



Снижение размерности многомерных данных

Центр биоэлектрических интерфейсов, 23 января 2019 г.

Денис Деркач, Влад Белавин

Оглавление

Введение

Мотивация

Постановка задачи снижения размерности

Линейные методы

Метод главных компонент

Неотрицательное матричное разложение

Проблема линейных методов

Нелинейные локальные методы

Некоррелированность и независимость

Выделение независимых компонент

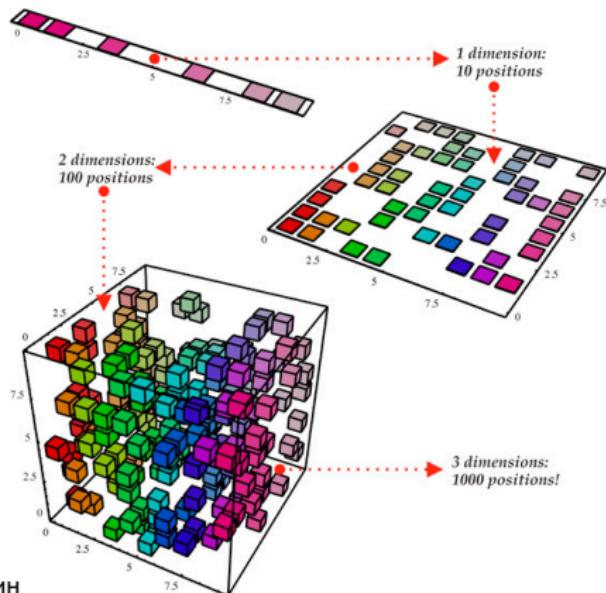
PCA vs ICA

Введение

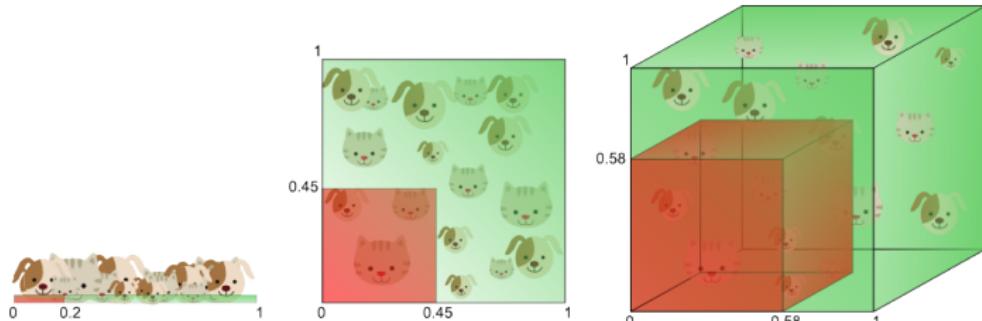
Проклятие размерности

Идея

Большая размерность требует большее число объектов для равномерного «покрытия» части пространства.



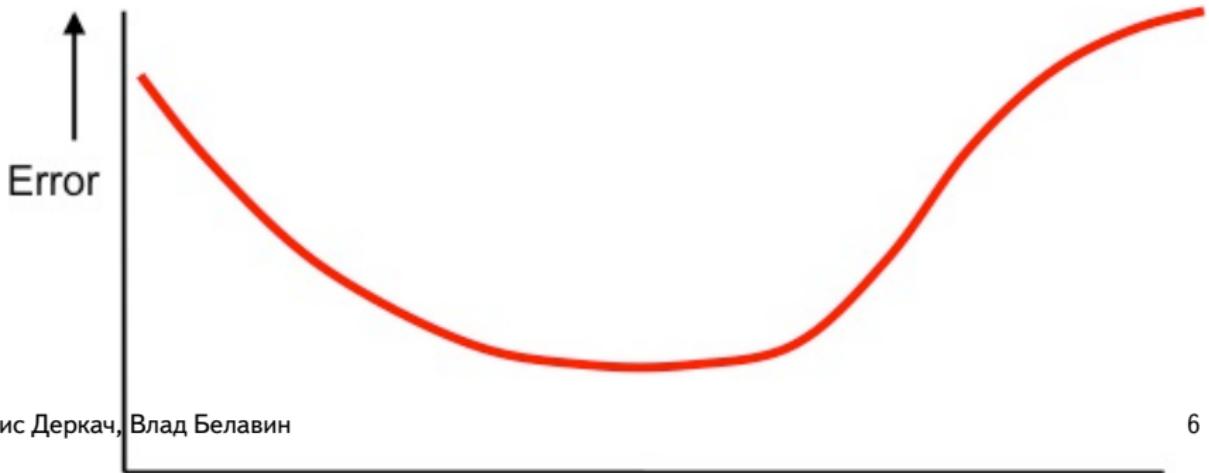
Проклятие размерности



Чтобы занять 20% объема пространства объектов, нужно захватить всё большую часть оси.

Проклятие размерности в задаче классификации

- › Малая размерность влечёт недостаточное описание для классификации.
- › Чрезмерно большая размерность влечёт разреженность пространства объектов и переобучение классификатора.



Цели снижения размерности

Проблема мультиколлинеарности

При решении задачи регрессии методом наименьших квадратов:

$$\beta = (X^T X)^{-1} X^T.$$

При $\det(X^T X) \approx 0 \Rightarrow$ неустойчивая оценка параметров моделей и их дисперсий.

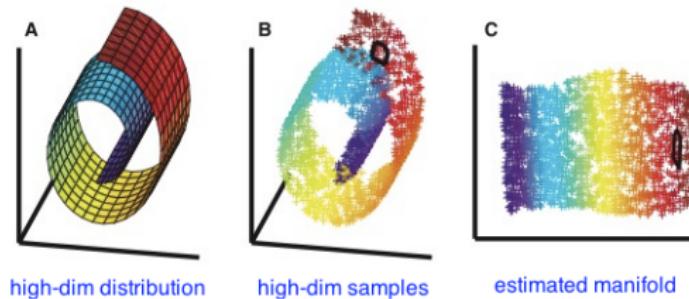
Упрощение вычислений

Меньше размерность \Rightarrow проще вычисления, требуется меньше памяти для хранения выборки.

Представление по выборке многообразия данных

Дано:

Выборка данных.



Цель:

Необходимо обнаружить некоторое многообразие, лежащее в
пространстве высокой размерности.

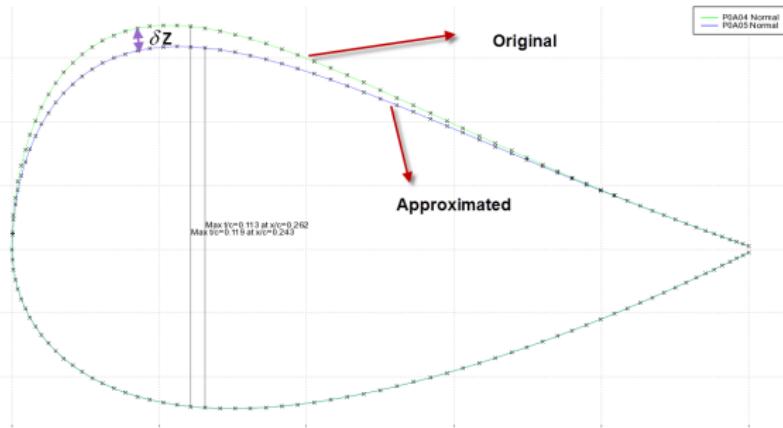
Представление по выборке многообразия данных

В пространстве всех возможных фотографий лежит пространство низкой размерности — фото лица со всех возможных сторон.

Мы можем описать изображение, задав положение камеры на единичной сфере. Это можно сделать с помощью двух углов.



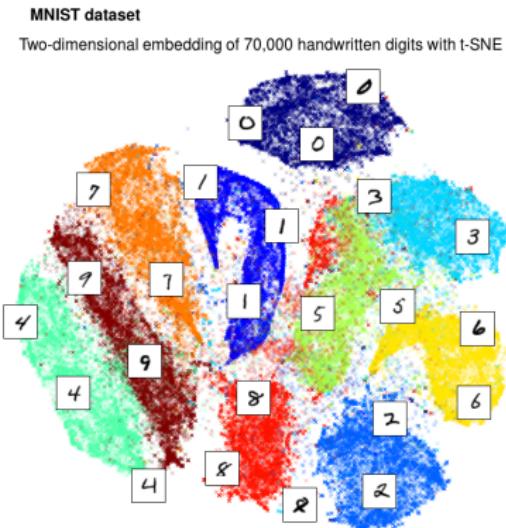
Представление по выборке многообразия данных



Множество сечений (профилей) некоего крыла, изначально описанного 59 координатами, может быть описано шестью координатами.

Визуализация

Для человека удобным представлением данных является двухмерное (трёхмерное) представление.



В данном случае размерность данных MNIST(изображения рукописных цифр 28x28) снижена до 2.

Постановка задачи снижения размерности

Дано:

Пусть объект $O \in S$ описывается вектором $X(O) \in \mathbb{R}^p$.

Всё множество объектов описано набором $\mathbb{X} = \{X(O), O \in S\}$.

Пусть описание доступно только для объектов $\{O_1, \dots, O_n\}$.

$\mathbf{X}_n = \{X_i = X(O_i), i = 1, 2, \dots, n\} = \{X_1, X_2, \dots, X_n\}$.

Задача:

Необходимо построить правило краткого описания объектов $y(X(O)) \in \mathbb{R}^q, q < p$ без «значимой» потери точности описания.

Виды задачи

- › Embedding Problem (E-problem)

Отображение должно быть определено только на обучающей выборке.

- › Extended Embedding Problem (EE-problem)

Отображение должно быть определено на любом детальном описании объекта.

- › Full Dimension Reduction problem (Full DR-problem)

Нужно также найти некоторое обратное преобразование из краткого описания в детальное.

Линейные методы

Содержание

Введение

Мотивация

Постановка задачи снижения размерности

Линейные методы

Метод главных компонент

Неотрицательное матричное разложение

Проблема линейных методов

Нелинейные локальные методы

Некоррелированность и независимость

Выделение независимых компонент

PCA vs ICA

Метод главных компонент

Доступно описание n объектов размерности p : X_1, X_2, \dots, X_n . Мы хотим построить отображение $y : \mathbb{R}^p \rightarrow \mathbb{R}^q, q < p$.

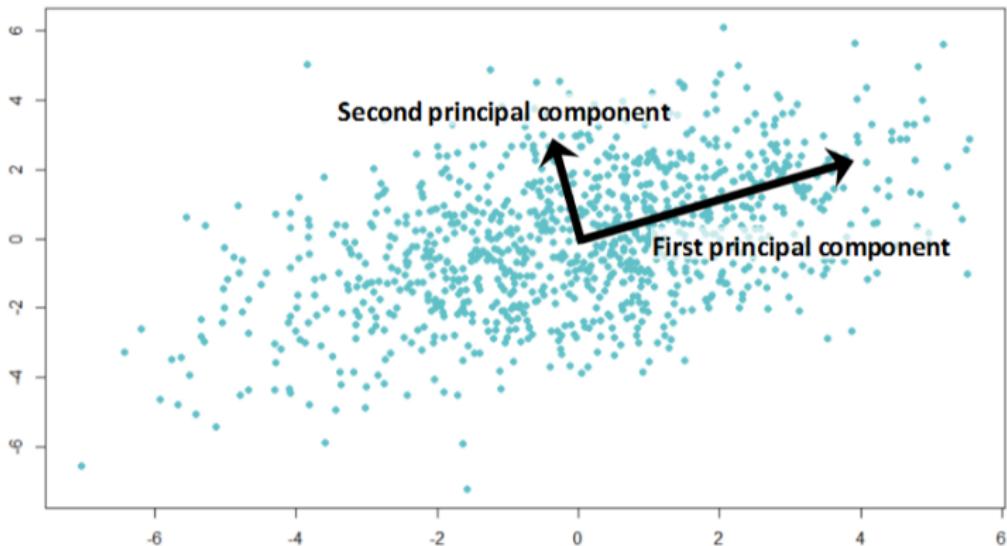
Идея: приблизим исходное пространство линейной оболочкой из q компонент p -мерного ортогонального базиса и p -мерного смещения:

$$x \approx \mu + \mathbf{V}_q y(x), \quad \mathbf{V}_q \in \mathbb{R}^{p \times q}, \quad \mathbf{V}_q^T \mathbf{V}_q = \mathbf{I}_p,$$

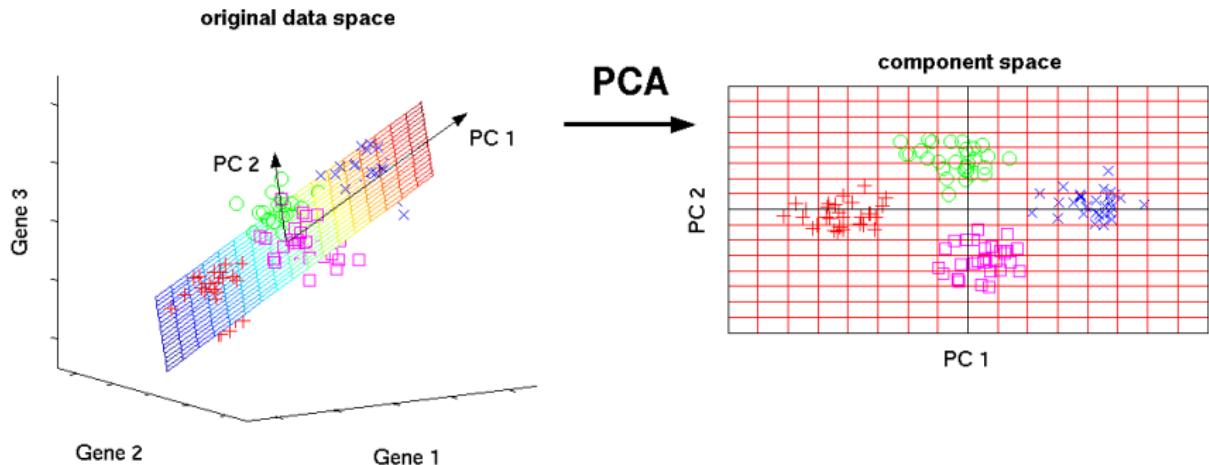
минимизируя квадрат ошибки восстановления:

$$\min_{\mu, \{y_i\}_{i=1}^n, \mathbf{V}_q} \sum_{i=1}^n \|X_i - \mu - \mathbf{V}_q y_i\|^2.$$

Метод главных компонент



Метод главных компонент



Метод главных компонент

$$\min_{\mu, \{y_i\}_{i=1}^n, \mathbf{V}_q} \sum_{i=1}^n \|X_i - \mu - \mathbf{V}_q y_i\|^2.$$

Продифференцировав, можно найти:

$$\hat{\mu} = \bar{x} = \sum_{i=1}^n \frac{X_i}{n},$$

$$\{y_i = \mathbf{V}_q^T X_i\}_{i=1}^n.$$

Тогда задача сводится к

$$\min_{\mathbf{V}_q} \sum_{i=1}^n \|(X_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (X_i - \bar{x})\|^2,$$

где $\mathbf{V}_q \mathbf{V}_q^T$ -матрица, производящая последовательное сжатие и разжатие центрированной выборки.

Метод главных компонент

Решение

$$\min_{\mathbf{V}_q} \sum_{i=1}^n \|(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})\|^2$$

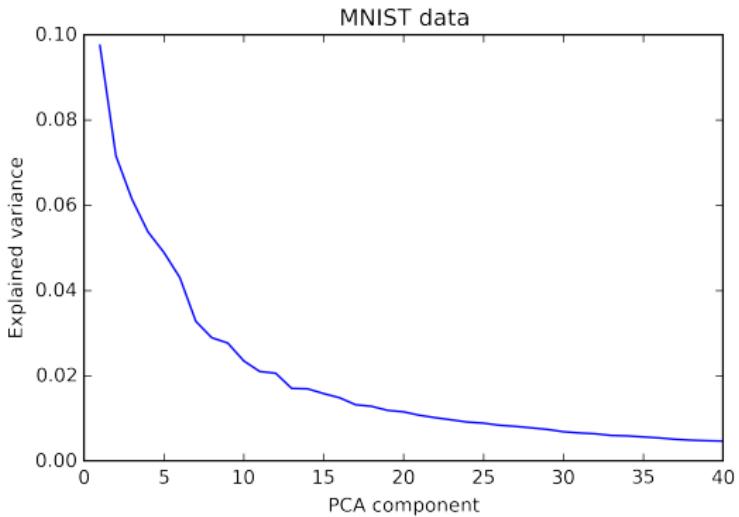
может быть найдено через SVD-разложение центрированной матрицы исходных данных, у которой в качестве строк записаны центрированные транспонированные векторы исходного пространства.

$$\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T.$$

При этом

- › \mathbf{V}_q состоит из первых q столбцов матрицы \mathbf{V} .
- › Столбцы матрицы \mathbf{V}_q называют главными компонентами матрицы \mathbf{X} .
- › Матрица \mathbf{X} есть матрица, строками которой являются векторы $X_i, i = 1, \dots, n$.

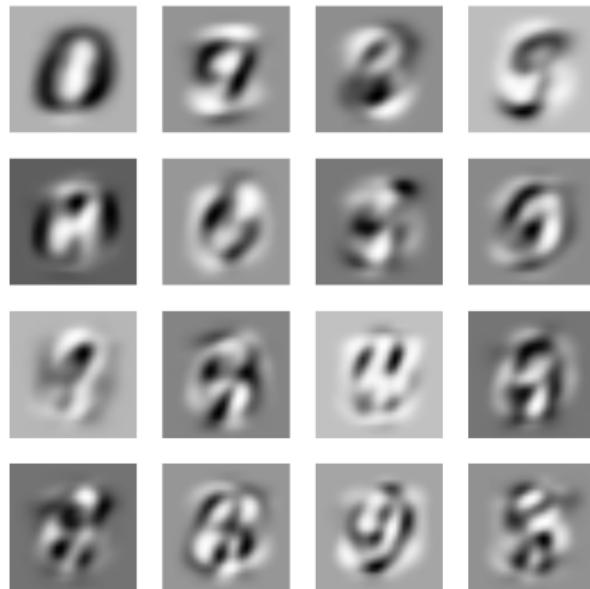
Метод главных компонент



Из графика можно видеть, что переменные «выбираются» по принципу убывания вдоль них дисперсии(«если дисперсия большая, то от этой переменной многое зависит, компонента значима для описания данных»).

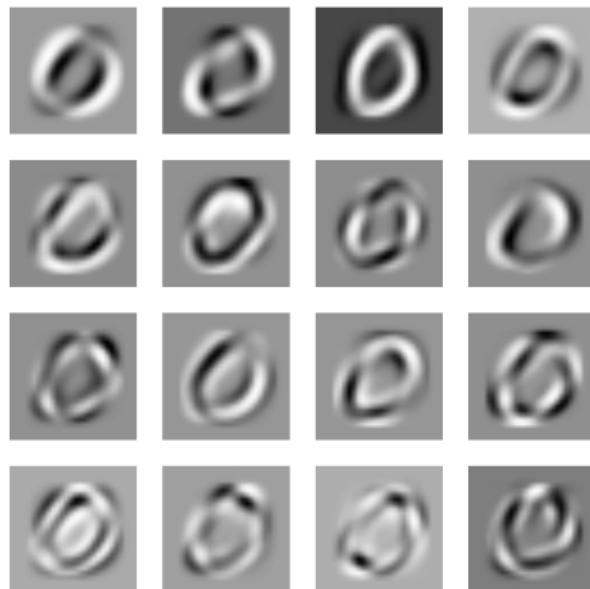
Метод главных компонент

Данные MNIST, первые 16 столбцов матрицы V_q .



Метод главных компонент

Данные MNIST, первые 16 столбцов матрицы \mathbf{V}_q для изображений нуля.



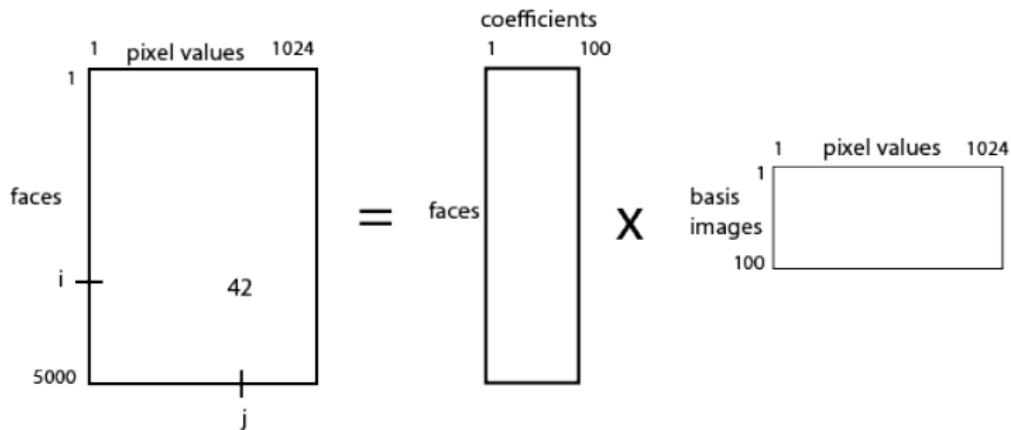
Преимущества метода

- › Легко считается.
- › Относительно интерпретируем.
- › Позволяет проводить и прямое, и обратное преобразование.

Неотрицательное матричное разложение

Неотрицательное матричное разложение состоит в представлении неотрицательных данных в виде:

$$\mathbf{X} = \mathbf{WH}, \mathbf{W} \in \mathbb{R}_+^{n \times q}, \mathbf{H} \in \mathbb{R}_+^{q \times p}.$$



Для получения такого разложения существует специальный итеративный алгоритм.

- › Надо оценить два фактора
 - › Чередуем их приближения
- › Пример алгоритма:
 - › Начнем со случайной W
 - › Оцениваем H с данной W
 - › Оцениваем W с данной H
 - › повторяем, пока не сойдется

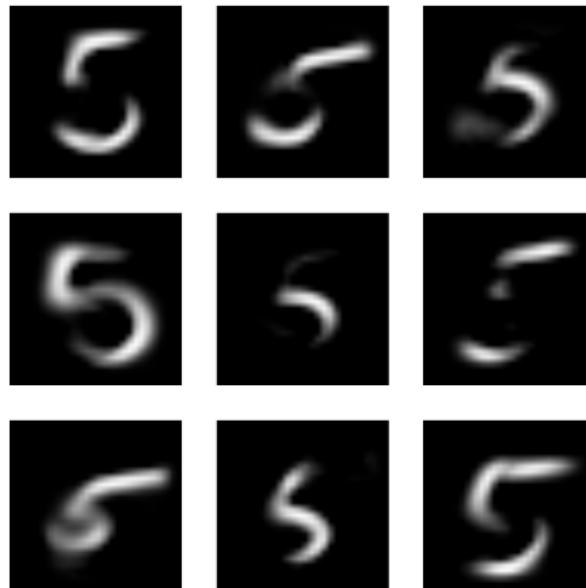
Неотрицательное матричное разложение

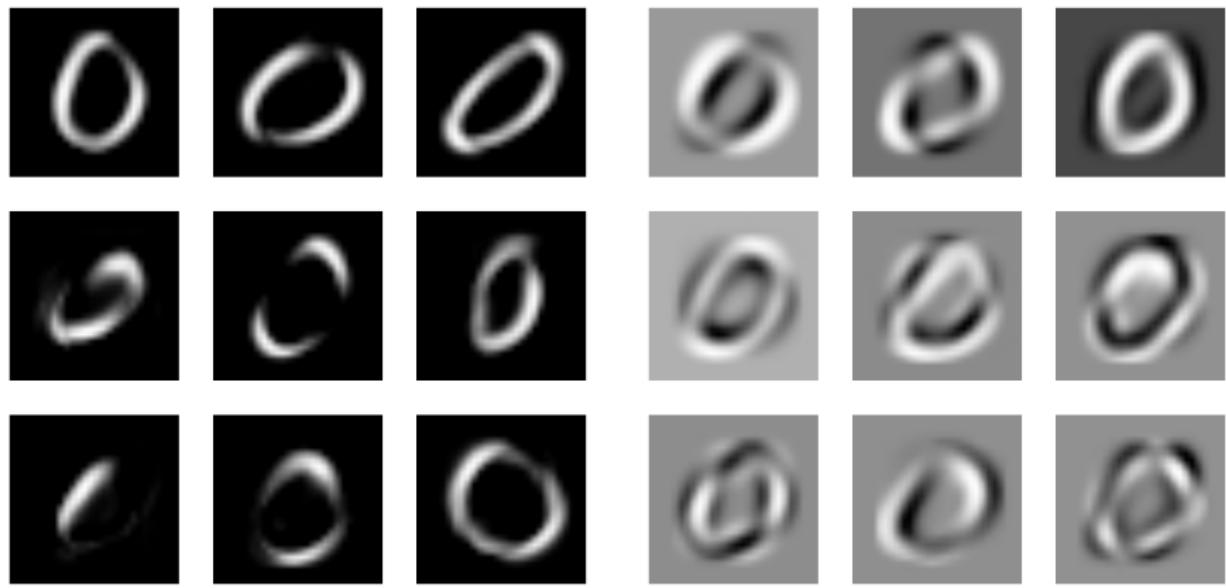
Данные MNIST, первые 9 строк матрицы H :



Неотрицательное матричное разложение

Данные MNIST, первые 9 строк матрицы H для изображений цифры 5:





Сравнение базиса, найденного неотрицательным матричным разложением и методом главных компонент для изображений нуля (MNIST).

Хорошая статья про визуализацию MNIST:

<http://colah.github.io/posts/2014-10-Visualizing-MNIST/>

Денис Деркач, Влад Белавин

Сравнение неотрицательного матричного разложения и метода главных компонент

Метод главных компонент:

- › Формулируется как наилучшее разложение данных по ортогональному базису.
- › Базис при этом может быть записан как линейная комбинация исходных данных.
- › Компоненты такого базиса может быть сложно интерпретировать.
- › Неясен смысл главных компонент для матриц количества упоминаний объекта в тексте, интенсивности пикселей.
- › Что значат отрицательные величины?

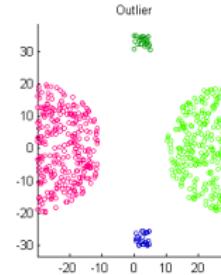
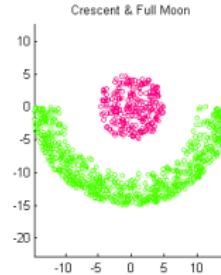
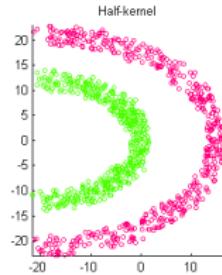
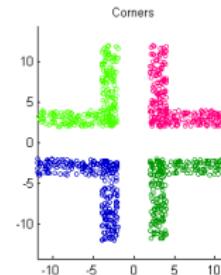
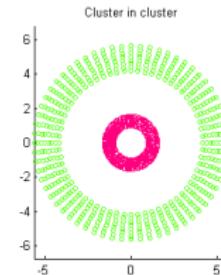
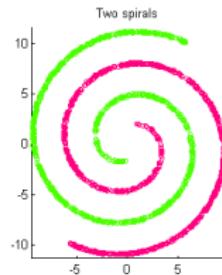
Сравнение неотрицательного матричного разложения и метода главных компонент

Неотрицательное матричное разложение:

- › Пытается найти некоторый набор базовых явлений в данных и восстановить каждый объект как линейную комбинацию этого базисного представления.
- › При этом каждое явление может входить лишь с неотрицательным коэффициентом.
- › Такое представление позволяет увеличить интерпретируемость.

Проблема линейных методов

Линейные методы позволяют находить лишь линейные многообразия данных, бессильны в следующих случаях:



Нелинейные локальные методы

Содержание

Введение

Мотивация

Постановка задачи снижения размерности

Линейные методы

Метод главных компонент

Неотрицательное матричное разложение

Проблема линейных методов

Нелинейные локальные методы

Некоррелированность и независимость

Выделение независимых компонент

PCA vs ICA

Нелинейные локальные методы

- › Locally Linear Embedding (LLE)
- › Laplacian Eigenmaps (LE)
- › Hessian Eigenmaps (HE)
- › ISOmetric MAPping (ISOMAP)
- › Kernel PCA
- › Riemannian Manifold Learning (RML)
- › Local Tangent Space Alignment (LTSA)

Нелинейные методы

Дадим представление о нелинейных методах. Большинство из них основываются на Kernel-PCA. Пусть X_1, \dots, X_N - центрированные данные:

$$\sum_{i=1}^N X_i = 0.$$

Обычный PCA состоял в диагонализации ковариационной матрицы:

$$C = \frac{1}{N} \sum_{i=1}^N X_i X_i^T.$$

Выберем функцию $\Phi : \mathbb{R}^p \rightarrow \mathbb{R}^N$. Обозначим

$$K = k(x, y) = \Phi(x)^T \Phi(y).$$

В Kernel-PCA необходимо диагонализовать матрицу K : $K\vec{a} = N\lambda\vec{a}$.
Kernel-PCA вычисляет проекцию $\Phi(x)$ на k -ую главную компоненту:

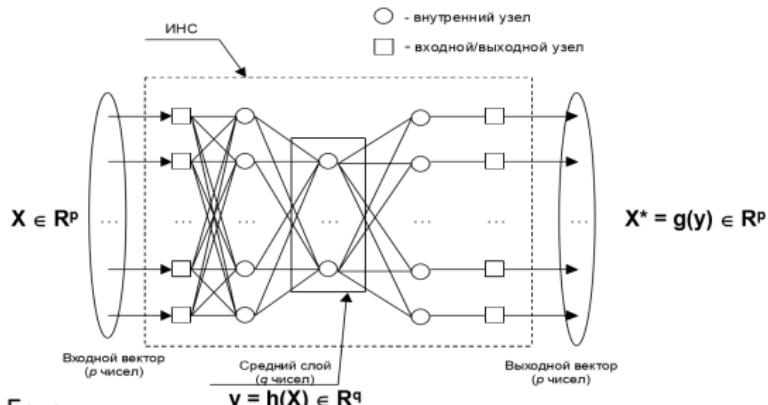
$$\mathbf{V}_q^{kT} \Phi(x) = \left(\sum_{i=1}^N a_i^k \Phi(x_i) \right)^T \Phi(x).$$

Многие методы основаны на данном подходе и различаются устройством ядра K .

Нейронная сеть как нелинейный метод

Идея состоит в том, чтобы сначала обучить нейронную сеть, а затем использовать промежуточный её слой для вывода сжатых данных, с которыми можно оперировать.

Чтобы получить из них данные первоначальной размерности предлагается передать вектор на вход промежуточному слою сети и получить выходной вектор первоначальной размерности.



Некоррелированность и независимость

Статистическая независимость

- › Мы определили статистическую независимость как

$$P(x, y) = P(x)P(y)$$

- › Что подразумевает

$$E\{f(x)g(y)\} = E\{f(x)\}E\{g(y)\}$$

$$\forall f, g$$

- › По сути независимость означает, что мы ничего не можем сказать про X, если наблюдаем Y

Некоррелированность и независимость

- › Некоррелированность не всегда влечет независимость!
 - › Некоррелированность: $E\{xy\} = E\{x\}E\{y\}$
 - › Независимость: $E\{f(x)g(y)\} = E\{f(x)\}E\{g(y)\}$
- › Но независимость всегда влечет некоррелированность
 - › Когда $f(x) = x$ и $g(y) = y$
 - › Некоррелированность - подмножество независимости

Некоррелированность и независимость

› Пример с дискретными величинами:

› Коррелируют ли они?

| | $x = -1$ | $x = 0$ | $x = 1$ |
|----------|----------|---------|---------|
| $y = -1$ | 0 | $1/4$ | 0 |
| $y = 0$ | $1/4$ | 0 | $1/4$ |
| $y = 1$ | 0 | $1/4$ | 0 |

› x, y не коррелируют:

$$E\{xy\} = E\{x\}E\{y\} = 0$$

Некоррелированность и независимость

- › Пример с дискретными величинами:
 - › Являются ли они зависимыми?

| | $x = -1$ | $x = 0$ | $x = 1$ |
|----------|----------|---------|---------|
| $y = -1$ | 0 | $1/4$ | 0 |
| $y = 0$ | $1/4$ | 0 | $1/4$ |
| $y = 1$ | 0 | $1/4$ | 0 |

Некоррелированность и независимость

- › Пример с дискретными величинами:
 - › Являются ли они зависимыми?

| | $x = -1$ | $x = 0$ | $x = 1$ |
|----------|----------|---------|---------|
| $y = -1$ | 0 | 1/4 | 0 |
| $y = 0$ | 1/4 | 0 | 1/4 |
| $y = 1$ | 0 | 1/4 | 0 |

- › Да, x и y являются зависимыми:

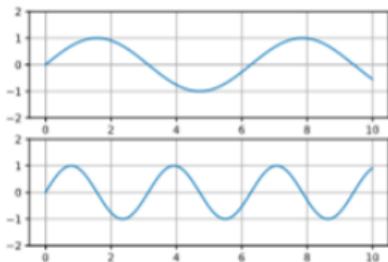
$$E\{x^2y^2\} = 0 \neq E\{x^2\}E\{y^2\} = 1/4$$

Пример с сигналами

- › Являются ли x, y некоррелированными?

$$x = \sin(t)$$

$$y = \sin(2t)$$



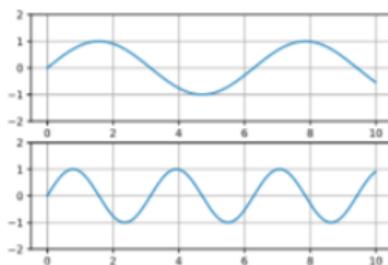
Пример с сигналами

- › Являются ли x, y некоррелированными?

$$x = \sin(t)$$

$$y = \sin(2t)$$

- › Да: $E\{xy\} = 0$
 - › Но можно предсказать один, взглянув на другой
 - › Значит, они зависимы



Выделение
независимых
компонент

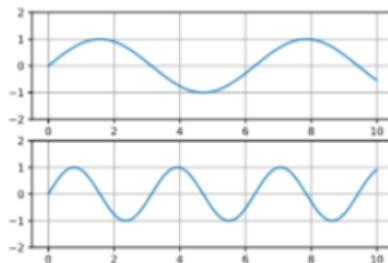
Слепое разделение сигналов (Blind Source Separation)

- › Являются ли x, y некоррелированными?

$$x = \sin(t)$$

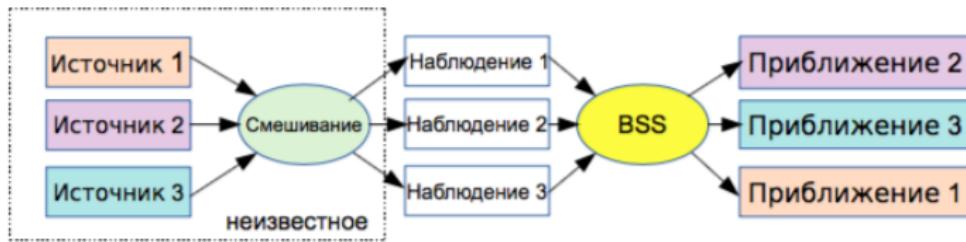
$$y = \sin(2t)$$

- › Да: $E\{xy\} = 0$
 - › Но можно предсказать один, взглянув на другой
 - › Значит, они зависимы



Формальная постановка задачи

- › BSS - это метод приближения исходных сигналов по наблюдаемым, которые могут содержать смесь исходных сигналов и шум



Как же получить независимость?

- › Пример коктейльной вечеринки

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t)$$

$$x_3(t) = a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t)$$

- › x - наблюдаемый сигнал, а s - исходный сигнал.
- › предполагаем, что s_1, s_2, s_3 попарно независимы
- › Метод должен оценить независимые компоненты $s(t)$ по $x(t)$

$$x(t) = A \cdot s(t)$$

Как же получить независимость?

- › Существует несколько способов
 - › Семейство алгоритмов ICA (independent component analysis)
- › Формальное определение:

$$y = W \cdot x,$$

где x - входной сигнал, W - обратная к матрице смешивания, y - оценка независимых компонент

$$P(y_i, y_j) = P(y_i)P(y_j) \forall i, j$$

Подход 1

- › Нелинейная декорреляция (предполагаем, что матожидание входа нулевое)
 - › Цель: $E\{f(y_i)g(y_j)\} = 0$ для фиксированных f, g
- › Алгоритм Cichocki-Unbehauen
 - › Прекратить, когда достигнута независимость

do

$$\Delta W \propto (D - f(y_i) \cdot g(y_i^T)) \cdot W$$

$$W = W + \mu \Delta W$$

repeat

$$D = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix}$$

$f(x), g(x)$ могут быть $\tanh(x), x^3, \dots$

Подход 2

- › "Диагонализация" высоких порядков
 - › В PCA мы диагонализуем ковариационную матрицу ($N \times N$ - двумерный объект)

$$Cov(y)_{i,j} = E\{y_i y_j\} = \kappa_2(y_i, y_j)$$

- › В ICA мы диагонализуем "квадриковариационный" тензор (четырехмерный объект)

$$\begin{aligned} Q(y)_{i,j,k,l} &= \kappa(y_i, y_j, y_k, y_l) = E\{y_i y_j y_k y_l\} - \\ &E\{y_i y_j\} E\{y_k y_l\} - E\{y_i y_k\} E\{y_j y_l\} - E\{y_i y_l\} E\{y_j y_k\} \end{aligned}$$

Подход 2

- › Идейно мы производим сингулярное разложение тензора
- › Алгоритм Комона (Comon P., 1994)
 - › Делаем РСА (декорреляция)
 - › Находим унитарное преобразование, минимизирующее кросс-кумулянты четвертого порядка

Подход 3

- › Подход на основе теории информации
 - › Минимизируем взаимную информацию:

$$I(\mathbf{y}) = \sum H(y_k) - H(\mathbf{y})$$

$$H(y) = - \int p_y(\eta) \log p_y(\eta) d\eta$$

- › Что подразумевает минимизацию:

$$D(\mathbf{y}) = - \int P(\mathbf{y}) \log \frac{P(\mathbf{y})}{\prod P(y_k)}$$

Подход 4

- › Негауссовость - мера независимости
- › Из ЦПТ следует, что гауссовость $x(t)$ должна быть больше гауссности $s(t)$
- › Хотим максимизировать негауссовость
- › Негэнтропия определяется как

$$J(\mathbf{y}) = H(\mathbf{y}_{Gauss}) - H(\mathbf{y}),$$

- › \mathbf{y}_{Gauss} - вектор с гауссовым распределением с $\mu = E(y)$ и $\sigma = \sqrt{E((y - \mu)^2)}$
- › Если y - распределена по гауссу, то $J(y) = 0$

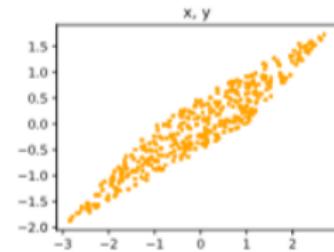
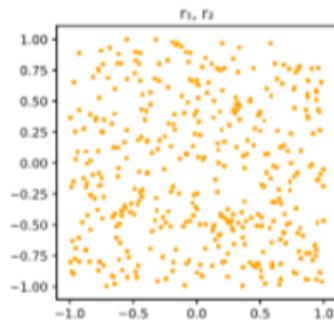
Прочие подходы

- › Метод максимального правдоподобия
- › FastICA
 - › Быстрый алгоритм с вычислениями фиксированной точности
- › Нейронные сети
 - › Напрямую оптимизируем KL-дивергенцию/взаимную информацию

PCA vs ICA

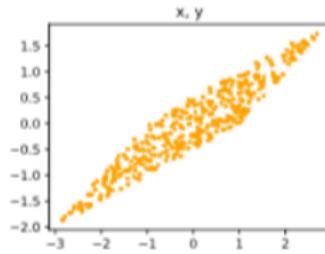
Исходные данные

- › Возьмем две равномерно распределенные случайные величины и перемешаем их
 $r_1, r_2 \sim U(-1, 1)$
 $x = 2r_1 + r_2$
 $y = r_1 + r_2$
- › Это создаст зависимые x и y
 - › Это видно на графике как поворот и растягивание данных



Применение PCA

- › PCA: декорреляция
 - › PCA основывается на направлении максимального разброса
- › Полученная проекция не привела к независимости в данных

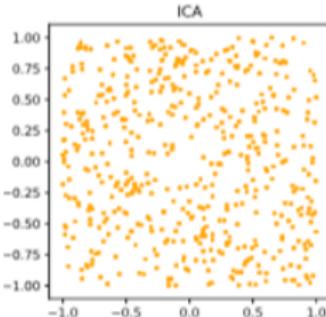
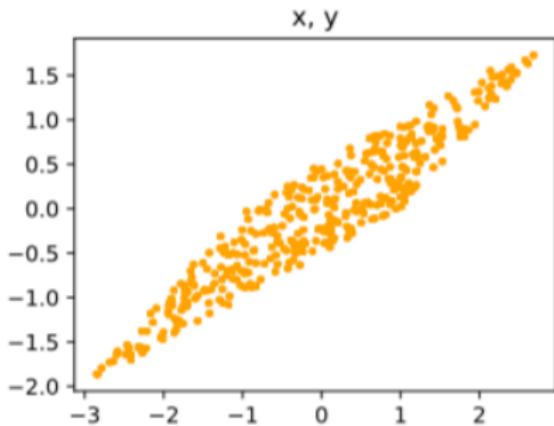


↓ PCA



Применение ICA

- › Результат ICA независим
 - › Мы получили исходные СВ, которые подавали на вход



Проблемы ICA

- › Большинство численных оценок (эстиматоры) приблизительны
 - › Результат работы не обязательно верен (находим не всегда то что искали)
 - › Может меняться между запусками и реализациями алгоритма
- › В данных может не быть независимости
 - › ICA возвращает максимально независимую проекцию, но не обязательно независимую
 - › В результате может получиться не то, что ожидали

Ограничения ICA

- › Только линейные связи
- › Инвариантность к перестановкам на выходе

$$P(y_1, y_2, y_3) = P(y_1)P(y_2)P(y_3) = P(y_2)P(y_1)P(y_3) = \dots$$

- › Порядок результата не гарантируется и может различаться между запусками
- › Не упорядочивает компоненты
 - › PCA упорядочивает результаты по их значимости в дисперсии
 - › ICA никак не упорядочивает
 - › Как следствие мы не можем понизить размерность

Совмещение PCA и ICA

- › Если нужно понизить размерность, перед ICA запускаем PCA
 - › 1) Используем PCA для понижения размерности
 - › 2) Используем ICA, чтобы установить независимость
 - › Применяем ICA к выходу PCA

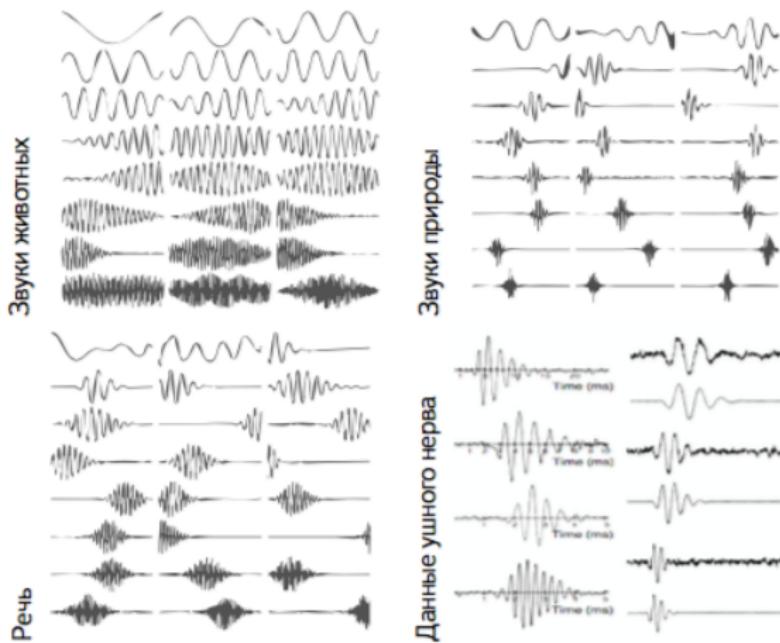
Выделение признаков

- › Собираем много естественных звуков
 - › Звуки природы, птиц, ходьба по листьям и т.п.
- › Помещаем маленькие окна в большую матрицу
 - › Применяем PCA и ICA

$$Z = W \cdot \begin{bmatrix} x(t) & x(t+1) & \dots \\ \vdots & \vdots & \\ x(t+N) & x(t+1+N) & \dots \end{bmatrix}$$

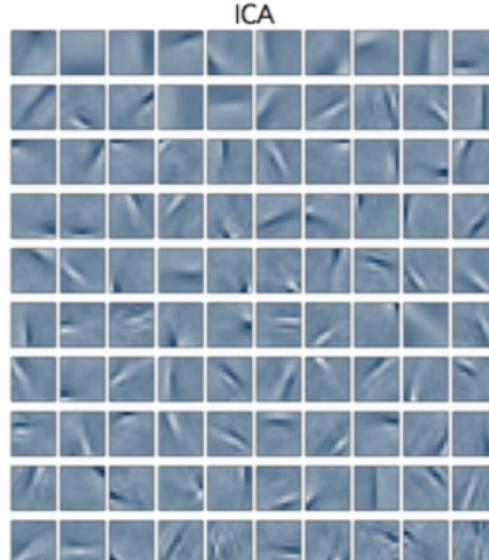
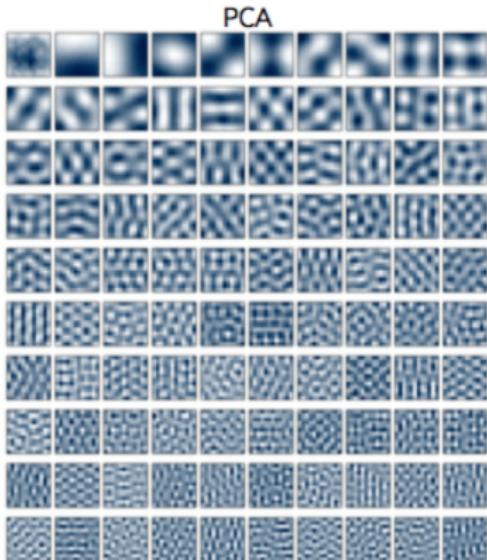
- › Мы знаем, что результат PCA - это синусоиды

Пример: признаки из звуков



Пример: признаки из изображений

- ICA компоненты похожи на первичные зрительные зоны



Пример: признаки из лиц

Eigenfaces



ICA-faces



ICA vs PCA

- › PCA предполагает "нормальный" мир
 - › Для многомерного гауссова вектора он возвращает независимые компоненты
 - › Гауссовые моменты второго порядка уже равны нулю
 - › ICA ослабляет предположение о гауссовости и предполагает более сложные распределения
 - › Это больше похоже на реальный мир