендрограмма будет иметь большое число самопересечений и изображение перестанет быть наглядным. Чтобы любой кластер мог быть представлен в виде непрерывного отрезка на оси объектов и ребра не пересекались, необходимо наложить ограничение монотонности на R_t.

Определение 1. *Функция расстояния R является монотонной, если на каждом следующем шаге расстояние между кластерами не уменьшается:* $R_1 \leqslant R_2 \leqslant \ldots \leqslant R_m$

Вторым желательным свойством является свойство растяжения. Кластеризация называется сжимающей, если $R_t \leqslant \rho(\mu_U, \mu_V), \forall t$ и называется растягивающей, если $R_t \geqslant \rho(\mu_U, \mu_V), \forall t$. При выборе сжимающих расстояний объекты с каждым шагом всё больше слипаются друг с другом. Свойство растяжения же позволяет

лучше отделять кластеры друг от друга. Однако, при этом расстояние не должно быть сильно растягивающим, иначе объекты удалятся друг от друга настолько, что появится много лишних кластеров. В качестве меры растяжения рассматривается отношение расстояния между кластерами к расстоянию между центрами кластеров. Относительно приведённых свойств самыми оптимальными оказываются расстояние Уорда и гибкое расстояние, а расстояние между центрами оказывается самым плохим, поскольку оно единственное не является монотонным. При этом гибкое расстояние оказывается сжимающим при $\beta > 0$ и растягивающим при $\beta < 0$.

Для определения числа кластеров находится интервал максимальной длины $|R_{t+1} - R_t|$ (длинная ветка у дерева). В качестве итоговых кластеров выдаются кластеры, полученные на шаге t. Однако, когда число кластеров заранее неизвестно и объектов в выборке не очень много, бывает полезно изучить дендрограмму целиком.

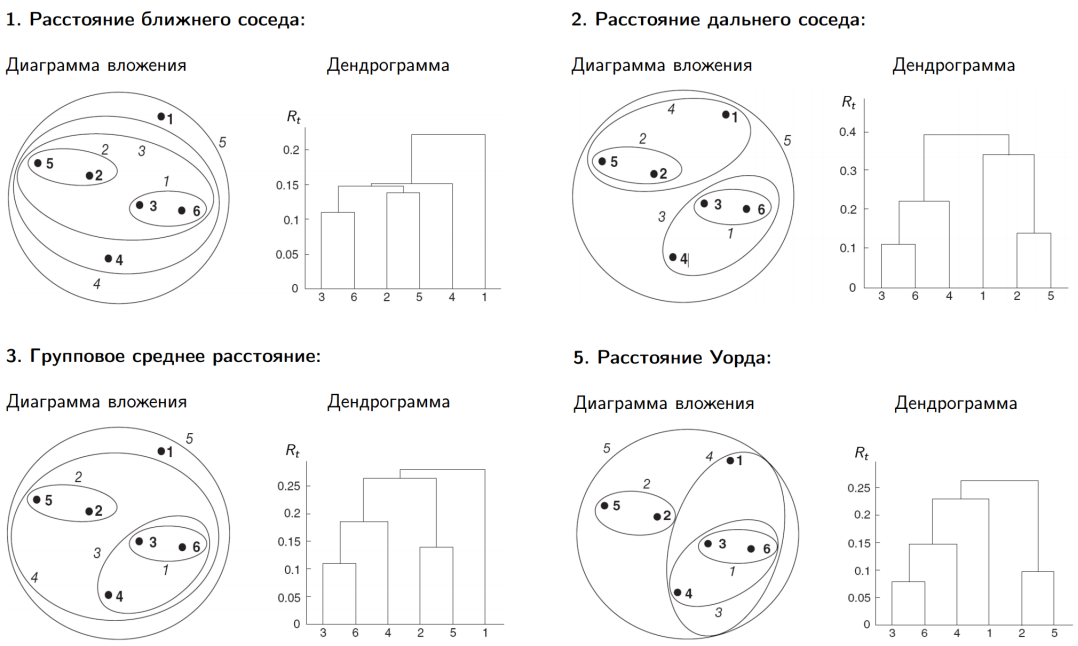


Рис. 1: Дендрограммы при выборе различных расстояний.

4.3 Плюсы и минусы

Достоинства:

- В качестве результата можно получить дендрограмму, которая может представлять самостоятельный интерес.
- Форма кластеров может быть произвольной.
- Количество кластеров можно определить по дендрограмме.

Недостатки:

- Необходимость подбирать одно из множества различных расстояний.
- Отсутствие модели в задаче не позволяет однозначно предпочесть одно разделение на кластеры другому.

5 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 $U_\varepsilon(\boldsymbol{x}) = \{\boldsymbol{u} \in U : \rho(\boldsymbol{x}, \boldsymbol{u}) \leqslant \varepsilon\}$.

Каждый объект может быть одного из трёх типов:

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

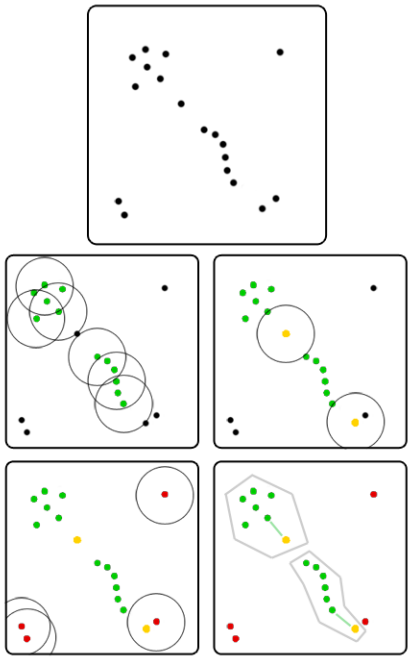


Рис. 2: Иллюстрация к алгоритму DBSCAN. На рисунке зелёным отмечены корневые объекты, жёлтым — граничные и красным — шумовые.

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.

5.1 Алгоритм

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** $|U_\varepsilon(\boldsymbol{x}')| \geqslant m$ **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. $U = U \setminus K$

5.2 Плюсы и минусы

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. Проблема особенно актуальна для данных большой размерности, так как чем больше p, тем больше мест, где могут случайно возникнуть проёмы или мосты.

6 Функционалы качества кластеризации

Задачу кластеризации можно ставить как задачу дискретной оптимизации: приписать номера кластеров объектам так, чтобы значение выбранного функционала качества приняло наилучшее значение. Существует много разновидностей функционалов качества кластеризации, но нет «самого правильного». По сути дела, каждый метод кластеризации можно рассматривать как точный или приближённый алгоритм поиска оптимума некоторого функционала. Исходя из начальной постановки задачи «меньше расстояние внутри кластеров — больше снаружи», естественно выбрать среднее внутрикластерное и среднее межкластерное расстояние.

6.1 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.

6.2 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: $F_0/F_1 \to min$.

Существует ещё множество различных более сложных функционалов качества. Смысл всех таких функционалов один и тот же: попытаться описать близость в кластерах и дальность между ними.

6.3 Коэффициент силуэта

Коэфициент силуэта является мерой того, насколько похож объект на другие объекты из своего кластера в сравнении с объектами из других кластеров.

Введем вспомогательные величины:

- Среднее расстояние между $\boldsymbol{x}_i$ и объектами того же кластера

$$c(\boldsymbol{x}_i) = \frac{1}{|K_i| - 1} \sum_{\boldsymbol{x}_j \in K_i,\, i\neq j} \rho(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- Среднее расстояние между $\boldsymbol{x}_i$ и объектами следующего ближайшего кластера.

$$b(\boldsymbol{x}_i) = \min_{i \neq j} \frac{1}{|K_j|} \sum_{\boldsymbol{x}_z \in K_j} \rho(\boldsymbol{x}_i, \boldsymbol{x}_z)$$

Коэффициент определяется для каждого объекта выборки, а метрика для результатов кластеризации всей выборки вводится как средний коэффициент силуэта для всех объектов выборки.

- Силуэт такого объекта тогда равен

$$s(\boldsymbol{x}_i) = \begin{cases} \frac{b(\boldsymbol{x}_i) - c(\boldsymbol{x}_i)}{\max\{c(\boldsymbol{x}_i), b(\boldsymbol{x}_i)\}}, & |K_i| > 1 \\ 0, & |K_i| = 1 \end{cases}$$

- Естественным образом силуэт кластеризации определяется как среднее силуэтов всех объектов: $S = \frac{1}{n} \sum\limits_i s(\boldsymbol{x}_i)$.

Данный функционал качества максимизируется. Значения силуэта изменяются от -1 до 1, их можно интерпретировать так: -1 — кластеризация точно не удалась, 0 — удалась кластеризация или нет относительно силуэта неизвестно, $+1$ — кластеризация точно удалась. Обычно, желательно, чтобы значение силуэта для кластеризации оказалось не менее 0.75. Как видно из определения, для вычисления силуэта необходимо, чтобы кластеров было как минимум два. Ещё одна проблема его в том, что он не очень корректно обрабатывает ленточные кластеры, перекрывающиеся и кластеры с перемычками. Как только расстояние между объектами одного кластера становится сравнимым с расстоянием между объектами разных кластеров, силуэт перестаёт быть адекватным функционалом качества. Кроме того, в силуэт никак не заложен шум. Это значит, что если в данных встретится выброс, то силуэт будет больше (а значит, согласно ему, кластеризация удалась лучше), если выброс будет посчитан как отдельный кластер. Таким образом, данный функционал качества заточен именно под кластеры такого вида, когда они представляют собой далеко отстоящие компактные скопления объектов.

6.4 Индекс Дэвиса-Болдина, DBI

Является средним отношением внутрикластерных разбросов к расстояниям между кластерами.

Предположим, имеется разбиение данных на K кластеров, и в нем каждый кластер C_i имеет размер $|C_i| = T_i$ и центроид A_i. Пусть объекты X_j принадлежат кластеру C_i.

Мерой компактности кластера C_i назовем величину

$$S_i = \left(\frac{1}{T_i} \sum_{j=1}^{T_i} |X_j - A_i|^p \right)^{1/p},$$

т.е. среднее расстояние от объектов кластера до их центроидов. Обычно $p = 2$ (эвклидово расстояние).

Мерой отделимости кластеров C_i и C_j назовем величину

$$M_{i,j} = ||A_i - A_j||_p = \Big(\sum_{k=1}^{n} |a_{k,i} - a_{k,j}|^p \Big)^{\frac{1}{p}},$$

т.е. расстояние между центроидами.

Введём величину $R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$ и найдём $D_i = \max_{j \neq i} R_{i,j}$, тогда

$$DBI = \frac{1}{N} \sum_{i=1}^{N} D_i.$$

Наилучшее разбиение на кластеры минимизирует DBI.

Существует ещё группа функционалов, которая использует внешнюю информацию об априорном разделении объектов на кластеры. К таковым относятся, например, Rand Index, Jaccard Index, Minkowski Score. Однако, подобные задачи уже не являются задачами обучения без учителя.

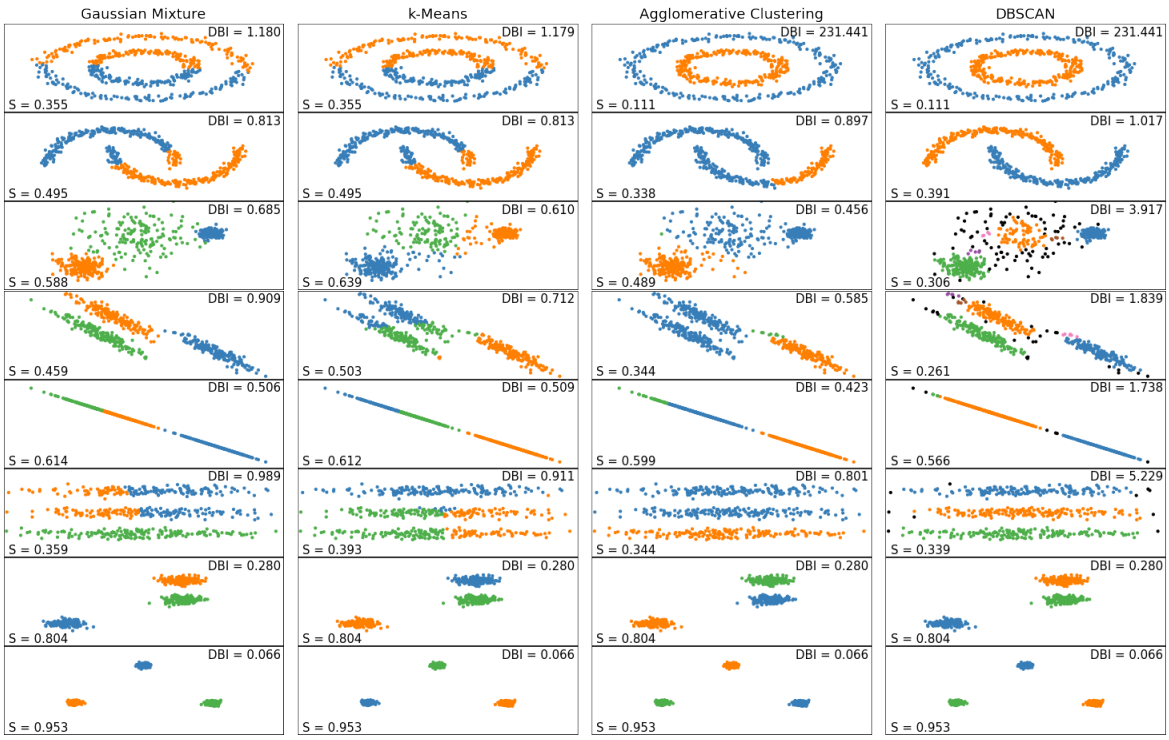


Рис. 3: Сравнение результатов работы различных алгоритмов кластеризации

7. Тематическое моделирование

Тематическое моделирование (topic modeling) -- приложение машиннного обучения к анализу текстов.

Тематическая модель (topic model) коллекции текстовых документов определяет, к каким темам относится каждый документ и какие слова (термины) образуют каждую тему.

Вероятностная тематическая модель (BTM) описывает каждую тему дискретным распределением на множестве терминов, каждый документ дискретным распределением на множестве тем. Предполагается, что коллекция документов -- это последовательность терминов, выбранных случайно и независимо из смеси таких рпспределений, и ставится задача восстановления компонент смеси по выборке.

BTM осуществляет мягкую классификацию (документ может относится к нескольким выборкам, при этом решается проблема синонимов и омонимов).

Применяется для:

- выявления трендов в научных публикациях/новостных потоках,
- классификации и категоризациии документов, изображений и видео,
- информационного поиска, в том числе многоязычного,
- тегирования веб-страниц,
- обнаружения спама,
- рекомендательных систем,
- и др.

7.1 Вероятностная модель колллекции документов

Пусть D -- множество (коллекция) текстовых документов, W -- множество (словарь) всех употребляемых в них терминов (слов или словосочетаний). Каждый документ $d \in D$ представляет собой последовательность n_d терминов $(w_1, \ldots, w_{n_d})$ из словаря W. Термин может повторяться в документе несколько раз.

7.1.1 Вероятностное пространство и гипотеза независимости

Предполагается, что существует конечное множество тем T, и каждое употребление термина w в каждом документе d связано с некоторой темой $t \in T$, которая неизвестна. Коллекция документов рассматривается как множество троек (d, w, t), заданного на конечном множестве $D \times W \times T$. Документы $d \in D$ и термины $w \in W$ являются наблюдаемыми переменными, тема $t \in T$ является латентной (скрытой) переменной.

Гипотеза о независимости элементов выборки эквивалентна предположению "Мешок слов" (bag-of-words), -- что порядок терминов в документах не важен для выявления тематики, то есть тематику документа можно узнать даже после произвольной

перестановки терминов, хотя для человека такой текст теряет смысл. Порядок документов в колллекции также не имеет значения -- предположение называют гипотезой "мешка документов".

Приняв гипотезу "мешка слов", можно перейти к более компактному представлению документа как подмножества $d \subset W$, в ккотором каждому жлементу $w \in d$ поставлено в соответствие число n_{dw} вхождений термина w в документ d.

7.1.2 Постановка задачи тематического моделирования

Построить тематическую модель коллекции документов D -- значит найти множество тем T, распределения $p(w|t)$ для всех тем $t \in T$ и распредления $p(t|d)$ для всех $d \in D$. Можно также говорить о задаче совместной "мягкой" класстеризации множества документов и множества слов по множесту кластер-тем. Мягкая кластеризация означает, что каждый документ или термин не жёстко приписывается какой-то одной теме, а распределяется по нескольким темам.

Найденные распределения используются затем для решения прикладных задач. Распределение $p(t|d)$ является удобным признаковым описанием документа в задачах информационного поиска, классификации и категоризации документов.

7.1.3 Гипотеза условной независимости

Будем полагать, что появление слов в документе d, относящихся к теме t, описывается общим для всей коллекции распредлением $p(w|t)$ и не зависит от документа d. Следующие представления этой гипотезы эквивалентны:

$$p(w|d,t) = p(w|t);$$
$$p(d|w,t) = p(d|t);$$
$$p(d,w|t) = p(d|t)p(w|t).$$

7.1.4 Вероятностная модель порождения данных

Согласно определению условной вероятности, формуле полной вероятности и гипотезе условной независимости

$$p(w|d) = \sum_{t \in T} p(t|d)p(w|t).$$

Если формула кажется не очевидной, приведём вывод:

Согласно опредлению условной вероятности:

$$p(w|d) = \frac{p(w,d)}{p(d)}$$

По формуле полной вероятности:

$$p(w,d) = \sum_{t \in T} p(w,d|t)p(t)$$

По гипотезе условной независимости (1. 3 пункт):

$$p(w, d) = \sum_{t \in T} p(w \mid t) p(d \mid t) p(t)$$

Расписав

$$p(d \mid t) = \frac{p(d, t)}{p(t)},$$

получим:

$$p(w, d) = \sum_{t \in T} p(w \mid t) p(d, t)$$

посмотри налево

Поскольку $p(d, t) = p(t, d)$,

$$p(t, d) = p(t \mid d) p(d),$$

и тогда:

$$p(w, d) = \sum_{t \in T} p(w \mid t) p(t \mid d) p(d).$$

Подставив в самое начало, получим:

$$p(w \mid d) = \frac{\sum\limits_{t \in T} p(w \mid t) p(t \mid d) p(d)}{p(d)},$$

откуда:

$$p(w \mid d) = \sum_{t \in T} p(t \mid d) p(w \mid t) p(d).$$

Если распределения $p(t \mid d)$ и $p(w \mid t)$ известны, то вероятностная модель (2) описывает процесс порождения коллекции D.

Построение тематической модели -- обратная задача: по известной коллекции D требуется восстановить породившие её распределения $p(t \mid d)$ и $p(w \mid t)$.

Алгоритм 1.1. Вероятностная модель порождения коллекции документов.

Вход: распределения $p(w \mid t)$, $p(t \mid d)$;
Выход: выборка пар (d_i, w_i), $i = 1, \ldots, n$;

```
для всех d ∈ D
    задать длину n_d документа d;
    для всех i = 1, ..., n_d
        выбрать случайную тему t из распределения p(t | d);
        выбрать случайный термин w из распределения p(w | t);
        добавить в выборку пару (d, w), при этом тема t «забывается»;
```

7.1.5 Гипотеза разреженности

Естественно предполагать, что каждый документ d и каждый термин w связан с небольшим числом тем t. В таком случае значительная часть вероятностей $p(t \mid d)$ и $p(w \mid t)$ должна обращаться в ноль.

Если документ относится к большому числу тем, то в задачах тематического поиска или классификации документов его имеет смысл разбивать на более однородные по тематике части.

Если термин относится к большому числу тем, то, скорее всего, это общеупотребительное слово, бесполезное для определение тематики.

Алгоритмы, в которых нулевые значения не хранятся, намного эффективнее по памяти и по скорости. Поэтому для больших коллекций разреженнность должна учитываться обязательно.

7.1.6 Частотные оценки условных вероятностей

Вероятности, связанные с наблюдаемыми перменными d и w, можно оценивать по выборке как частоты:

$$\hat{p}(d, w) = \frac{n_{dw}}{n}, \quad \hat{p}(d) = \frac{n_d}{n}, \quad \hat{p}(w) = \frac{n_w}{n}, \quad \hat{p}(w \mid d) = \frac{n_{dw}}{n_d},$$

где

- n_{dw} -- число вхождений термина w в документ d,
- $n_d = \sum\limits_{w \in W} n_{dw}$ -- длина документа d в терминах,
- $n_w = \sum\limits_{d \in D} n_{dw}$ -- число вхождений термина w во все документы коллекции,
- $n = \sum\limits_{d \in D} \sum\limits_{w \in d} n_{dw}$ -- длина коллекции в терминах.

Вероятности, связанные со скрытой переменной t, также можно оценивать как частоты, если рассматривать коллекцию документов как выборку троек (d, w, t):

$$\hat{p}(t) = \frac{n_t}{n}, \quad \hat{p}(w \mid t) = \frac{n_{wt}}{n_t}, \quad \hat{p}(t \mid d) = \frac{n_{dt}}{n_d}, \quad \hat{p}(t \mid d, w) = \frac{n_{dwt}}{n_{dw}},$$

где

- n_{dwt} -- число троек, в которых термин w документа d связан с темой t,
- $n_{dt} = \sum\limits_{w \in W} n_{dwt}$ -- число троек, в которых термин документа d связан с темой t,
- $n_{wt} = \sum\limits_{d \in D} n_{dwt}$ -- число троек, в которых термин w связан с темой t,
- $n_t = \sum\limits_{d \in D} \sum\limits_{w \in d} n_{dw}$ -- число троек, связанных с темой t.

По ЗБЧ в пределе при $n \to \infty$ частотные оценки $\hat{p}(\cdot)$, определяемые формулами 3, 4, стремятся к соответствующим вероятностям $p(\cdot)$. Частотная интерпретация даёт ясное понимание всех условных вероятностей, которые будут использоваться в дальнейшем.

7.1.7 Стохастическое матричное разложение

Если число тем $|T|$ много меньше числа документов $|D|$ и числа терминов $|W|$, то равенство (2) можно понимать как задачу приближённого представления заданной матрицы частот

$$\mathrm{F} = (\hat{p}_{wd})_{W \times D}, \quad \hat{p}_w d = \hat{p}(w \mid d) = n_{dw}/n_d,$$

в виде произведения $\mathrm{F} \approx \Phi\Theta$ двух неизвестных матриц меньшего размера -- матрицы терминов тем Φ и матрицы тем документов Θ:

$$\Phi = (\varphi_{wt})_{W \times T}, \quad \varphi_{wt} = p(w \mid t);$$
$$\Theta = (\theta_{td})_{T \times D}, \quad \theta_{td} = p(t \mid d).$$

Матрицы, столбцы которых неотрицательны и нормированны, называются стохастическими.

Представление матрицы F получается с помощью принципа максимума правдоподобия.

7.1.8 Принцип максимума правдоподобия.

Для оценивания параметров Φ, Θ тематической модели по коллекции документов D будем максимизировать правдоподобие (плотность распределения) выборки:

$$p(D; \Phi, \Theta) = C \prod_{d \in D} \prod_{w \in d} p(d, w)^{n_{dw}} = \prod_{d \in D} \prod_{w \in d} p(w \mid d)^{n_{dw}} C p(d)^{n_{dw}} \to \max_{\Phi, \Theta},$$

где C -- нормировочный множитель, зависящий от чисел n_{dw}. Отбросим постоянную часть:

$$\tilde{p}(D; \Phi, \Theta) = \prod_{d \in D} \prod_{w \in d} p(w \mid d)^{n_{dw}} \to \max_{\Phi, \Theta}.$$

Подставим выражение для $p(w \mid d)$ из (2):

$$\tilde{p}(D; \Phi, \Theta) = \prod_{d \in D} \prod_{w \in d} \left(\sum_{t \in T} p(t \mid d) p(w \mid t) \right)^{n_{dw}} \to \max_{\Phi, \Theta}.$$

Обозначим $\theta_{td} = p(t \mid d)$, $\varphi_{wt} = p(w \mid t)$:

$$\tilde{p}(D; \Phi, \Theta) = \prod_{d \in D} \prod_{w \in d} \left(\sum_{t \in T} \theta_{td} \varphi_{wt} \right)^{n_{dw}} \to \max_{\Phi, \Theta}.$$

Логарифмируем:

$$L(\Phi, \Theta) = \ln \tilde{p}(D; \Phi, \Theta) = \ln \left(\prod_{d \in D} \prod_{w \in d} \left(\sum_{t \in T} \theta_{td} \varphi_{wt} \right)^{n_{dw}} \right) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \theta_{td} \varphi_{wt} \to \max_{\Phi, \Theta}.$$

Получили задачу максимизации логарифма правдоподобия при ограничениях неотрицательности и нормированности стобцов матриц Φ и Θ:

$$L(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \theta_{td} \varphi_{wt} \to \max_{\Phi, \Theta},$$

$$s.\,t.$$

$$\sum_{w \in W} \varphi_{wt} = 1; \quad \varphi_{wt} \geqslant 0,$$

$$\sum_{t \in T} \theta_{td} = 1; \quad \theta_{td} \geqslant 0.$$

7.2 Предварительная обработка текстовых данных

При построении тематической модели нет смысла различать формы (склонения, спряжения) одного и того же слова. Это приведёт к неоправданному разрастанию словаря, дроблению статистики, увеличению ресурсоёмкости и снижению качества модели.

7.2.1 Лемматизация и стемминг

Лемматизация -- приведение каждого слова в документе к нормальной форме.

В русском:

- Существительные: именительный падеж, единственное число;
- Прилагательные: именительный падеж, единственное число, мужской род;
- Глагол, причастие, деепричастие: глагол в инфинитиве.

Лемматизаторы разрабатываются с помощью составления грамматического словаря со всеми формами слов, либо аккуратной формализации правил языка со всеми исключениями. Недостатками лемматизаторов является неполнота словарей, особенно по части специальной терминологии и неологизмов, которые часто и представляют наибольший интерес.

Стемминг -- отбрасывание изменяемых частей слов, в основном, окончаний.

Плюсы:

- Не требует словаря;
- Основана на правилах морфологии языка;
- Хорошо работает с английским языком;

Минусы:

- Большое число ошибок;
- Плохо работает для русского языка;

7.2.2 Отбрасывание стоп-слов

Слова, которые встречаются во многих текстах различной тематики -- бесполезны и могут быть отброшены. Это союзы, предлоги, числительные, местоимения, некоторые

глаголы, прилагательные и наречия. Отбрасывание почти не влияет на длину словаря, но может приводить к заметному сокращению длины некоторых текстов.

7.2.3 Отбрасывание редких слов

Слова, встречающие в длинном документе слишком редко, например, только один раз, можно отбросить, полагая, что данное слово не характеризует тематику данного документа.

7.2.4 Выделение ключевых фраз

При обработке специальных текстов вместо отдельных слов выделяют *ключевые фразы* -- словосочетания, являющиеся терминами предметной области. Это отдельная сложная задача, для решения которой необходимо привлечение экспертов (даже при использовании методов машинного обучения).

Далее будем полагать, что словарь W получен в результате предварительной обработки всех документов коллекции D и может содержать как отдельные слова, так и ключевые фразы. Элементы словаря $w \in W$ будем называть "терминами".

7.3 Вероятностный латентный сематический анализ (PLSA)

Вероятностный латентный семантический анализ (probabilistic latent semantic analysis).

Вероятностная модель появления пары "документ-термин" (d, w) записывается тремя эквивалентными способами:

$$p(d,w) = \sum_{t \in T} p(t)p(w|t)p(d|t) = \sum_{t \in T} p(d)p(w|t)p(t|d) = \sum_{t \in T} p(w)p(t|w)p(d|t),$$

где $p(t)$ -- распределение тем во всей коллекции. Первое представление называется симметричным, остальные -- несимметричными. Они приводят к немного разным итерационным процессам обучения тематической модели.

Сейчас возьмём второе представление, совпадающее с (2).

7.3.1 EM-алгоритм

Для решение задачи (6) в PLSA применяется итерационный процесс, в котором каждая итерация состоит из двух шагов E и M. Перед первой итерацией выбирается начальное приближение параметров φ_{wt} и θ_{td}.

На E-шаге по текущим значениям параметров φ_{wt} и θ_{td} с помощью формулы Байеса вычисляются условные вероятности $p(t|d,w)$ всех тем $t \in T$ для каждого термина $w \in d$ в каждом документе d:

$$H_{dwt} = p(t \mid d, w) = \frac{p(w \mid t)p(t \mid d)}{p(w \mid d)} = \frac{\varphi_{wt}\theta_{td}}{\sum\limits_{s \in T} \varphi_{ws}\theta_{sd}}$$

На M-шаге, наоборот, по условным вероятностям тем H_{dwt} вычисляется новое приближение параметров φ_{wt}, θ_{td}. Поскольку

$$\hat{n}_{dwt} = n_{dw}p(t \mid d, w) = n_{dw}H_{dwt}$$

оценивает (не обязательно целое) число n_{dwt} вхождений термина w в документ d, связанных с темой t. Просуммировав $\hat{n}_{dwt}$ по документам d и по терминам w, получим оценки $\hat{n}_{wt}, \hat{n}_{dt}, \hat{n}_t$ и через них, согласно (4) -- частотные оценки условных вероятностей φ_{wt}, θ_{td}:

$$\varphi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t}, \quad \hat{n}_t = \sum_{w \in W} \hat{n}_{wt}, \quad \hat{n}_{wt} = \sum d \in D n_{dw}H_{dwt}.$$

$$\theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d}, \quad \hat{n}_d = \sum_{t \in T} \hat{n}_{dt}, \quad \hat{n}_{dt} = \sum w \in d n_{dw}H_{dwt}.$$

Покажем, что эти оценки действительно решают задачи (6).

Запишем лагранжиан задачи (6) при ограничениях нормировки, но проигнорировав ограничения неотрицательности (позже убедимся, что решение неотрицательно):

$$\mathrm{L}(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \varphi_{wt}\theta_{td} - \sum_{t \in T} \lambda_t \Big(\sum_{w \in W} \varphi_{wt} - 1 \Big) - \sum_{d \in D} \mu_d \Big(\sum_{t \in T} \theta_{td} - 1 \Big) =$$
$$= \sum_{d \in D} \sum_{w \in d} n_{dw} \ln p(w \mid d) - \sum_{t \in T} \lambda_t \Big(\sum_{w \in W} \varphi_{wt} - 1 \Big) - \sum_{d \in D} \mu_d \Big(\sum_{t \in T} \theta_{td} - 1 \Big).$$

Продифференцировав лагранжиан по φ_{wt} и приравняв к нулю производную, получим

$$\lambda_t = \sum_{d \in D} n_{dw} \frac{\theta_{td}}{p(w \mid d)}.$$

Домножим обе части этого равенства на φ_{wt}, просуммируем по всем терминам $w \in W$, применим услование нормировки вероятностей φ_{wt} в левой части и выделим переменную H_{dwt} в правой части. Получим

$$\lambda_t = \sum_{d \in D} \sum_{w \in W} n_{dw}H_{dwt}.$$

Снова домножим обе части (12) на φ_{wt}, выделим переменную H_{dwt} в правой части и выразим φ_{wt} из левой части, подставим уже известное выражение для λ_t. Получим

$$\varphi_{wt} = \frac{\sum\limits_{d \in D} n_{dw}H_{dwt}}{\sum\limits_{w' \in W} \sum\limits_{d \in D} n_{dw'}H_{dwt}}.$$

Преобразуем:

$$\varphi_{wt} = \frac{\sum\limits_{d \in D} n_{dw} H_{dwt}}{\sum\limits_{w' \in W} \sum\limits_{d \in D} n_{dw'} H_{dwt}} = \frac{\sum\limits_{d \in D} \hat{n}_{dwt}}{\sum\limits_{w' \in W} \sum\limits_{d \in D} \hat{n}_{dw't}} = \frac{\hat{n}_{wt}}{\sum\limits_{w' \in W} \sum\limits_{d \in D} \hat{n}_{dw't}} = \frac{\hat{n}_{wt}}{\hat{n}_t}.$$

Получили (10). Проделав аналогичные действия с производной лагранжиана по θ_{td}, получим (11).

Если начальные приближения θ_{td} и φ_{wt} положительны, то и после каждой итерации они будут оставаться положительными, несмотря на то, что ограничение неотрицательности было проигнорировано в ходе решения.

7.3.2 Эффективность EM-алгоритма

Число операций -- $O(n|T|I)$, где n -- длина коллекции $|T|$ -- число тем, I -- число итераций.

Перевбор всех терминов w во всех документах d можно организовать очень эффективно, если хранить каждый документ d в виде последовательности пар (w, n_{dw}).

7.3.3 Рациональный EM-алгоритм

Вычисление переменных $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$ на M-шаге требует однократного прохода всей коллекции в цикле по всем документам $d \in D$ и всем терминами $w \in d$. Внутри этого цикла переменные H_{dwt} можно вычислять непосредственно в тот момент, когда они понадобятся. От этого результат алгоритма не изменяется, E-шаг встраивается внутрь M -шага без дополонительных вычислительных затрат, отпадает необходимость хранения трёхмерной матрицы H_{dwt}. Заметим также, что переменную $\hat{n}_d$ можно не вычислять, поскольку $\hat{n}_d = n_d$.

Алгоритм 2.1. PLSA-EM: рациональный EM-алгоритм для модели PLSA.

Вход: коллекция документов D, число тем $|T|$, начальные приближения Θ, Φ;
Выход: распределения Θ и Φ;

повторять
  обнулить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$ для всех $d \in D,\ w \in W,\ t \in T$;
  для всех $d \in D,\ w \in d$
    $Z := \sum\limits_{t \in T} \varphi_{wt} \theta_{td}$;
    для всех $t \in T$ таких, что $\varphi_{wt}\theta_{td} > 0$
      увеличить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$ на $\delta = n_{dw}\varphi_{wt}\theta_{td}/Z$;
  $\varphi_{wt} := \hat{n}_{wt}/\hat{n}_t$ для всех $w \in W,\ t \in T$;
  $\theta_{td} := \hat{n}_{dt}/n_d$ для всех $d \in D,\ t \in T$;
пока Θ и Φ не сойдутся;

7.3.4 Обобщённый EM-алгоритм

Поскольку функционал правдопободия известен не точно, он зависит от приближённых значений H_{dwt}, полученных на E-шаге, нет необходимости сверхточно решать задачу

максимизации на M-шаге, достаточно ещё немного приблизиться к точке максимума правдоподобия и снова выполнить E-шаг.

В обобщённом EM-алгоритме (generalized EM-algorithm, GEM) сокращённый M-шаг.

В другом обобщении E-шаг выполняется для части скрытых переменных H_{dwt}. После этого M выполняется только для тех основных переменных φ_{wt}, θ_{td}, которые зависят от изменившихся скрытых переменных.

Алгоритм 2.2. PLSA-GEM: обобщённый EM-алгоритм для модели PLSA.

Вход: коллекция документов D, число тем $|T|$, начальные приближения Θ, Φ;
Выход: распределения Θ и Φ;

обнулить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$, $\hat{n}_d$, n_{dwt} для всех $d \in D,\ w \in W,\ t \in T$;
повторять
  для всех $d \in D,\ w \in d$
    $Z := \sum_{t \in T} \varphi_{wt}\theta_{td}$;
    для всех $t \in T$ таких, что $n_{dwt} > 0$ **или** $\varphi_{wt}\theta_{td} > 0$
      $\delta := n_{dw}\varphi_{wt}\theta_{td}/Z$;
      увеличить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$, $\hat{n}_d$ на $(\delta - n_{dwt})$;
      $n_{dwt} := \delta$;
    если пора обновить параметры Φ, Θ **то**
      $\varphi_{wt} := \hat{n}_{wt}/\hat{n}_t$ для всех $w \in W,\ t \in T$ таких, что $\hat{n}_{wt}$ изменился;
      $\theta_{td} := \hat{n}_{dt}/\hat{n}_d$ для всех $d \in D,\ t \in T$ таких, что $\hat{n}_{dt}$ изменился;
пока Θ и Φ не сойдутся;

Сокращение M-шага сводится к более частому обновлению параметров θ_{td} и φ_{wt}.

Параметры Φ, Θ пора обновить, когда:

- После каждого прохода коллекции; (тогда не суть не поменяется, а это медленно)
- После каждого документа;
- После каждого термина (d, w);
- После заданного числа терминов;
- После каждого вхождения термина;

На больших коллекциях частые обновления повышают скорость сходимости и почти не влияют на результат. Отсюда практическая рекомендация дделать обновления после каждого термина, при этом каждый термин документа обрабатывается только один раз. Этот способ позволяет ещё отказаться от матриц Θ и Φ, поскольку значения θ_{td} и φ_{wt} можно вычислять "на лету".

При первом проходе коллекции частые обновления не делаются, чтобы в счётчиках накопилась информация по всей коллекции. В противном случае оценки параметров θ_{td} и φ_{wt} по начальному фрагменту выборки могут оказаться хуже начального приближения. Начиная со второй итерации для каждой пары (d, w) из счётчиков $\hat{n}_{wt}$ и $\hat{n}_{dt}$ вычисляется n_{dwt} -- то самое δ, которое было к ним прибавлено при обработке пары (d, w) на предыдущей итерации. Т.о., счётчики $\hat{n}_{wt}$ и $\hat{n}_{dt}$ всегда содержат результат последнего однократного прохода всей матрицы.

Необходимость хранения трёхмерной матрицы n_{dwt} делает этот алгоритм неприменимым к большим коллекциям. Это можно устранить путём реорганизации итераций или применением сэмплирования.

7.4 Начальные приближения

Начальные приближения φ_t и θ_d можно задавать нормированными случайными векторами из равномерного распределения.

Другая распространённая рекомендация -- пройти по всей коллекции, выбрать для каждой пары (d, w) случайную тему t и вычислить частотные оценки (4) вероятностей φ_{wt} и θ_{td} для всех $d \in D$, $w \in W$, $t \in T$.

7.4.1 Инициализация с частичным обучением

Применяется в случаях, когда темы известны заранее и имеются дополнительные данные о привязке некоторых документов или терминов к темам. Учёт этих данных улучшает интерпретируемость тем.

Если известно, что документ d относится к подмножеству тем $T_d \subset T$, то в качестве начального θ_{td} можно взять равномерное распределение на этом подмножестве:

$$\theta_{td}^0 = \frac{1}{|T_d|}[t \in T_d].$$

Если известно, что подмножество терминов $W_t \subset W$ относится к теме t, то в качестве начального φ_{wt} можно взять равномерное распределение на W_t:

$$\varphi_{wt} = \frac{1}{|W_t|}[w \in W_t].$$

Если известно, что подмножество документов $D_t \subset D$ относится к теме t, то можно взять эмпирическое распределение слов в объединённом документе:

$$\theta_{wt}^0 = \frac{\sum\limits_{d \in D_t} n_{dw}}{\sum\limits_{d \in D_t} n_d}.$$

Если нет никакой априорной информации о связи документов с темами, то последнюю формулу можно применить к случайным подмножествам документов D_t, как вариант -- предлагается брать один случайный документ.

7.4.2 Инициализация Θ по Φ

Если для всех тем известны начальные приближения φ_{wt}^0, то первая итерация $\mathrm{E}\ \mathrm{M}$ -алгоритма при равномерном распределении $\theta_{td}^0 = 1/|T|$ даёт ещё одну интуитивно очевидную формулу инициализации:

$$\theta_{td} = \frac{1}{n_d}\sum_{w\in d} n_{dw}H_{dwt} = \sum_{w\in d}\frac{n_{dw}}{n_d}\frac{\varphi_{wt}}{\sum\limits_s \varphi_{ws}} = \sum_{w\in d}\hat{p}(w\,|\,d)\hat{p}(t\,|\,w).$$

Здесь распределение тем в документе d оценивается путём усреднения распределений тем $p(t\,|\,w)$ по словам документа d, вычисленных по формуле Байеса.

7.4.3 Недостатки PLSA

- Слишком много параметров φ_{wt} и θ_{td}: $(|W|\,|T| + |T|\,|D|)$.
- Неверно оценивает вероятность новых слов ($\hat{p}(w\,|\,t) = 0$ для слова, которого не было в обучающейся коллекции, но оно встретилось в каком-нибудь документе).

7.5 Дивергенция Кульбака-Лейблера (или KL -дивергенция)

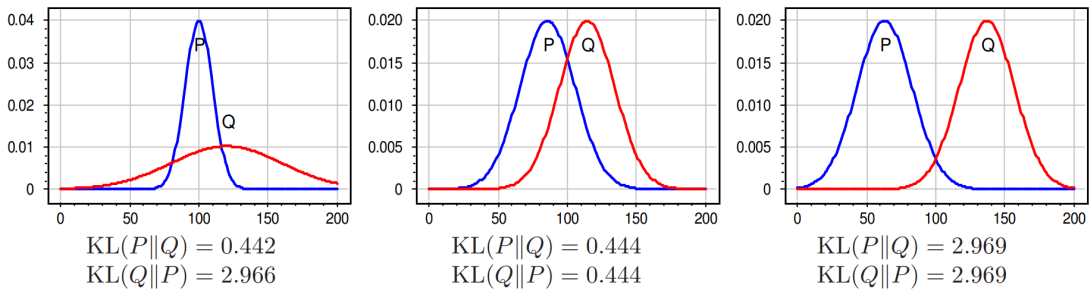
KL -дивергенция между дискретными распределениями $P = (p_i)_{i=1}^n$ и $Q = (q_i)_{i=1}^n$ -- несимметричная функция расстояния (и поэтому называть её фунцией расстояния -- некорректно):

$$\mathrm{KL}(P\,||\,Q) \equiv \mathrm{KL}_i(p_i\,||\,q_i) = \sum_{i=1}^n p_i \ln\frac{p_i}{q_i}.$$

Предполагается, что $p_i > 0$ и $q_i > 0$. KL -дивергенция не является вполне адекватной функцией расстояния, когда у распределений P и Q не совпадают носители $\Omega_P = i : p_i > 0$ и $\Omega_Q = i : q_i > 0$.

Наиболее важные свойства:

1. Неотрицательна. Если $\Omega_P = \Omega_Q$, то $\mathrm{KL}(P\,||\,Q) = \mathrm{KL}(Q\,||\,P) = 0 \Leftrightarrow p_i = q_i$ (когда распределения совпадают).
2. Является мерой вложенности распределений. Если $\mathrm{KL}(P\,||\,Q) < \mathrm{KL}(Q\,||\,P)$, то распределение P сильнее вложено в Q, чем Q в P.



3. Если P -- эмпирическая функция распределения, а $Q(\alpha)$ параметрическое семейство (модель) распределений, то минимизация KL -дивергенции эквивалентна максимизации правдоподобия:

$$\mathrm{KL}(P||Q(\alpha)) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i(\alpha)} \to \min_{\alpha} \Leftrightarrow \sum_{i=1}^{n} p_i \ln q_i(\alpha) \to \max_{\alpha}$$

Максимизация правдоподобия (6) эквивалентна минимизации взвешенной суммы дивергенций Кульбака-Лейблера между эмпирическими распределениями $\hat{p}(w|d) = n_{dw}/n_d$ и модельными $p(w|d)$, по всем документам $d \in D$:

$$\sum_{d \in D} n_d \mathrm{KL}_w \left(\frac{n_{dw}}{n_d} || \sum_{t \in T} \varphi_{wt} \theta_{td} \right) \to \min_{\Phi, \Theta},$$

где весом документа d является его длина n_d. Если веса n_d убрать, то все документы будут искусственно приведены к одинаковой длине. Такая модификация функционала качества может быть полезна при моделировании коллекций,содержащих документы одинаковой важности, но существенно разной длины.
