

Методы современной прикладной статистики

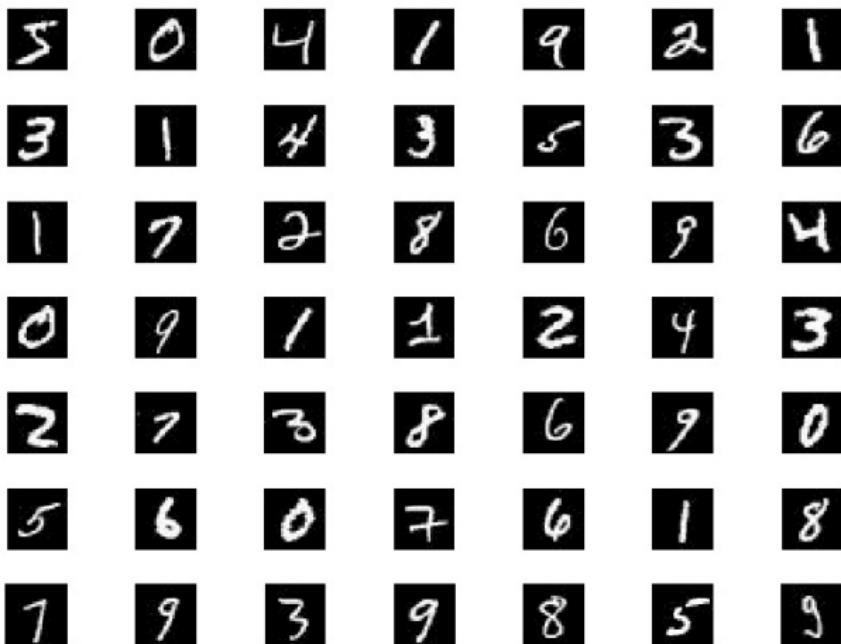
11. Методы понижения размерности.

Родионов Игорь Владимирович
vecsell@gmail.com

ШАД, 2020

Пример

Задача: распознать цифры на картинке, для чего нужно выбрать компактное признаковое описание изображения.



Выбор признакового пространства

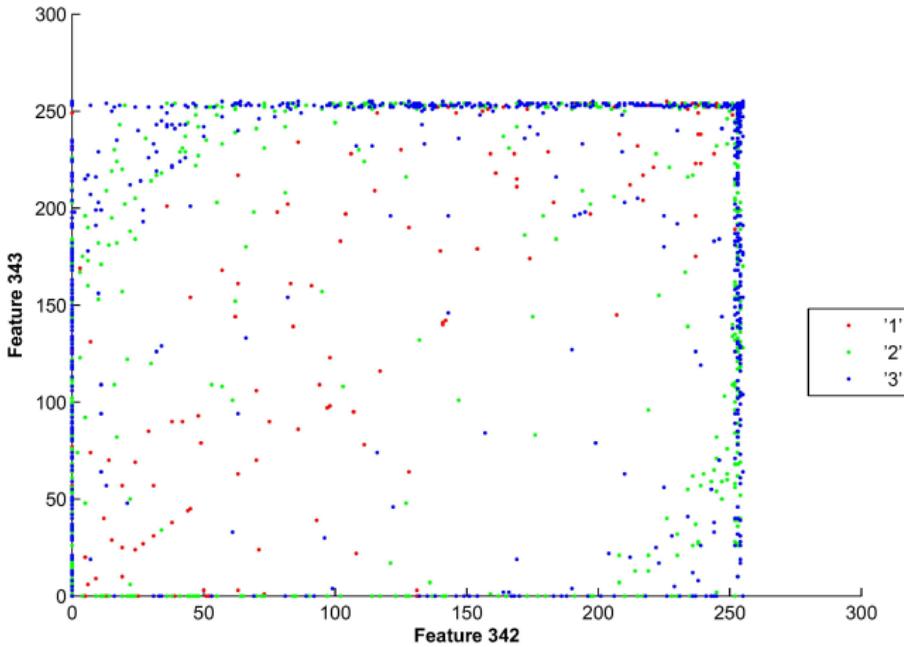
Прямой способ решения: вытянуть интенсивности каждой точки изображения в вектор.

Такой способ описания данных не является адекватным по следующим причинам:

- Слишком большое число признаков (для картинки $28 \times 28 - 784$ признака). Как следствие, большое время обработки, проблемы переобучения и т.д.
- Близкие в пространстве признаков объекты часто лежат в разных классах. Гипотеза компактности является одним из основных предположений большинства методов распознавания.

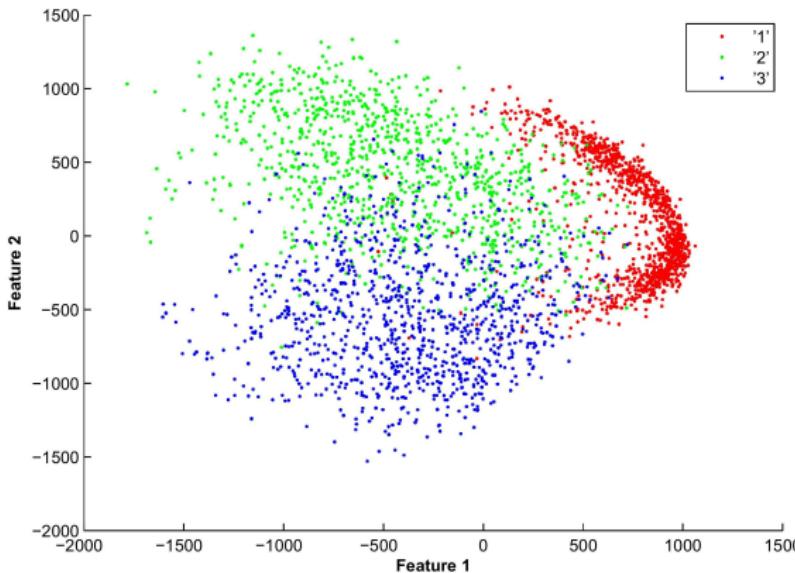
Гипотеза компактности

Близкие в пространстве признаков объекты не являются объектами одного класса.



Преобразование пространства признаков

После применения метода главных компонент количество признаков уменьшается (с 784 до нескольких десятков), причем объекты одного класса образуют компактные области в пространстве признаков.



Снижение размерности данных

Пусть имеются признаки $X = \{X_j\}_{j=1}^k$, $X_j \in \mathbb{R}^n \forall j$. Цель – выбрать пространство меньшей размерности $m < k$ так, чтобы схожие объекты оставались схожими и образовывали компактные области. Причины понижения размерности

- уменьшение вычислительных затрат;
- борьба с переобучением;
- сжатие данных для более эффективного хранения информации;
- визуализация и интерпретация данных;
- извлечение признаков;
- борьба с мультиколлинеарностью;
- ...

Метод главных компонент (PCA)

Метод является ещё одним способом борьбы с проблемой мультиколлинеарности признаков в задаче линейной регрессии. Идея метода состоит в том, чтобы подвергнуть исходные признаки некому линейному преобразованию так, чтобы новые признаки были ортогональны, а число их уменьшилось.

Несмотря на очевидные плюсы метода, он обладает и недостатками: в частности, новые признаки зачастую не являются наглядными. Также метод дает не самые лучшие результаты, если данные расположены на неком многообразии, плохо приближающемся гиперплоскостью.

Постановка задачи

Пусть X_1, \dots, X_k – исходные признаки модели размерности n (т.е. значения признаков посчитаны на n объектах).

Обозначим через $X = \|x_{ij}\|$ матрицу признаков размера $n \times k$. Хотим перейти от признаков X к новым признакам $Z = \|z_{ij}\|$, где Z – матрица размера $n \times m$, $m < k$.

Кроме того, потребуем, чтобы старые признаки по новым восстанавливались с приемлемой точностью, т.е. \exists

$U_m = \|u_{ij}\|$ размера $k \times m$ такая, что $\hat{X}_j = \sum_{l=1}^m u_{jl} Z_l$ было бы хорошим приближением признака X_j . Получается, мы хотим решить следующую задачу

$$\Delta^2(Z, U_m) = \sum_{j=1}^k (\hat{X}_j - X_j)^2 = \|ZU_m^T - X\|^2 \rightarrow \min_{U_m, Z} .$$

Свойства метода

Теорема. Если $m < \text{rank}(X)$, то минимум $\Delta^2(Z, U_m)$ достигается, когда столбцы матрицы U_m есть собственные векторы матрицы $X^T X$, соответствующие m максимальным собственным значениям этой матрицы. При этом $Z = XU_m$, а матрицы U_m и Z – ортогональны.

Свойства матриц U_m и Z :

- 1) Матрица U_m ортонормирована, т.е. $U_m^T U_m = I_m$;
- 2) $Z^T Z = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$, где $\lambda_1 \geq \dots \geq \lambda_m - m$ максимальных собственных значений матрицы $X^T X$.
- 3) $X^T X U_m = U_m \Lambda$, $XX^T Z = Z\Lambda$;
- 4) $\|ZU_m^T - X\|^2 = \|X\|^2 - \text{tr}\Lambda = \sum_{j=m+1}^k \lambda_j$.

Свойства метода

Главные компоненты – это собственные векторы матрицы $X^T X$, соответствующие m максимальным собственным значениям этой матрицы. В качестве новых признаков $\{Z_j\}_{j=1}^m$ модели мы выбираем проекции старых объектов на эти собственные векторы. Тогда

5) Проекции объектов на первую главную компоненту c_1 имеют наибольшую выборочную дисперсию среди проекций на всевозможные направления $d \in \mathbb{R}^k$. Далее, $\forall j \geq 2$ проекции объектов на c_j – j -тую главную компоненту – имеют наибольшую выборочную дисперсию среди проекций на всевозможные направления $d \in \mathbb{R}^k$, перпендикулярные c_1, \dots, c_{j-1} .

Выбор количества признаков

Из свойства 4 вытекает, что чем меньше

$E_m = \sum_{j=m+1}^k \lambda_j / \sum_{j=1}^k \lambda_j$, тем лучше новые признаки приближают старые.

Отсюда количество новых признаков можно выбирать так:

$\tilde{m} := \min_m \{E_m < \varepsilon\}$, \tilde{m} называется эффективной размерностью пространства признаков X .

Также m можно выбрать с помощью **метода крутого склона**: если E_{m+1} достаточно мало и $E_m \gg E_{m+1}$, то в качестве эффективной размерности берем m .

Наконец, **метод сломанной трости** состоит в следующем. Обозначим $C = \sum_{j=1}^k \lambda_j$, введем $I_i = \frac{1}{n} \sum_{j=i}^n \frac{1}{j}$. Тогда

$$\overline{m} = \inf \left\{ k : \frac{\lambda_k}{C} < I_k \right\}.$$

Подготовка данных к применению метода

- 1) Выбросы могут сильно помешать работе алгоритма (потому что метод линейный), поэтому стоит их удалить.
- 2) Если 2 признака имеют очень большую корреляцию ($\rho > 0.95$), то один из признаков тоже стоит удалить, иначе матрицы будут плохо обращаться.
- 3) Если признаки – в различных шкалах (например, литры и метры), то стандартизация данных **обязательна!**

Задача регрессии в новых признаках

Посмотрим, как будет выглядеть постановка задачи линейной регрессии для новых признаков. Заменяя X на ZU_m^T , имеем

$$\|Y - ZU_m^T\beta\|^2 = \|Y - Z\theta\|^2 \rightarrow \min_{\theta},$$

где $\theta = U_m^T\beta$.

Тогда решение записывается как $\hat{\theta} = \Lambda^{-1}Z^TY = D_mV^TY$,

$$\hat{\beta} = U_mD_mV^TY = \sum_{j=1}^m \frac{1}{\sqrt{\lambda_j}} u_j(v_j^TY),$$

где $D_m = \text{diag}\{(\lambda_1)^{-1/2}, \dots, (\lambda_m)^{-1/2}\}$ – матрица размера $m \times n$, а матрица V появляется в сингулярном разложении, к которому сейчас и перейдем.

Сингулярное разложение (SVD)

Известно, что любая матрица размера $n \times k$ (ранга k) представима в виде $X = VDU^T$, где

- 1) V – ортогональная матрица размера $n \times n$, ее столбцы – собственные векторы матрицы XX^T ;
- 2) D – матрица размера $n \times k$, $d_{ii} = \sqrt{\lambda_i}$, $d_{ij} = 0$, если $i \neq j$, где $\{\lambda_i\}_{i=1}^k$ – собственные числа матрицы X^TX (и ненулевые собственные значения матрицы XX^T);
- 3) U – ортогональная матрица размера $k \times k$, её столбцы – собственные векторы матрицы X^TX .

Сингулярное разложение

Тогда решение метода наименьших квадратов линейной регрессии выглядит так:

$$\begin{aligned}\hat{\beta} &= (X^T X)^{-1} X^T Y = (U D^T V^T V D U^T)^{-1} U D^T V^T Y = \\ U(D^T)^{-1} V^T Y &= \sum_{j=1}^k \frac{1}{\sqrt{\lambda_j}} u_j (V^T Y)_j,\end{aligned}$$

где $(D^T)^{-1}$ – матрица размера $k \times n$, на диагонали которой стоят числа $\frac{1}{\sqrt{\lambda_i}}$, $i = 1, \dots, k$, а все остальные элементы нулевые.

Если есть сингулярное разложение, то в методе главных компонент $Z = V \sqrt{D_m^T}$, где D_m – матрица размера $m \times n$, на диагонали которой стоят числа $\frac{1}{\sqrt{\lambda_i}}$, $i = 1, \dots, m$, а все остальные элементы нулевые, а U_m – это матрица U , у которой убрали $k - m$ последних столбцов.

Степенной метод

В случае, если n и k являются большими, то довольно сложно найти собственные числа и собственные векторы матрицы $X^T X$. В этом случае, чтобы понизить размерность, можно воспользоваться степенным методом вычисления главных компонент.

Пусть есть произвольная симметричная матрица A размера n , а её собственные числа $|\lambda_1| \geq |\lambda_2| \geq \dots$, c_1 – собственный вектор, соответствующий λ_1 , и для некоторого $x_0 \in \mathbb{R}^n$ скалярное произведение $(c_1, x_0) \neq 0$. Положим $x_{k+1} = Ax_k$, $k \geq 0$.

Теорема.

$$t_{k+1} = \frac{(x_k, x_{k+1})}{|x_k|^2} \rightarrow \lambda_1 \text{ при } k \rightarrow \infty.$$

Степенной метод

Алгоритм вычисления λ_k и c_k (в предположении, что $\{\lambda_i\}_{i=1}^{k-1}$ и $\{c_i\}_{i=1}^{k-1}$ известны или уже найдены):

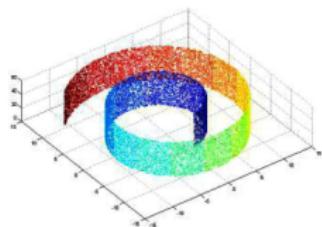
- 1) выбираем произвольное $x_0 \in \mathbb{R}^n$, кладем $t_0 = 0$;
- 2) (для $i \geq 1$ и $k \geq 2$) ортогонализируем x_i по векторам (c_1, \dots, c_{k-1}) : $y_i = x_i - (c_1, x_i)c_1 - \dots - (c_{k-1}, x_i)c_{k-1}$;
- 3) нормируем y_i : $e_i = y_i / |y_i|$;
- 4) вычисляем $x_{i+1} = Ae_i$ и $t_{i+1} = (x_{i+1}, e_i)$;
- 5) нормируем x_{i+1} : $z_{i+1} = x_{i+1} / |x_{i+1}|$;
- 6) если $|t_i - t_{i+1}| \leq \varepsilon$, то полагаем $t_k = e_{i+1}$ и $c_k = z_{i+1}$ и заканчиваем процесс; если нет, то полагаем $x_{i+1} = z_{i+1}$ и возвращаемся к шагу 2.

Скорость сходимости алгоритма экспоненциальная.

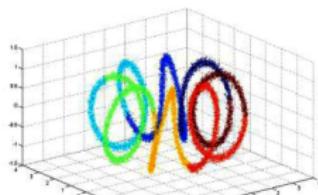
Недостатки метода главных компонент

- 1) Метод главных компонент способен находить только линейные подпространства исходного пространства, которые “объясняют” данные с высокой точностью. На практике поверхность, вдоль которой располагаются данные, может сильно отличаться от линейной.
- 2) PCA является инвариантным относительно поворота координат в пространстве переменных. Это означает, что восстановление значений исходных переменных является неоднозначным.
- 3) Если некоторые собственные значения матрицы $X^T X$, совпали, то новое пространство признаков может определяться неоднозначно.

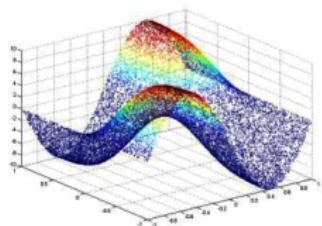
Нелинейные методы понижения размерности



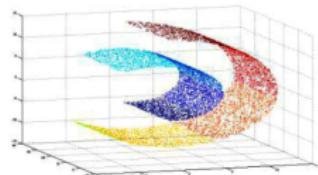
(a) Swiss roll dataset.



(b) Helix dataset.



(c) Twinpeaks dataset.



(d) Broken Swiss roll dataset.

Типовые датасеты, на которых проверяется качество нелинейных алгоритмов понижения размерности и на которых РСА плохо работает. Реальные данные бывают устроены гораздо сложнее.

Ядерный PCA (Kernel PCA)

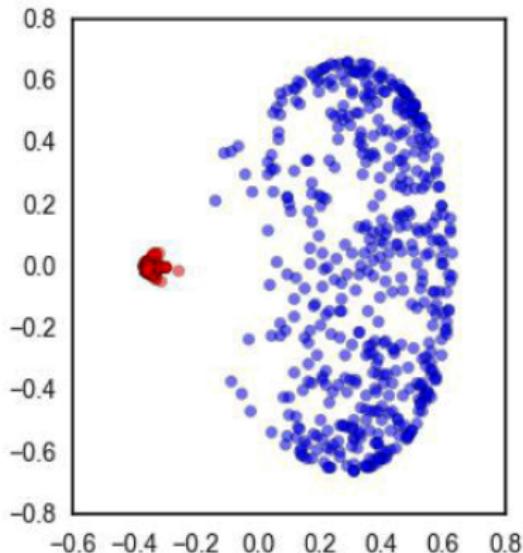
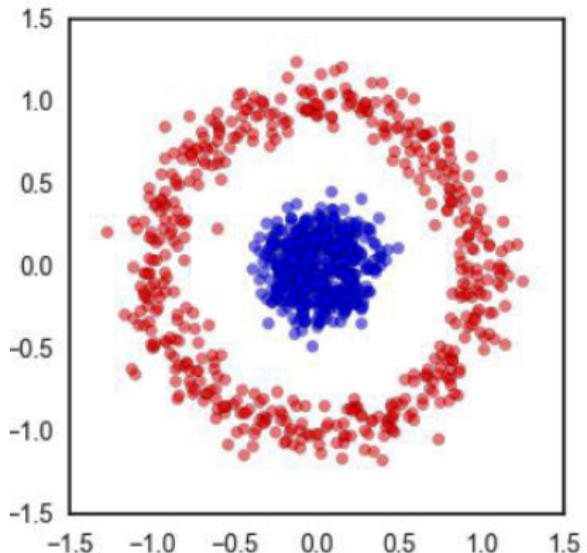
Kernel PCA основан на том же принципе, что и обычный PCA, однако он позволяет создавать новые признаки с помощью более сложных, нелинейных преобразований старых признаков.

Заметим, что в PCA мы работаем только с матрицей $X^T X$, которая (при условии стандартизации X) является матрицей корреляций признаков X_1, \dots, X_k .

Если мы вместо обычного скалярного произведения (x, y) в пространстве \mathbb{R}^n в PCA рассмотрим скалярное произведение $K(x, y) = (\Phi(x), \Phi(y))$, где $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^N$, $N > n$, то как раз и получим метод Kernel PCA. Для его применения знать явный вид Φ не нужно.

Зачем нужны ядра (повышение размерности)

Иногда данные в задаче не являются линейно разделимыми. Однако, после некоторого нелинейного преобразования признаков удается разделить данные прямой. Ядро - это и есть то самое нелинейное преобразование.



Виды ядер

- Линейное (приводит к обычному PCA);
- Полиномиальные (например, квадратичное ядро $K(x, y) = ((x, y) + 1)^2$);
- Гауссово (RBF) $K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$;
-

Многомерное шкалирование (MDS)

Многомерное шкалирование (multi-dimensional scaling), как и метод главных компонент, относится к так называемым глобальным методам, т.е. сохраняющим глобальные свойства объектов.

Цель метода: минимизировать суммарное расхождение между расстояниями между объектами, $d_{ij} = \rho(x_i, x_j)$, и расстояниями между объектами в новом пространстве признаков, $\delta_{ij} = \rho(z_i, z_j)$.

Выделяют подходы

1) классический: $F_0 = \sum_{i < j} (d_{ij} - \delta_{ij})^2 \rightarrow \min_{\delta}$;

2) метод Сэммона: $F_1 = \sum_{i < j} \frac{(d_{ij} - \delta_{ij})^2}{d_{ij}} \rightarrow \min_{\delta}$.

- 1) Прежде чем применять метод многомерного шкалирования, нужно стандартизировать данные, если они померяны в разных шкалах.
- 2) Возможные виды расстояний: а) евклидово; б) метрика Манхэттена: $\rho(x, y) = \sum_i |x_i - y_i|$, в) равномерная метрика: $\rho(x, y) = \max_i |x_i - y_i|$ и другие.
- 3) В случае, если мы используем классическое MDS для евклидовой метрики, то результат метода будет совпадать с применением PCA.
- 4) Метод Сэммона более точно передает небольшие различия и менее точно – большие (действительно, ведь при отображении больших расстояний допустимы большие ошибки). Также метод устойчив к наличию пропусков в данных.

- 5) Готового способа выяснить, какие признаки вносят вклад в новые признаки и коррелируют с ними, нет.
- 6) Выбирать размерность пространства новых признаков m можно следующим образом: если величина stress $= \sqrt{F_0 / \sum_{i < j} \delta_{ij}^2} < 0.2$, то приближение считается приемлемым.
- 7) Минимум функционалов F_0 и F_1 ищется градиентными методами. Начальное приближение для них можно выбрать так: либо спроектировать исходное пространство признаков на произвольное подпространство размерности m , либо воспользоваться для выбора этого подпространства PCA.

Метод сопряженных градиентов

Хотим найти $x_{min} = \arg \min_x F(x)$, $x \in \mathbb{R}^{nm}$.

1) Определим \bar{p}_t в пространстве \mathbb{R}^{nm} по формулам

$$p_1 = -g_1, \quad p_t = -g_t + \beta_t p_{t-1} \text{ при } t \geq 2,$$

где p_{t-1} – направление на предыдущем шаге,

$$\beta_t = \|g_t/g_{t-1}\|^2, \quad \text{а } g_t = \left(\frac{\partial F}{\partial x_{11}}, \dots, \frac{\partial F}{\partial x_{nm}} \right) \Big|_{x=x_t}.$$

2) Перемещаемся из x_t в x_{t+1} по следующему правилу:

$$x_{t+1} = x_t + \alpha_t p_t,$$

где $\alpha_t = \arg \min_{\alpha > 0} F(x_t + \alpha p_t)$.

3) Если $|g_{t+1}| < \varepsilon$, то заканчиваем.

Метод сопряженных градиентов

1) На втором шаге алгоритма можно выбрать

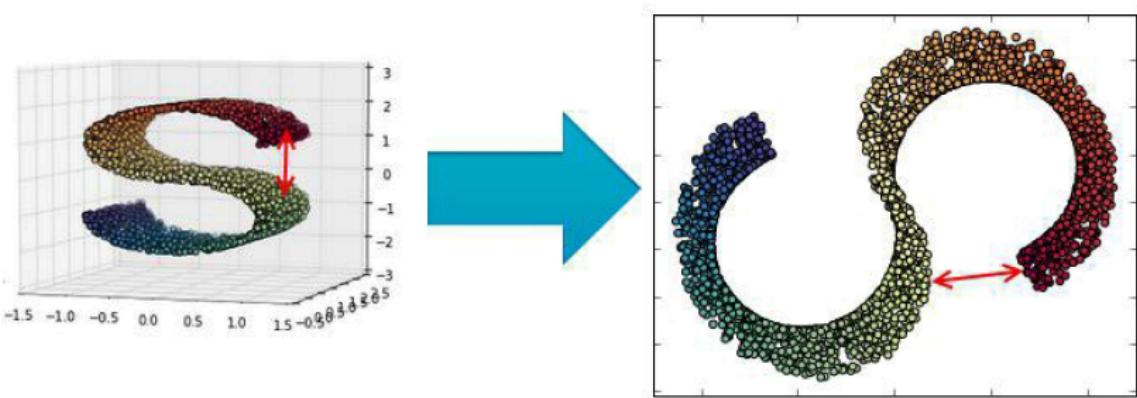
$$\alpha_t = \frac{((x_t - x_{t-1}), (g_t - g_{t-1}))}{\|g_t - g_{t-1}\|^2}.$$

2) Если брать на первом шаге $\beta_t = 0$ (метод наискорейшего спуска), то процесс рискует никогда не остановиться.

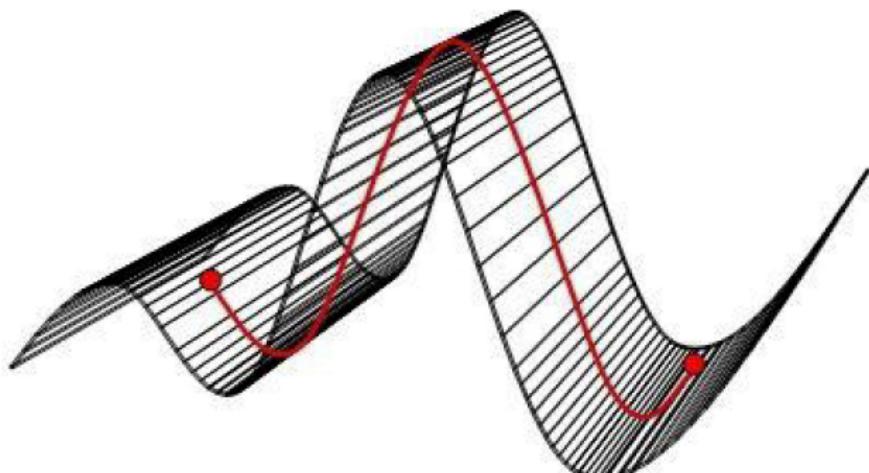
3) В **методе стохастического градиента** на t -м шаге алгоритма берем $g_t = \left(0, \dots, 0, \frac{\partial F}{\partial x_{ij}}, 0, \dots, 0\right)$, где i и j выбираются случайно. Метод сходится медленнее, но не требует больших вычислительных затрат.

Критика MDS

Малые значения d_{ij} не всегда должны влечь за собой малые значения δ_{ij} . Как исправить этот недостаток?



В методе Isomap минимизируется суммарное расхождение между расстояниями между объектами в пространстве новых признаков и расстояниями по поверхности (по геодезическим) в исходном пространстве.



Алгоритм устроен следующим образом:

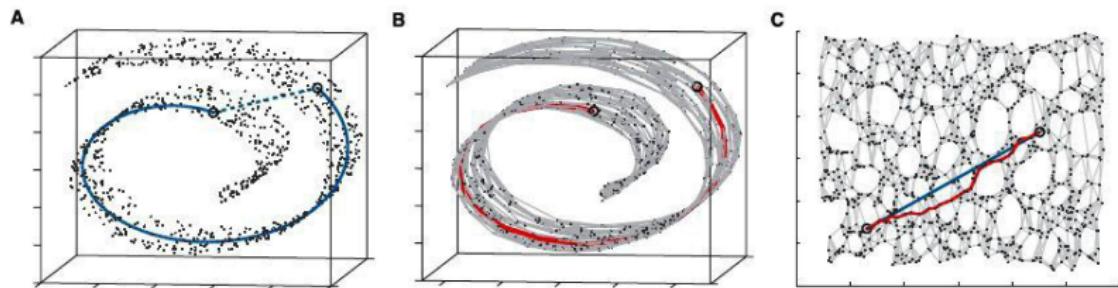
- 1) Для каждой точки находятся M её ближайших соседей, после чего строится взвешенный неориентированный граф только по расстояниям до этих ближайших соседей.
- 2) Используя алгоритм поиска кратчайших расстояний в графе, например, алгоритм Дейкстры, находятся кратчайшие расстояния (обозначим их $\|x_i - x_j\|_G$) между каждой парой вершин по графу, построенному в предыдущем пункте.
- 3) Применяем алгоритм MDS для расстояния $\|x_i - x_j\|_G$ и евклидовой нормы $\|z_i - z_j\|$.

Алгоритм определения места нового объекта в новых признаках:

- 1) Находит M ближайших соседей нового объекта из набора для обучения.
- 2) Строит граф, отображающий геодезические расстояния от данной точки до точек обучающей выборки. Для этого взвешенный неориентированный граф расстояний до ближайших соседей нового объекта дополняется графом, полученным на шаге 2) алгоритма, а затем по этому графу находятся кратчайшие расстояния от нового объекта до каждой точки обучающего набора.
- 3) Вектор, содержащий кратчайшие геодезические расстояния от нового объекта до каждой точки обучающего набора, подаётся на вход алгоритма MDS, обученного на шаге 3) алгоритма. После этого MDS возвращает точку в К-мерном пространстве.

- 1) Выбросы, лежащие между далекими частями поверхности в исходном пространстве признаков, могут сделать эти далекие части близкими в пространстве новых признаков. В таком случае нужно убрать мешающие алгоритму объекты (например, с помощью Isolation Forest).
- 2) Выбор M : слишком малое значение M ведет к плохой аппроксимации геодезического расстояния, а слишком большое может привести к “зацикливанию” через объекты из пункта 1.

Пример работы алгоритма:



t-SNE (t-distributed stochastic neighbor embedding) – один из лучших алгоритмов понижения размерности, существующих на настоящий момент, в частности, хорошо подходящий для визуализации данных.

Вначале вычислим меру “похожести” объектов x_i и x_j в исходном пространстве. Определим

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)},$$

и положим $p_{ij} = \frac{1}{2n}(p_{j|i} + p_{i|j})$, т.е. чем дальше друг от друга x_i и x_j , тем экспоненциально меньше мера их близости. Параметр σ_i настраивается аналогично bandwidth в ядерных оценках плотности.

Далее, определим объекты $\{z_i\}_{i=1}^n$ в пространстве меньшей размерности (размерность можно выбрать с помощью PCA) так, чтобы они отражали меру похожести объектов $\{x_i\}_{i=1}^n$. Обозначим

$$q_{ij} = \frac{(1 + \|z_i - z_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|z_k - z_i\|^2)^{-1}}.$$

Видим, что удаленность объектов в пространстве меньшей размерности штрафуется меньше. Расположение объектов $\{z_i\}_{i=1}^n$ определяется с помощью минимизации градиентным спуском расстояния Кульбака-Лейблера “распределения” Q от распределения P

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \ln \frac{p_{ij}}{q_{ij}}.$$

Алгоритм предложен в 2018 году с целью получить аналог t-SNE, но с более сильным математическим обоснованием. Определим для каждого объекта x_i множество k его ближайших соседей $T_i = \{t_{i1}, \dots, t_{ik}\}$.

Определим также $\rho_i = \min_{t \in T_i} d(x_i, t)$ – расстояние до ближайшего соседа для i -того объекта – и величину σ_i , заданную уравнением

$$\sum_{t \in T_i} \exp\left(-\frac{d(x_i, t) - \rho_i}{\sigma_i}\right) = \log_2 k.$$

Сначала алгоритм выполняет построение взвешенного ориентированного графа, где ребра соединяют объекты с его k ближайшими соседями.

Вес ребра $(x_i \rightarrow t_{ij})$ рассчитывается как

$$w(x_i \rightarrow t_{ij}) = \exp\left(-\frac{d(x_i, t_{ij}) - \rho_i}{\sigma_i}\right).$$

Видим, что сумма этих весов для каждого i равна $\log_2 k$.

Вполне могло оказаться так, что между 2 объектами есть 2 ребра с разными весами. Но хочется интерпретировать вес ребра как вероятность того, что 2 объекта соединены ребром.

Поэтому объединим такие ребра в одно с весом, равным вероятности существования хотя бы одного ребра:

$$w_{ij} = w(x_i \rightarrow x_j) + w(x_j \rightarrow x_i) - w(x_i \rightarrow x_j)w(x_j \rightarrow x_i).$$

Так мы получаем взвешенный неориентированный граф (по аналогии с Isomap).

Последний шаг алгоритма создает новый граф в низкоразмерном пространстве и приближает множество его ребер к исходному, минимизируя сумму расстояний Кульбака-Лейблера:

$$\sum_{i < j} \left(w_{ij} \log \frac{w_{ij}}{\tilde{w}_{ij}} + (1 - w_{ij}) \log \frac{1 - w_{ij}}{1 - \tilde{w}_{ij}} \right) \rightarrow \min_{\tilde{w}},$$

где \tilde{w}_{ij} – веса (вероятности) ребер в новом пространстве признаков меньшей размерности.

Минимизация проводится с помощью стохастического градиентного спуска. Полученные веса ребер определяют расположение объектов в новом пространстве.

Локальное линейное погружение (LLE)

Наряду с t-SNE, методы, основанные на LLE (local linear embedding), показывают наилучшие результаты при снижении размерности данных. Алгоритм построен на простой идеи – методе взвешенного среднего.

Итак, найдем для каждого объекта x_i M его ближайших соседей $x_{i(1)}, \dots, x_{i(M)}$ и найдем такие веса $\{w_{ij}\}_{j=1}^M$, что

$$L(w) = \sum_{j=1}^M (x_i - w_{ij} x_{i(j)})^2 \rightarrow \min_w$$

при условии $\sum_j w_{ij} = 1$.

Локальное линейное погружение (LLE)

Чтобы определить объекты $\{z_i\}_{i=1}^n$ в пространстве меньшей размерности, решим задачу оптимизации

$$\sum_{i=1}^n \left(z_i - \sum_{j=1}^M w_{ij} z_{i(j)} \right)^2 \rightarrow \min_z .$$

Существуют разновидности метода LLE (например, Hessian LLE и Modified LLE), которые работают лучше обычного LLE, но, в отличие от HLLE, MLLE не сильно отличается по вычислительной сложности от LLE.

Laplacian eigenmaps

Ещё одной модификацией LLE является метод Laplacian eigenmaps, где в качестве весов выбираются

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right),$$

где $j \in \{i(1), \dots, i(M)\}$, а затем объекты в пространстве меньшей размерности выбираются с помощью решения задачи оптимизации

$$\sum_{i=1}^n \sum_{j \in \{i(1), \dots, i(M)\}} w_{ij} (z_i - z_j)^2 \rightarrow \min_z .$$

Замечания о локальных методах

- 1) “Зацикливание” (short-circuiting) может произойти только на маленьких участках поверхности и не будет затрагивать всю поверхность;
- 2) Локальные методы, основанные на KNN, склонны к проклятию размерности;
- 3) Локальные методы склонны к переобучению на выбросах (когда они являются ближайшими соседями для каких-то точек поверхности).
- 4) В случае, если $\lambda_{max} \gg \lambda_{min}$, решение задачи оптимизации может быть неустойчивым.

Свойства методов

Technique	Convex	Parameters	Computational	Memory
PCA	yes	none	$O(D^3)$	$O(D^2)$
MDS	yes	none	$O(N^3)$	$O(N^2)$
Isomap	yes	K	$O(N^3)$	$O(N^2)$
MVU	yes	K	$O((NK)^3)$	$O((NK)^3)$
Kernel PCA	yes	kernel	$O(N^3)$	$O(N^2)$
Diffusion maps	yes	σ, T	$O(N^3)$	$O(N^2)$
Autoencoders	no	network shape	$O(INW)$	$O(W)$
LLE	yes	K	$O(pN^2)$	$O(pN^2)$
Laplacian eigenmaps	yes	K, σ	$O(pN^2)$	$O(pN^2)$

Обозначения: D – размерность исходного пространства, N – количество объектов, K – кол-во ближайших соседей, σ – параметр гауссова ядра, p – доля ненулевых элементов в матрице весов.

Видим, что по вычислительной сложности PCA вне конкуренции, потом идут локальные методы (LLE и Laplacian eigenmaps), а потом все остальные (t-SNE там же).

Сравнение глобальных и локальных методов

Глобальные методы стремятся сохранить общую структуру данных

- 1) в целом менее эффективны;
- 2) шумовые точки могут испортить всю картину в целом.

Локальные методы стремятся сохранить только локальные свойства поверхности, вдоль которой расположены данные

- 1) в целом более эффективны;
- 2) эффект от шумовых точек остается локальным и не распространяется на все данные.

Finita!