Машинное обучение. Кластеризация. Понижение размерности

Алексей Колесов

Белорусский государственный университет

22 ноября 2017 г.

Содержание

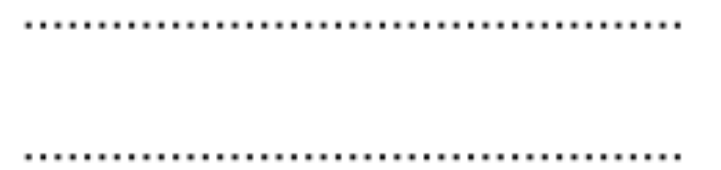
Кластеризация

- кластеризация — группировка множества объектов таким образом, чтоб похожие объекты были в одной группе (кластере), а непохожие — в разных
- применяется в анализе данных, как один из первых этапов
- магазины кластеризуют покупателей по покупкам; астрономы — звёзды по близости друг к другу; биологи — гены по их показателям в экспериментах

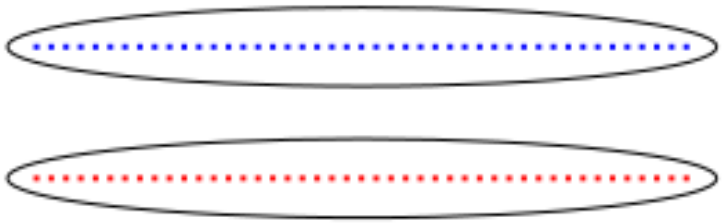
Сложность кластеризации

- кластеризация преследует две цели:
  - близкие объекты — в одном классе
  - далёкие — в разных
- близость — не транзитивное понятие
- разбиение на кластеры — отношение эквивалентности
- можно предложить последовательность $x_1, \ldots, x_m$, что x_i близка к соседям, но x_1 далёк от x_m

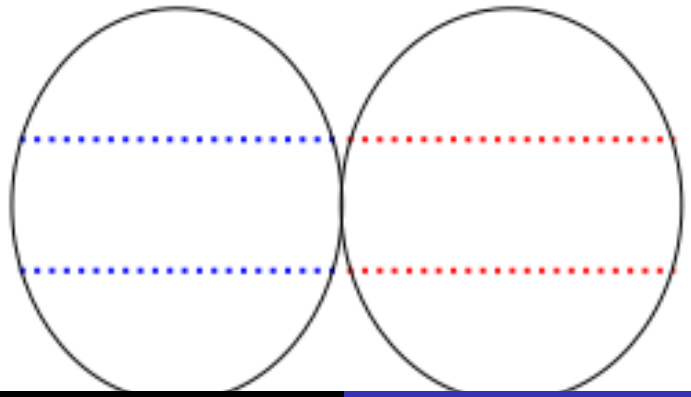
Пример



Близкие в одном:



Далёкие в разных:

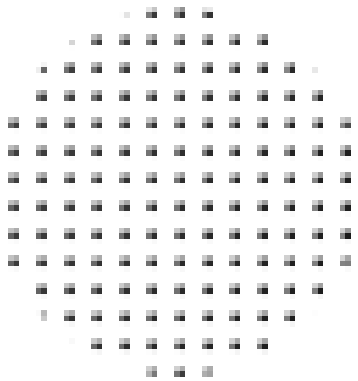
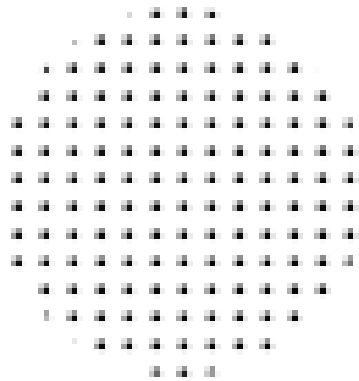
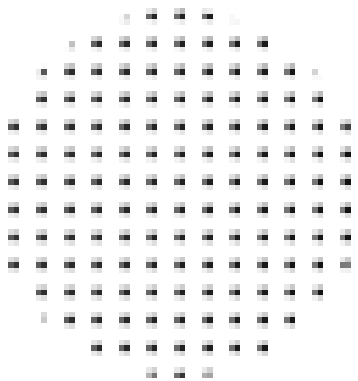
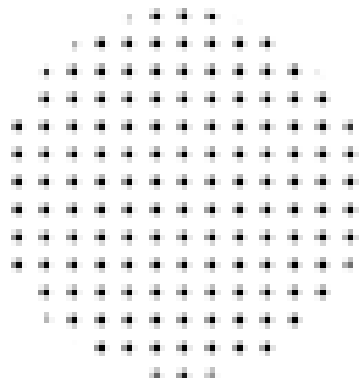


Отсутствие ground truth

- вторая проблема — отсутствие ground truth
- в supervised learning мы можем оценить качество решения по тренировочной выборке
- в кластеризации нет чёткого критерия успеха (что такое «правильная» кластеризация?)

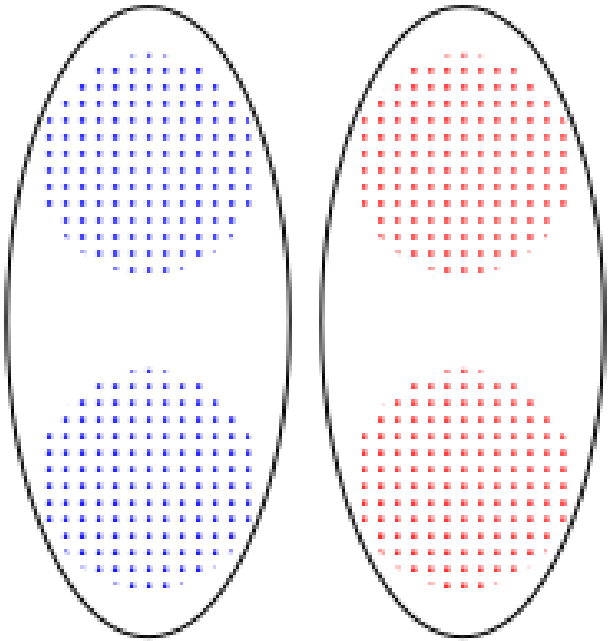
Отсутствие ground truth

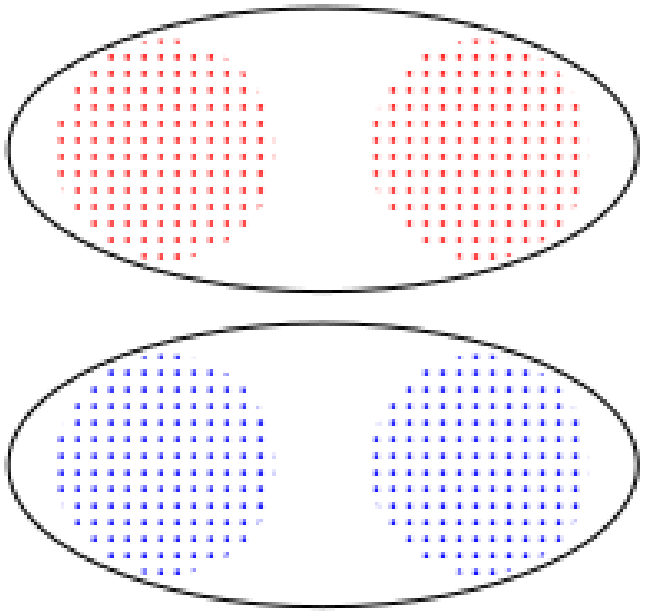
Пусть хотим кластеризовать:

Отсутствие ground truth

Какой вариант выбрать?





Отсутствие ground truth

Проблема встречается и в приложениях:

- как кластеризовать речь: по акценту или по содержанию?
- как кластеризовать фильмы: по жанру или по рейтингу?

Модель кластеризации

- **Вход**:
  - множество элементов X
  - функция расстояния на объектах: $d : X \times X \to \mathbb{R}_+$, $d(x, y) = d(y, x)$, $d(x, x) = 0$, $d(x, y) + d(y, z) \geqslant d(x, z)$
  - альтернативно может задаваться функция «похожести»: $s : X \times X \to [0, 1]$, $s(x, y) = s(y, x)$, $s(x, x) = 1$
  - некоторые алгоритмы также принимают k — количество кластеров
- **Выход**:
  - разбиение X на непересекающиеся кластера: $C = (C_1, \ldots, C_k)$
  - мягкая кластеризация: $p_i(x) = \mathbb{P}[x \in C_i]$
  - дендрограмма: дерево вложений кластеров

Linkage-based кластеризация

- Linkage-based кластеризация — самая простая кластеризация
- алгоритм состоит из последовательности раундов
- на нулевом раунде — каждый объект образует свой кластер
- на каждом раунде два самых близких кластера сливаются в один

Нужно определиться:

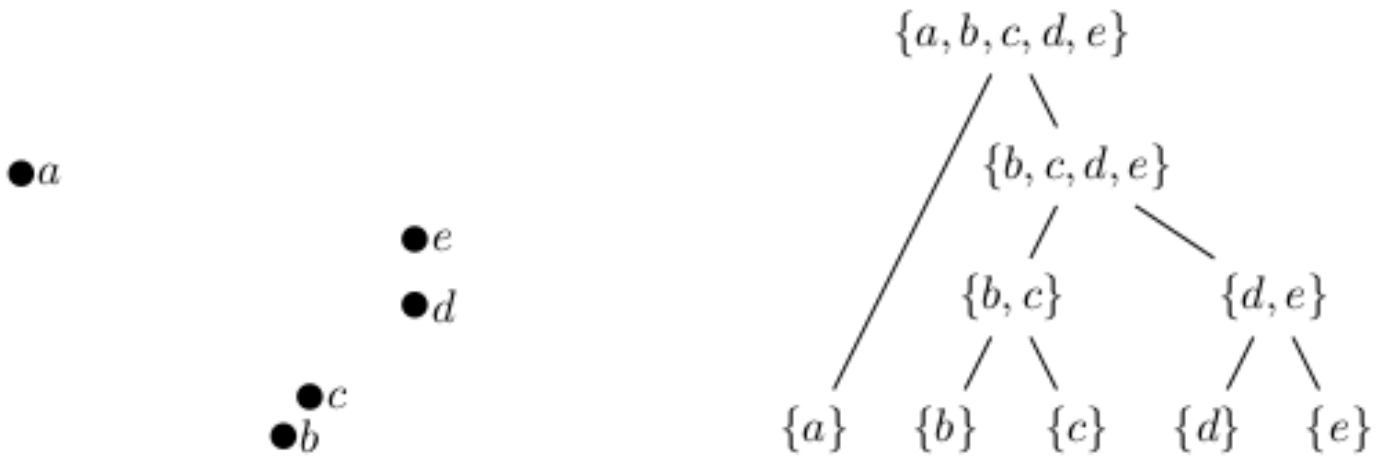
- как измерять расстояние между кластерами
- когда заканчивать алгоритм

Расстояние между кластерами

- Single linkage clustering:
  $D(A, B) = \min\{d(x, y) : x \in A, y \in B\}$
- Average linkage clustering: $D(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y)$
- Max Linkage clustring:
  $D(A, B) = \min\{d(x, y) : x \in A, y \in B\}$

Когда заканчивать

Если не останавливать алгоритм, то образуется дендрограмма:



Когда заканчивать

- фиксированное количество кластеров
- верхняя оценка на расстояние между кластерами
- шкалированная верхняя оценка: расстояние между кластерами не меньше, чем $\alpha \max\{d(x, y) : x, y \in X\}$, $\alpha \in (0, 1)$

Стоимость кластеризации

- можно задать стоимость кластеризации: $G : (X, d) \times C \to \mathbb{R}_+$
- тогда задача кластерезации: минимизация G
- получаем задачу оптимизации
- большинство полезных постановок NP-сложны
- некоторые NP-сложны даже для аппроксимации (например, k-means)

k-means

- в k-means каждый кластер C_i представляется своим центроидом: μ_i
- предполагается, что $\mu_i \in X'$, $X \subseteq X'$, d расширяется на X'
- $\mu_i(C_i) = \underset{\mu \in X'}{\operatorname{argmin}} \sum_{x \in C_i} d(x, \mu)^2$
- $G_{\text{k-means}}((X, d), (C_1, \ldots, C_k)) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i(C_i))^2$
- можно переписать:

$$G_{\text{k-means}}((X, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots, \mu_k \in X'} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

Часто применяемые стоимости

- $G_{\text{k-means}}(\ldots) = \min\limits_{\mu_1,\ldots,\mu_k \in X'} \sum\limits_{i=1}^{k} \sum\limits_{x \in C_i} d(x, \mu_i)^2$
- $G_{\text{k-medoids}}(\ldots) = \min\limits_{\mu_1,\ldots,\mu_k \in X} \sum\limits_{i=1}^{k} \sum\limits_{x \in C_i} d(x, \mu_i)^2$
- $G_{\text{k-medians}}(\ldots) = \min\limits_{\mu_1,\ldots,\mu_k \in X} \sum\limits_{i=1}^{k} \sum\limits_{x \in C_i} d(x, \mu_i)$
- $G_{\text{SOD}}(\ldots) = \min\limits_{\mu_1,\ldots,\mu_k \in X} \sum\limits_{i=1}^{k} \sum\limits_{x,y \in C_i} d(x, y)$

Center-based clustering

- k-means, k-medoids, k-medians являются center-based clustering функциями
- для решения достаточно определить центроиды кластеров: μ_i
- каждый x нужно отнести к ближайшему центроиду
- SOD (sum of in-cluster distances) таким не является

Алгоритм k-means

- k-means стоимость популярна на практике
- минимизация вычислительно сложна
- на практике применяют некоторый алгоритм, на качество которого нет хороших оценок
- рассмотрим на примере евклидового расстояния

Алгоритм k-means

Алгоритм 1 k-means

Вход: $X \subset \mathbb{R}^n$,

Вход: k — количество кластеров

1: Случайно выбрать начальные центроиды: $\mu_1, \ldots, \mu_k$
2: **while** не сошлось **do**
3: $C_i = \{x \in X : i = \text{argmin}_j \|x - \mu_j\|\}, \forall i \in [k]$
4: $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x, \forall i \in [k]$
5: **end while**
6: **return** центроиды: $\mu_1, \ldots, \mu_k$

Лемма о k-means

Лемма о k-means

На каждой итерации алгоритма k-means стоимость кластеризации не увеличивается

Доказательство

- если зафиксировать кластер C_i, то
  $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x = \text{argmin}_\mu \sum_{x \in C_i} \|x - \mu\|^2$
- строчка 4 не увеличивает стоимость
- строчка 3 не увеличивает стоимость, т.к. center-based функция

k-means

- нет хорошей оценки на количество шагов до сходимости
- нет оценки на разницу между оптимальным решением и полученным
- на практике, лучше несколько раз запускать алгоритм из разных точек

Спектральные методы кластеризации

- часто отношения между объекта представляются в виде взвешенного графа (similarity graph)
- $W_{i,j} = s(x_i, x_j)$, например $W_{i,j} = \exp(-d(x_i, x_j)^2/\sigma^2)$
- хотим, что рёбра между вершинами одного кластера были большими, а между разных — маленькими

Разрезы графа

- естественная формулировка задачи: минимизировать графовый разрез:

$$\text{cut}(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{r \in C_i, s \notin C_i} W_{r,s}$$

- для $k = 2$ есть эффективное решение
- к сожалению, зачастую это отрезать одну вершину
- можно сформулировать другой показатель:

$$\text{RatioCut}(C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{r \in C_i, s \notin C_i} W_{r,s}$$

- такая задача — вычислительно трудна

Лемма о спектральной кластеризации

Лемма о спектральной кластеризации

Пусть L (ненормализованный лапласиан графа) — матрица размера $m \times m$: $L = D - W$, где D — диагональная матрица с $D_{i,i} = \sum_{j=1}^{m} W_{i,j}$.

Пусть кроме этого $C_1, \ldots, C_k$ — кластеризация и $H \in \mathbb{R}^{m \times k}$ такая что:

$$H_{i,j} = \frac{1}{\sqrt{|C_j|}} 1_{i \in C_j}$$

Тогда $H^T H = I_k$ и $\text{RatioCut}(C_1, \ldots, C_k) = \text{trace}(H^T L H)$

Лемма о кластеризации

- для минимизации RatioCut можем искать матрицу H, такую что её элементы равны 0 или $\frac{1}{\sqrt{|C_j|}}$ и столбцы ортонормальны
- такая задача эффективно не решается
- можно просто искать ортонормальную матрицу, которая минимизирует $\text{trace}(H^T L H)$
- достаточно выбрать собственные вектора L, которые соответствуют наименьшим собственным значениям L

Unnormalized spectral clustering

Алгоритм 2 Unnormalized spectral clustering

Вход: $W \in \mathbb{R}^{m \times m}$,

Вход: k — количество кластеров

1: $L = W - D$

2: $U \in \mathbb{R}^{m \times k}$ — матрица со столбцами — собственные векторами L, соответствующие k наименьшим собственным значениям

3: $v_1, \ldots, v_m$ — строки U

4: Кластеризация $v_1, \ldots, v_m$ с помощью k-means

Аксиомы кластеризации

Пусть $F(X, d)$ — функция кластеризации (X – множество объектов, d – функция «непохожести»)

- **Scale Invariance** $\forall X, d$, $\alpha > 0$ должно выполняться $F(X, d) = F(x, \alpha d)$
- **Richness** для любого X и любого разбиения $C = (C_1, \ldots, C_k)$ должна существовать d, что $F(X, d) = C$
- **Consistency** для любых d и d', таких что для любых x, y выполняется, что если x и y в одном кластере в $F(x, d)$, то $d'(x, y) \leqslant d(x, y)$ и если x, y в разных в $F(X, d)$, то $d'(x, y) \geqslant d(x, y)$, то $F(X, d) = F(X, d')$

Теорема Kleinberg-a

Теорема Kleinberg-a

Не существует F, которая удовлетворяет одновременно: Scale Invariance, Richness и Consistency

Содержание

Методы понижения размерности

Понижение размерности — отображение данных высокой размерности в низкоразмерное пространство

- уменьшение вычислительной сложности
- улучшения обобщения (например, k-NN)
- повышение интерпретируемости данных

Рассматриваемые методы

- будем отображать данные из $\mathbb{R}^d$ в $\mathbb{R}^n$ $(n < d)$
- наиболее распространены линейные методы: $x \mapsto Wx$, где $W \in \mathbb{R}^{n \times d}$
- выбирать W стоит так, чтоб можно было «восстановить» x из Wx
- точное восстановление не всегда возможно

Рассматриваемые методы

- PCA: линейное восстановление
- случайные проекции: увидим, что «искажение» достаточно небольшое

Постановка

- $x_1, \ldots, x_m$ — m векторов из $\mathbb{R}^d$
- хотим уменьшить размерность векторов линейным преобразованием
- матрица $W \in \mathbb{R}^{n \times d}$ порождает отображение $x \mapsto Wx$
- можно *попытаться* восстановить линейным преобразованием
- $U \in \mathbb{R}^{d \times n}$: $y = Wx$, $\tilde{x} = Uy = UWx$

Задача PCA

Давайте решим:

$$\underset{W \in \mathbb{R}^{n \times d}, U \in \mathbb{R}^{d \times n}}{\operatorname{argmin}} \sum_{i=1}^{m} ||x_i - UWx_i||_2^2$$

Полученный метод носит название **Principal Component Analysis** (**метод главных компонент**). Изобретён Карлом Пирсоном в 1901-м году.

Лемма о форме решения

Лемма о форме решения PCA

Пусть (U, W) — искомые матрицы в задаче PCA. Тогда $U^T U = I_n$ и $W = U^T$.

Доказательство

- зафиксируем U, W; $R = \{UWx : x \in \mathbb{R}^d\}$ — линейное пространство размерности n
- пусть $V \in \mathbb{R}^{d \times n}$ — матрица, у которой столбцы ортонормированный базис R
- любой $x \in R$ может быть представлен как Vy, где $y \in \mathbb{R}^n$
- для любого $x \in \mathbb{R}^d$ и $y \in \mathbb{R}^n$ имеем:

$$||x - Vy||_2^2 = ||x||^2 + y^T V^T V y - 2y^T V^T x = ||x||^2 + ||y||^2 - 2y^T (V^T x)$$

Доказательство

- для любого $x \in \mathbb{R}^d$ и $y \in \mathbb{R}^n$ имеем:
$$||x - Vy||_2^2 = ||x||^2 + ||y||^2 - 2y^T(V^T x)$$
- минимизируя по y получаем, что $y = V^T x$
- т.е. для любого x: $VV^T x = \underset{\tilde{x} \in R}{\operatorname{argmin}} ||x - \tilde{x}||_2^2$
- $\sum_{i=1}^{m} ||x_i - UWx_i||_2^2 \geqslant \sum_{i=1}^{m} ||x_i - VV^T x_i||_2^2$ (выполняется для любых U, W)

Дальнейшие рассуждения

Можем заменить:

$$\underset{W \in \mathbb{R}^{n \times d}, U \in \mathbb{R}^{d \times n}}{\operatorname{argmin}} \sum_{i=1}^{m} ||x_i - UWx_i||_2^2$$

на

$$\underset{U \in \mathbb{R}^{d \times n}, U^T U = I_n}{\operatorname{argmin}} \sum_{i=1}^{m} ||x_i - UU^T x_i||_2^2$$

Дальнейшие рассуждения

$$||x - UU^T x||^2 = ||x||^2 - 2x^T UU^T x + x^T UU^T UU^T x \quad (1)$$
$$= ||x||^2 - x^T UU^T x \quad (2)$$
$$= ||x||^2 - \text{trace}(U^T xx^T U) \quad (3)$$

Т.к. trace — линейный оператор, то можем переписать:

$$\underset{U \in \mathbb{R}^{d \times n}, U^T U = I_n}{\text{argmin}} \text{trace}\left(U^T \sum_{i=1}^{m} x_i x_i^T U\right)$$

Дальнейшие рассуждения

Оптимизируем:

$$\underset{U \in \mathbb{R}^{d \times n}, U^T U = I_n}{\operatorname{argmin}} \operatorname{trace} \left(U^T \sum_{i=1}^{m} x_i x_i^T U \right)$$

- пусть $A = \sum_{i=1}^{m} x_i x_i^T$; она симметричная, положительно полуопределённая
- можно представить $A = VDV^T$, где D — диагональная, $V^T V = VV^T = I_d$
- диагональ D — собственные значения A, столбцы V — соответствующие собственные вектора
- без потери общности: $D_{1,1} \geqslant \ldots \geqslant D_{d,d}$
- $D_{d,d} \geqslant 0$

Решение PCA

Решение PCA

Пусть $x_1, \ldots, x_m$ вектора из $\mathbb{R}^d$, $A = \sum_{i=1}^{m} x_i x_i^T$, пусть $u_1, \ldots, u_n$ — n собственных векторов A, соответствующие n наибольшим собственным значениям A. Тогда решение задачи PCA — взять U матрицу, колонки которой — вектора $u_1, \ldots, u_n$, а $W = U^T$

Доказательство

- $A = VDV^T$ (спектральное разложение)
- пусть $U \in \mathbb{R}^{d \times n}$, такая что $U^T U = I_n$
- пусть $B = V^T U \Rightarrow VB = VV^T U = U$, а значит:
$$U^T AU = B^T V^T VDV^T VB = B^T DB$$
- поэтому:
$$\text{trace}(U^T AU) = \sum_{j=1}^{d} D_{j,j} \sum_{i=1}^{n} B_{j,i}^2$$
- $B^T B = U^T VV^T U = U^T U = I$, т.е. столбцы B ортонормированы, а значит $\sum_{j=1}^{d} \sum_{i=1}^{n} B_{j,i}^2 = n$
- $\sum_{i=1}^{n} B_{j,i}^2 \leqslant 1$

Доказательство

Имеем:

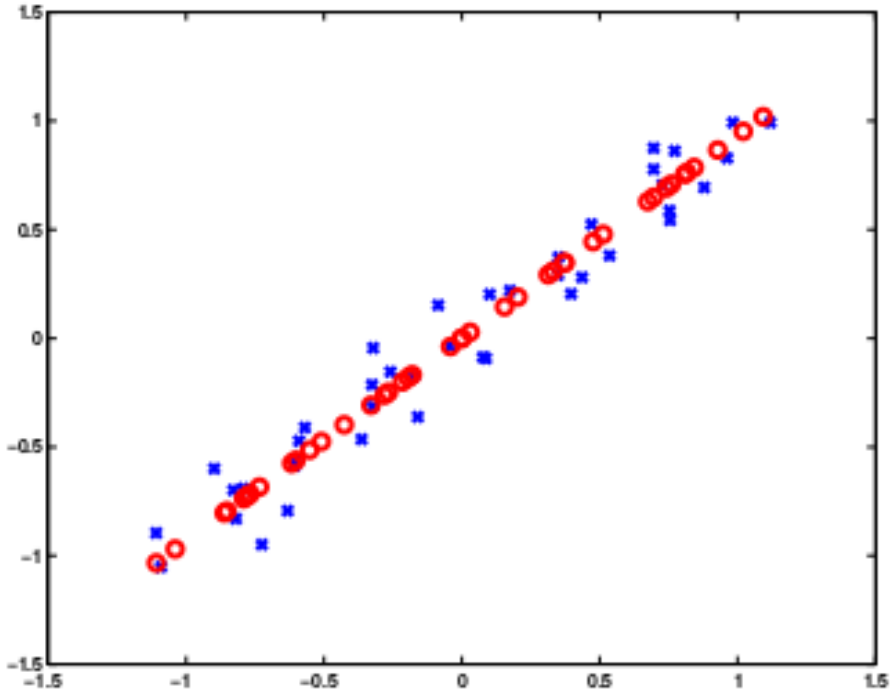
$$\text{trace}(U^T A U) \leqslant \max_{\beta \in [0,1]^d : ||\beta||_1 \leqslant n} \sum_{j=1}^{d} D_{j,j} \beta_j$$

- правая часть равна $\sum_{j=1}^{n} D_{j,j}$
- т.е. для подходящих $U \in \mathbb{R}^{d \times n}$: $\text{trace}(U^T A U) \leqslant \sum_{j=1}^{d} D_{j,j}$
- но если мы возьмём в качестве U собственные вектора A, то получим $\text{trace}(U^T A U) = \sum_{j=1}^{d} D_{j,j}$

Вычислительная сложность

- сложность нахождения значений $\mathcal{O}(d^3)$ и сложность составления A: $\mathcal{O}(md^2)$
- если d сильно больше m, то есть вариант получше
- $A = X^T X$; рассмотрим $B = XX^T$
- пусть u — собственный вектор $B \Rightarrow Bu = \lambda u \Rightarrow X^T(XX^T)u = \lambda X^T u \Rightarrow A(X^T u) = \lambda(X^T u) \Rightarrow \frac{X^T u}{||X^T u||}$ — собственный вектор A
- в таком варианте нужно $\mathcal{O}(m^3)$ на нахождение собственных значений и $\mathcal{O}(m^2 d)$ на составление матрицы B

Иллюстрация



Алгоритм

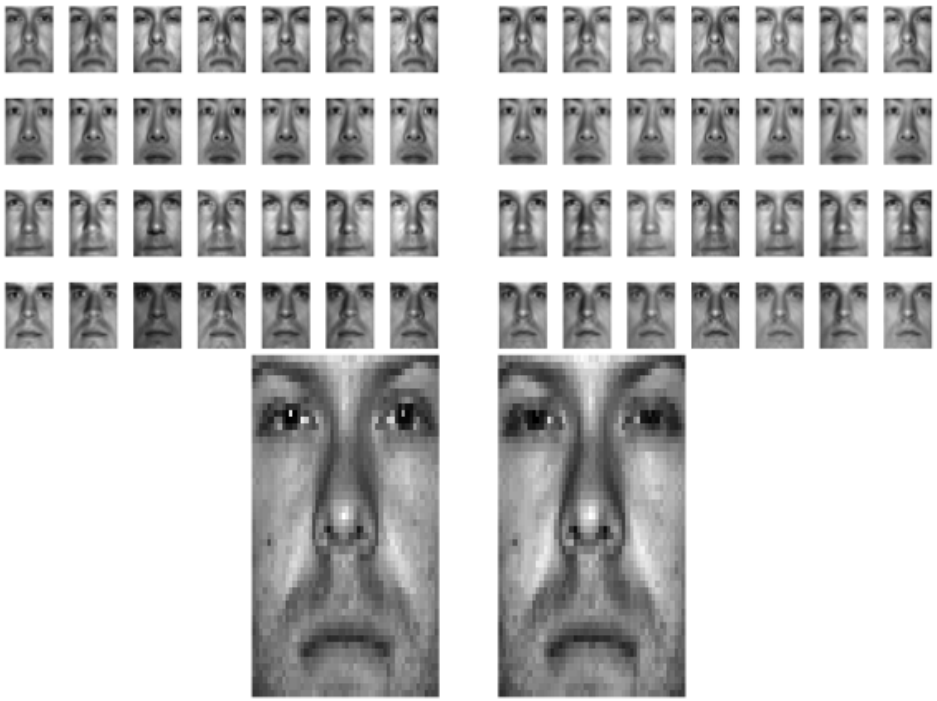
Алгоритм 3 PCA

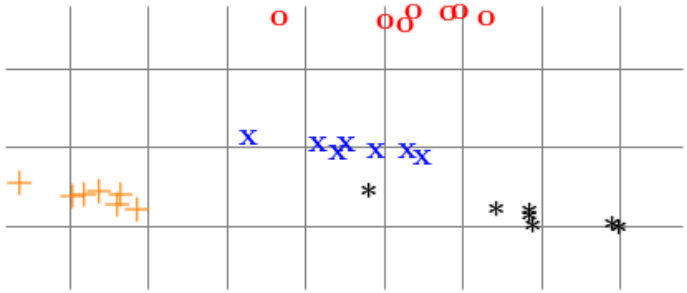
Вход: $X \in \mathbb{R}^{m \times d}$

Вход: n — количество компонент

1: **if** $m > d$ **then**
2: $A = X^T X$
3: $u_1, \ldots, u_n$ — собственные вектора A с наибольшими значениями
4: **else**
5: $B = XX^T$
6: $v_1, \ldots, v_n$ — собственные вектора B с наибольшими значениями
7: $u_i = \frac{X^T v_i}{||X^T v_i||} \ \forall i$
8: **end if**
9: **return** матрицу со столбцами $u_1 \ldots u_n$

Применение к лицам





Случайные проекции

- покажем, что домножение на случайную матрицу не сильно «искажает» изображение
- если выбрать W из правильного распределения, то с высокой вероятностью, евклидовы расстояния между точками изменятся несильно

Формализация

- пусть x_1, x_2 — два вектора из $\mathbb{R}^d$
- будем говорить, что W несильно искажает расстояния, если $\frac{||Wx_1 - Wx_2||}{||x_1 - x_2||}$ близко к единице
- достаточно исследовать изменение нормы $x = x_1 - x_2$ (т.е. $\frac{||Wx||}{||x||}$)

Лемма об искажении

Лемма об искажении

Пусть $x \in \mathbb{R}^d$ и $W \in \mathbb{R}^{n,d}$ — случайная матрица, такая что $W_{i,j}$ — случайные независимые нормальные величины (нулевое МО, единичная дисперсия). Тогда для любого $\epsilon \in (0, 3)$ выполняется:

$$\mathbb{P}\left[\left|\frac{\|(1/\sqrt{n})Wx\|^2}{|x|^2} - 1\right| > \epsilon\right] \leqslant 2e^{-\epsilon^2 n/6}$$

Доказательство

- можно считать, что $\|x\|^2 = 1$, тогда нужно доказать, что
$$\mathbb{P}[(1-\epsilon)n \leqslant \|Wx\|^2 \leqslant (1+e)n] \geqslant 1 - 2e^{-\epsilon^2 n/6}$$
- пусть w_i — строка W
- $\langle w_i, x\rangle$ — взвешенная сумма d стандартных нормальных величин, т.е. нормальна распределена с МО 0 и дисперсией $\sum_j x_j^2 = \|x\|^2 = 1$
- случайная величина $\|Wx\|^2 = \sum_{i=1}^{n} (\langle w_i, x\rangle)^2$ имеет ? распределение

Доказательство

- можно считать, что $\|x\|^2 = 1$, тогда нужно доказать, что
$$\mathbb{P}[(1-\epsilon)n \leqslant \|Wx\|^2 \leqslant (1+e)n] \geqslant 1 - 2e^{-\epsilon^2 n/6}$$
- пусть w_i — строка W
- $\langle w_i, x\rangle$ — взвешенная сумма d стандартных нормальных величин, т.е. нормальна распределена с МО 0 и дисперсией $\sum_j x_j^2 = \|x\|^2 = 1$
- случайная величина $\|Wx\|^2 = \sum_{i=1}^{n} (\langle w_i, x\rangle)^2$ имеет χ_n^2 распределение

Лемма Джонсона-Линденштрауса

Лемма Джонсона-Линденштрауса

Пусть Q — конечное множество векторов из $\mathbb{R}^d$. Тогда для $\delta \in (0, 1)$ и натурального n зафиксируем:

$$\epsilon = \sqrt{\frac{6 \log(2|Q|/\delta)}{n}} \leqslant 3$$

Тогда с вероятностью не меньше $1 - \delta$ при выборе $W \in \mathbb{R}^{n \times d}$, такой что каждый элемент W распределён нормально с нулевым матожиданием и дисперсией $1/n$ мы получаем:

$$\sup_{x \in Q} \left| \frac{\|Wx\|^2}{\|x\|^2} - 1 \right| < \epsilon$$

Содержание

Итоги

- разобрали различные методы кластеризации
- рассмотрели задачу понижения размерности

Литература

- Shai Shalev-Shwartz and Shai Ben-David — Understanding Machine Learning: From theory to algorithms (главы 22-23)