ТЕХНОСФЕРА

Лекция 4
Методы снижения размерности
Линейные методы выделения признаков

Владимир Гулин

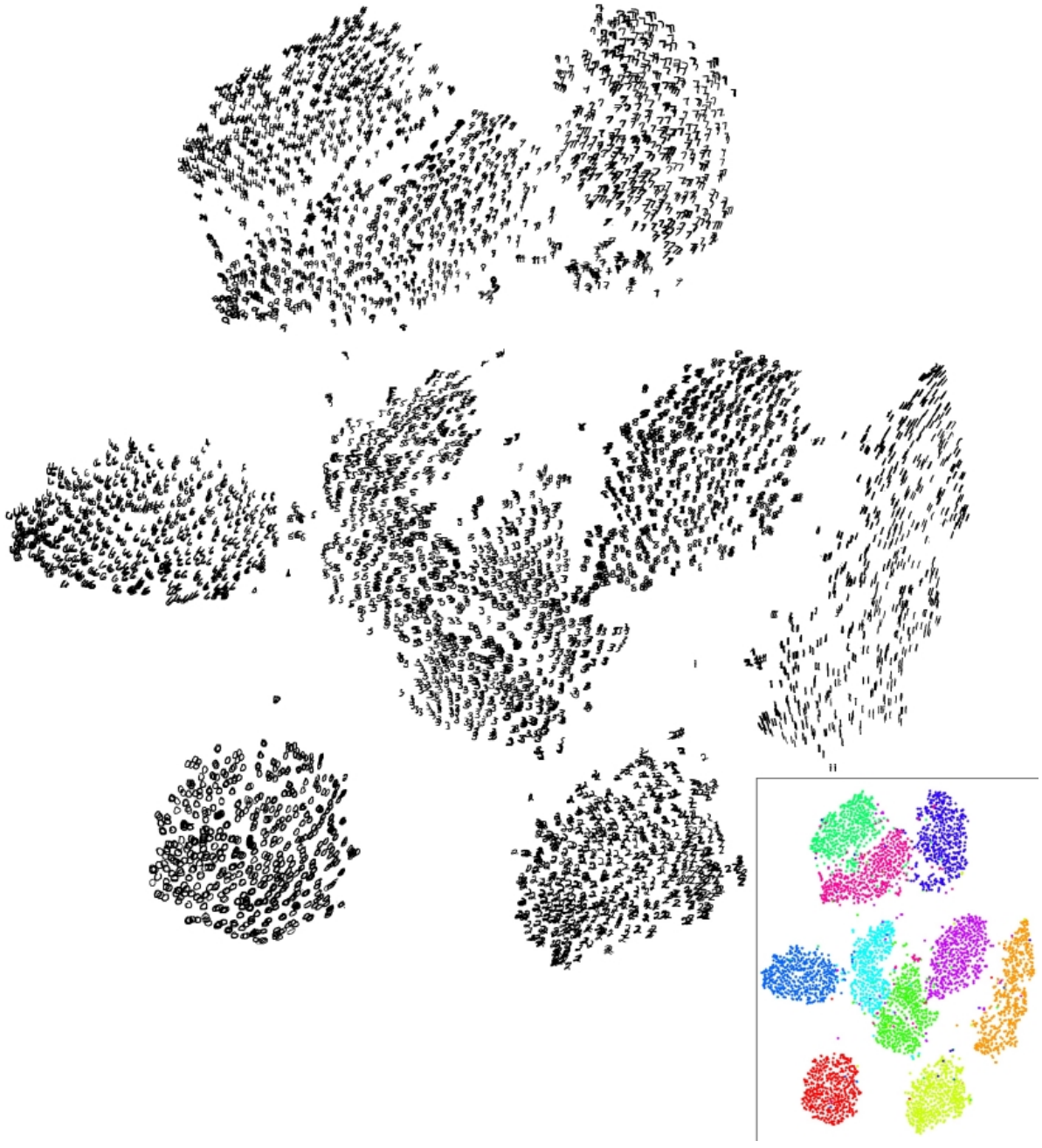
https://goo.gl/Df4u7W

20 февраля 2016 г.

План лекции

MNIST

Latent data structure



Задача выделения/синтеза признаков

Feature Extraction

Дано. N обучающих D-мерных объектов $\mathbf{x}_i \in \mathcal{X}$, образующих тренировочный набор данных (training data set) $\mathbf{X}$.

Найти. Найти преобразование $A : \mathcal{X} \to \mathcal{P}$, $dim(\mathcal{P}) = d < D$, сохранив при этом большую часть "полезной информации" об $\mathcal{X}$.

Что мы рассмотрим:

- PCA
- ICA
- Autoencoders with bottleneck

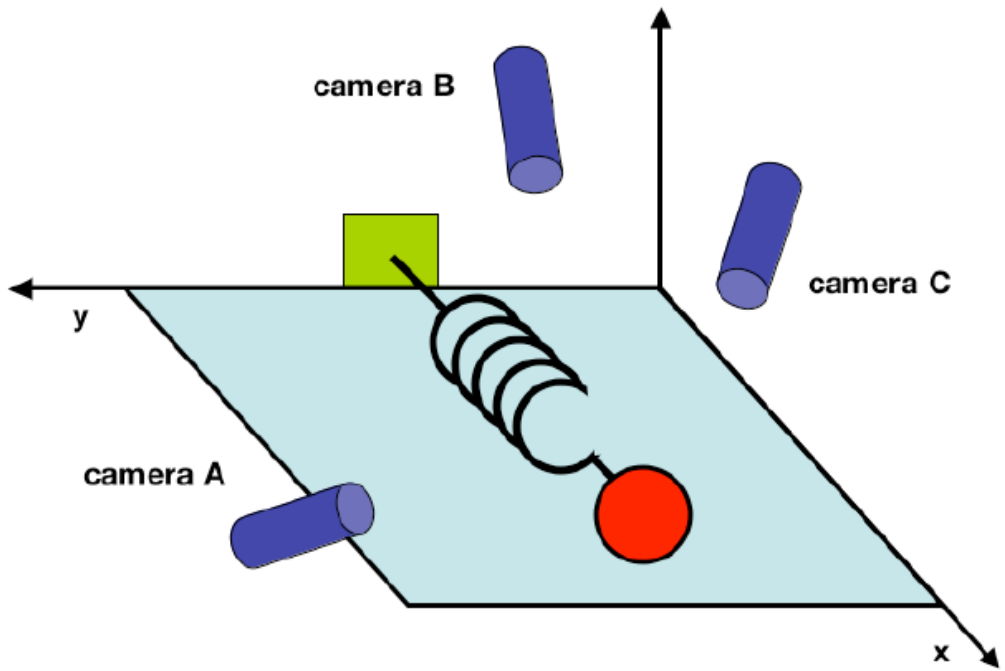
Johnson–Lindenstrauss lemma

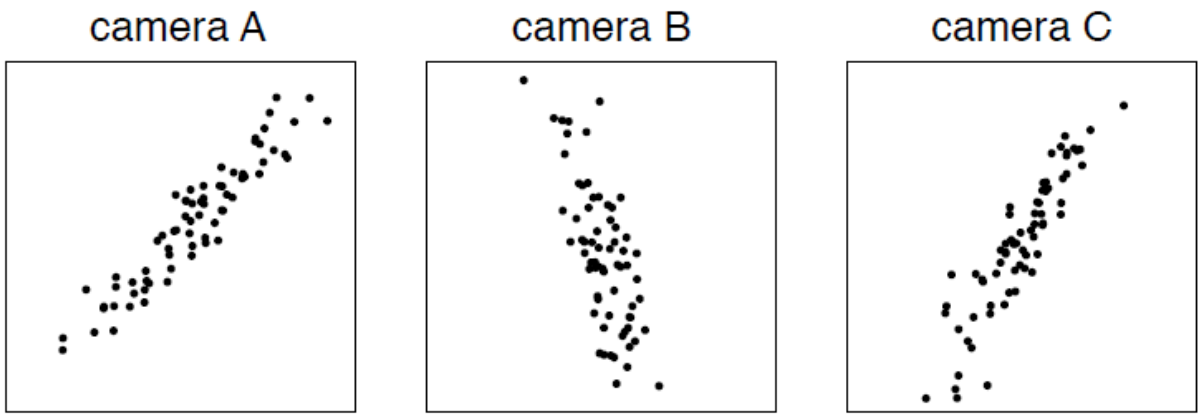
Lemma

Для любого $0 < \varepsilon < 1/2$ и $\mathbf{x}_1, \ldots, \mathbf{x}_N \in R^D$, существует линейное преобразование $f : \mathcal{R}^D \to \mathcal{R}^k$, $k = O(\varepsilon^{-2} logN)$,
такое что, для любых i, j справедливо неравентсво

$$(1 - \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2 \leq (1 + \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2$$

Toy example





Постановка задачи

Данные - серия записи с камер (N)

$$\mathbf{X}(D \times N)$$

Каждое наблюдение - вектор-столбец из $D = 6$ элементов

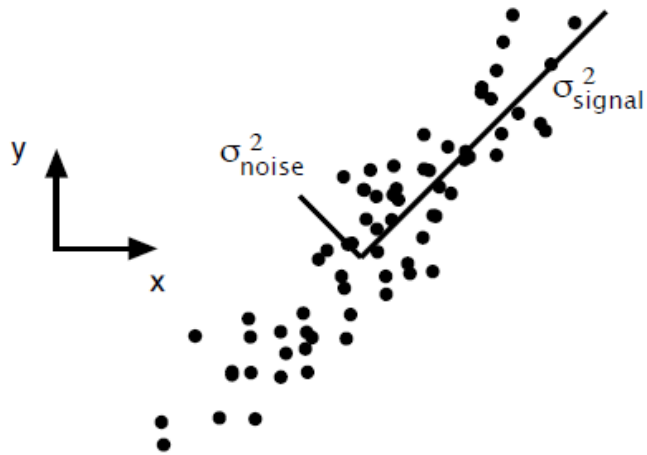
$$col = \begin{bmatrix} x_A \\ y_A \\ x_B \\ y_B \\ x_C \\ y_C \end{bmatrix}$$

Задача:

Ищем базис, который будет фильтровать шум и отражать скрытую структуру данных.

Signal-to-noise ratio

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}$$



Redundancy

- минимизируем избыточность информации в измерениях

Выборочная матрица ковариации

Для двух множеств измерений

$$A = \{a_1, \ldots, a_N\}, \quad B = \{b_1, \ldots, b_N\}$$

с дисперсиями

$$\sigma_A^2 = \frac{1}{N}\sum_{i=1}^{N} a_i^2, \quad \sigma_B^2 = \frac{1}{N}\sum_{i=1}^{N} b_i^2$$

Ковариация

$$cov(A, B) = \frac{1}{N}\sum_{i=1}^{N} a_i b_i$$

- $cov(A, B) = 0$ тогда и только тогда, когда A и B некоррелированы
- $cov(A, B) = \sigma_A^2$, если $A = B$

Выборочная матрица ковариации

Матрица наблюдений $(D \times N)$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ . \\ . \\ . \\ \mathbf{x}_D \end{bmatrix}$$

Матрица ковариации

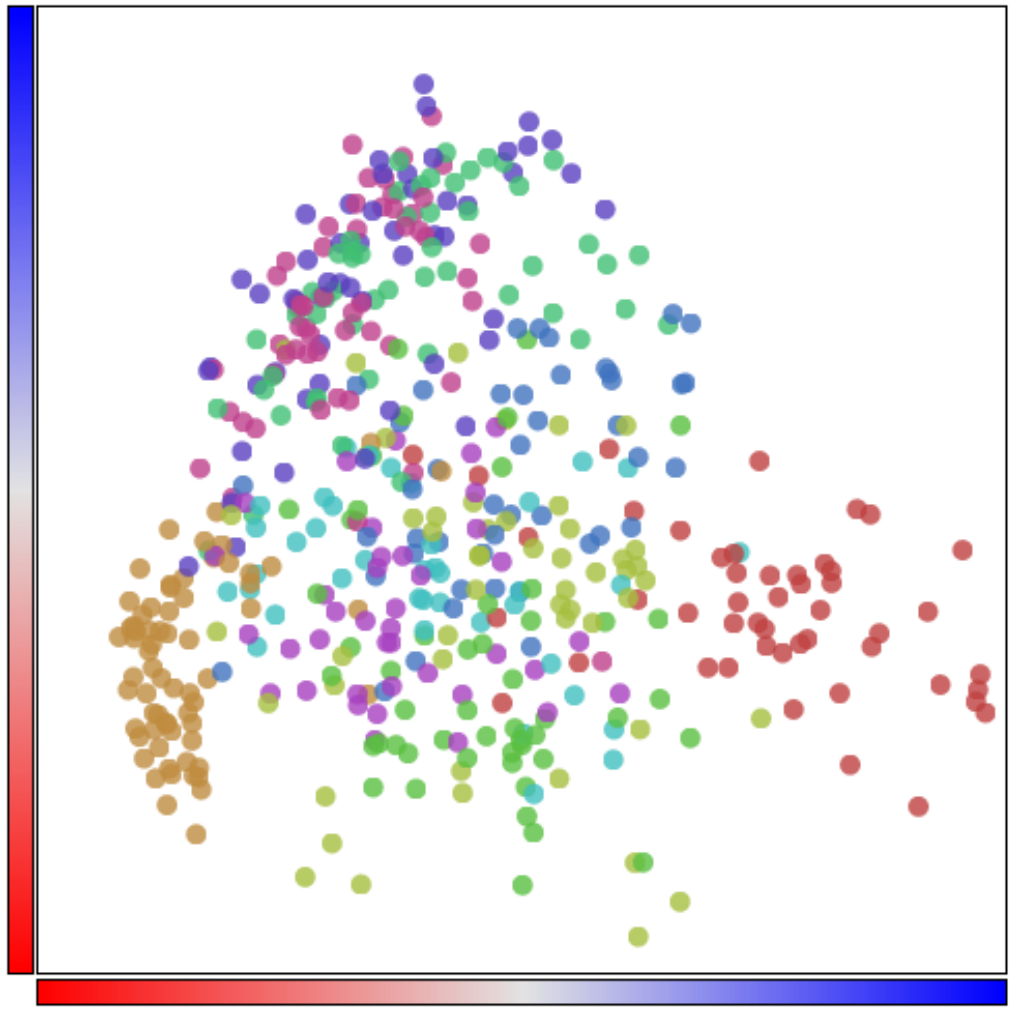
$$\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}\mathbf{X}^T, \quad (D \times D)$$

Свойства

- $\mathbf{\Sigma}$ - квадратная симметричная матрица
- Диагональные элементы $\mathbf{\Sigma}$ это дисперсии соотвествующих измерений
- Внедиагональные элементы $\mathbf{\Sigma}$ - попарные ковариации измерений

Principal Component Analysis

MNIST

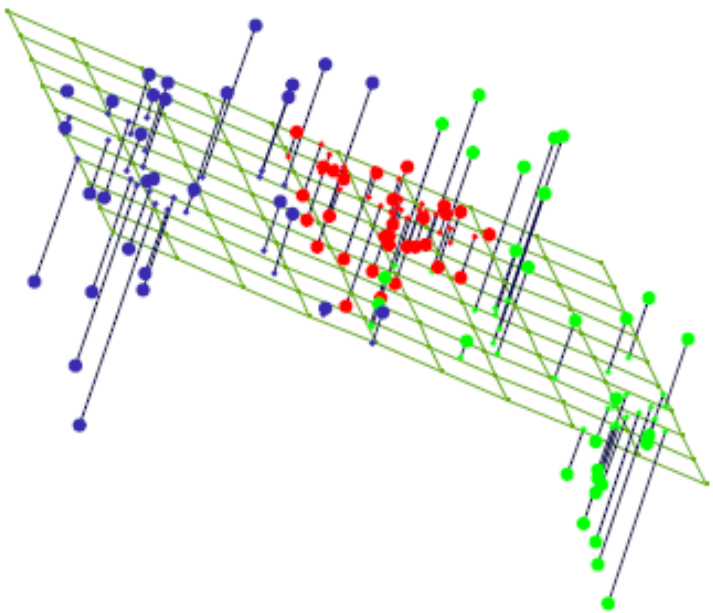


Principal Component Analysis

PCA (Principal Component Analysis) - анализ главных компонент. В теории информации известен также как преобразование Карунена-Лоева.

Суть метода:

Ищем гиперплоскость заданной размерности, такую что ошибка проектирования выборки на данную гиперплоскость была бы минимальной.



Principal Component Analysis

Будем искать преобразование в семействе линейных функций:

$$\mathbf{x} = \mathbf{A}\mathbf{p} + \mathbf{b}, \quad \text{где}$$

- $\mathbf{x} \in \mathcal{R}^D$ - представление объекта в исходном пространстве,
- $\mathbf{p} \in \mathcal{R}^d$ - новые координаты объекта
- $\mathbf{b} \in \mathcal{R}^D$, $\mathbf{A} \in \mathcal{R}^{D \times d}$

$$\mathbf{x}_j = \sum_{i=1}^{D} (\mathbf{x}_j^T \mathbf{a}_i)\mathbf{a}_i \quad \text{- исходные точки}$$

$$\tilde{\mathbf{x}}_j = \sum_{i=1}^{d} p_{j,i}\mathbf{a}_i + \sum_{i=d+1}^{D} b_i \mathbf{a}_i \quad \text{- проекции}$$

Тогда критерий выбора гиперплоскости имеет вид:

$$J = \frac{1}{N} \sum_{j=1}^{N} \|\mathbf{x}_j - \tilde{\mathbf{x}}_j\|^2 \to \min_{\mathbf{a},\mathbf{p},\mathbf{b}}$$

Principal Component Analysis

$$J = \frac{1}{N}\sum_{j=1}^{N} \|\mathbf{x}_j - \tilde{\mathbf{x}}_j\|^2 \to \min_{\mathbf{a},\mathbf{p},\mathbf{b}}$$

Несложно показать, что решение будет иметь вид:

$$p_{j,i} = \mathbf{x}_j^T \mathbf{a}_i$$

$$b_i = \bar{\mathbf{x}}^T \mathbf{a}_i$$

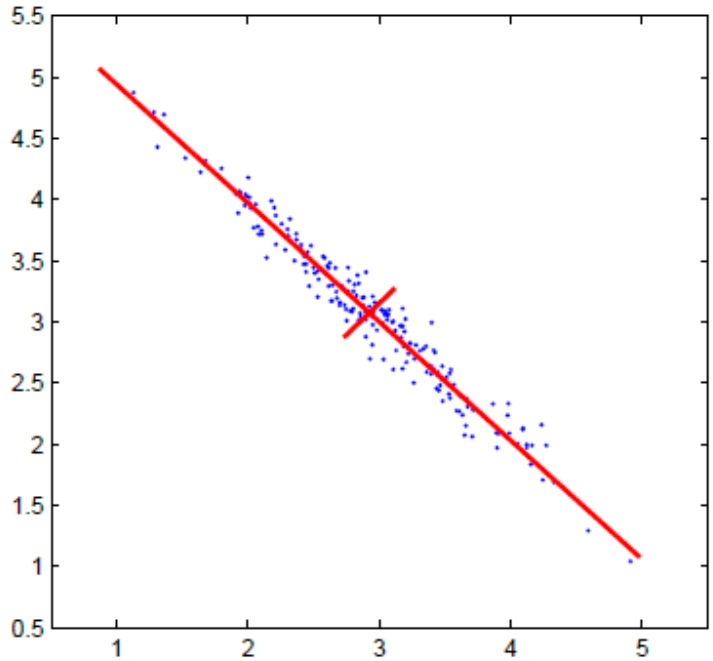
где

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j$$

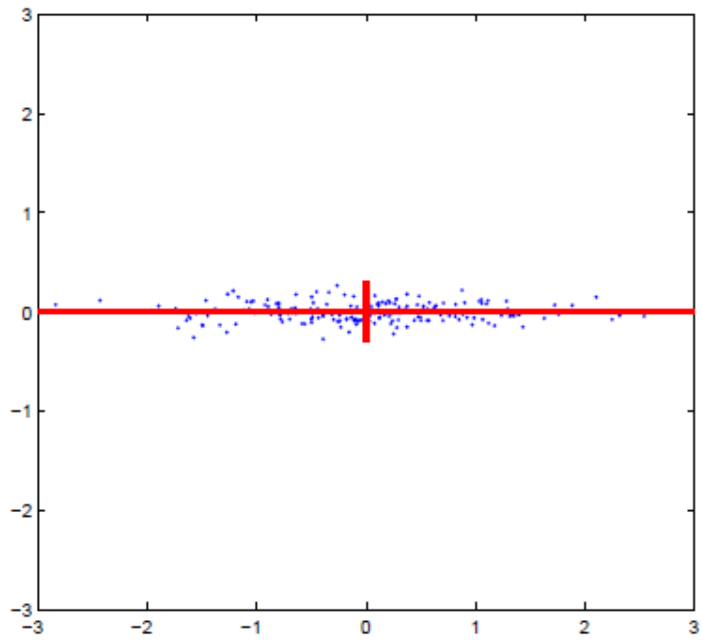
$$\mathbf{R} = cov(\mathbf{X}) = \frac{1}{N}\sum_{j=1}^{N} (\mathbf{x}_j - \bar{\mathbf{x}})^T (\mathbf{x}_j - \bar{\mathbf{x}})$$

$\mathbf{a}_i$, $i = 1, \ldots, d$ - базис из собственных векторов ковариационной матрицы $\mathbf{R}$, отвечающих d наибольших собственным значениям $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$

Иллюстрация PCA



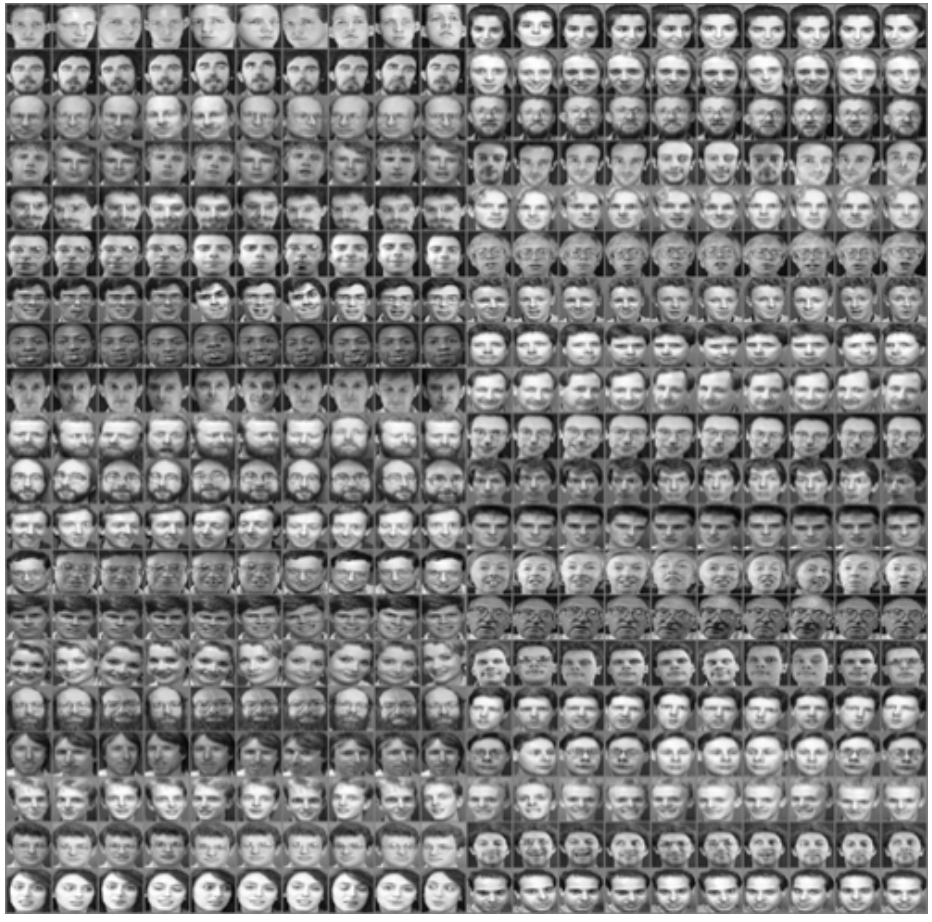
(a) Исходное пространство



(b) Итоговое пространство

- Сдвигаем начало координат в центр выборки
- Поворачиваем оси, чтобы признаки не коррелировали
- Избавляемся от координат с малой дисперсией

Eigenfaces



Eigenfaces

- Eigenfaces = Главные компоненты на датасете из лиц

Связь PCA & SVD

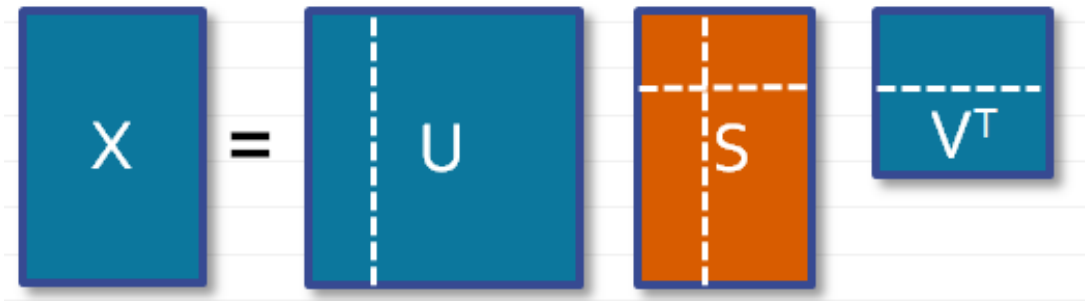
$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

где
$\mathbf{U}(N \times N)$ - ортогональная матрица левых собственных векторов (собственные вектора матрицы $\mathbf{X}\mathbf{X}^T$)
$\mathbf{V}(D \times D)$ - ортогональная матрица правых собственных векторов (собственные вектора матрицы $\mathbf{X}^T\mathbf{X}$)
$\mathbf{\Sigma}(N \times D)$ - диагональная матрица с сингулярными числами на главной диагонали

Матрица главных компонет может быть вычислена:

$$\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$$

Сжатие данных при помощи PCA



- Вместо исходной матрицы размера $N \times D$ можем хранить две матрицы $d \times N$ и $D \times d$ + вектор из d сингулярных чисел $= d(N + D + 1)$

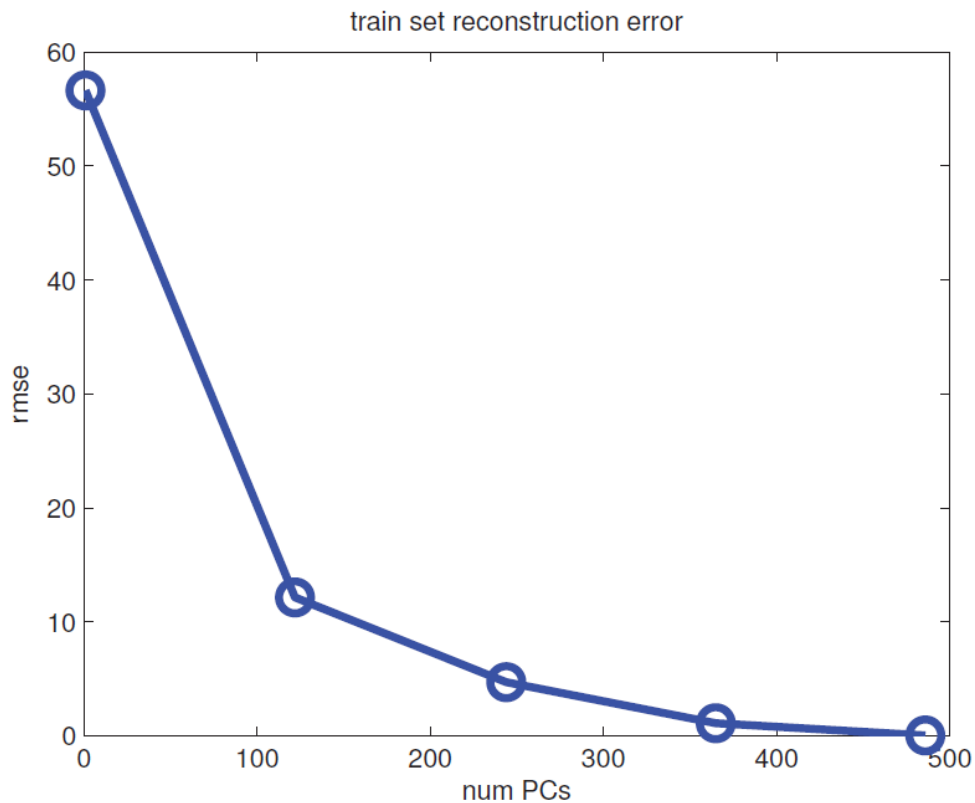
Вопрос:

- Каким образом выбрать d?

Выбор размерности редуцированного пространства

Критерий

$$J = \frac{1}{N} \sum_{j=1}^{N} \|\mathbf{x}_j - \tilde{\mathbf{x}}_j\|^2$$

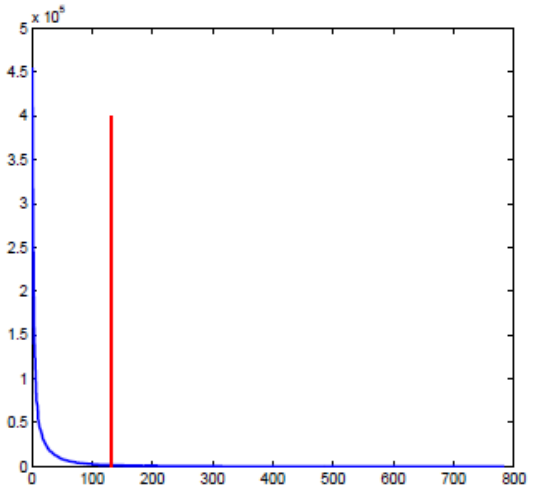


Выбор размерности редуцированного пространства

Поскольку собственные значения ковариационной матрицы **R** отсортированы в порядке убывания $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$

Критерий выбора размерности будет иметь вид:

$$d : \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq \eta, \text{где} \quad \eta = \{0.95, 0.99\}$$



Применение PCA

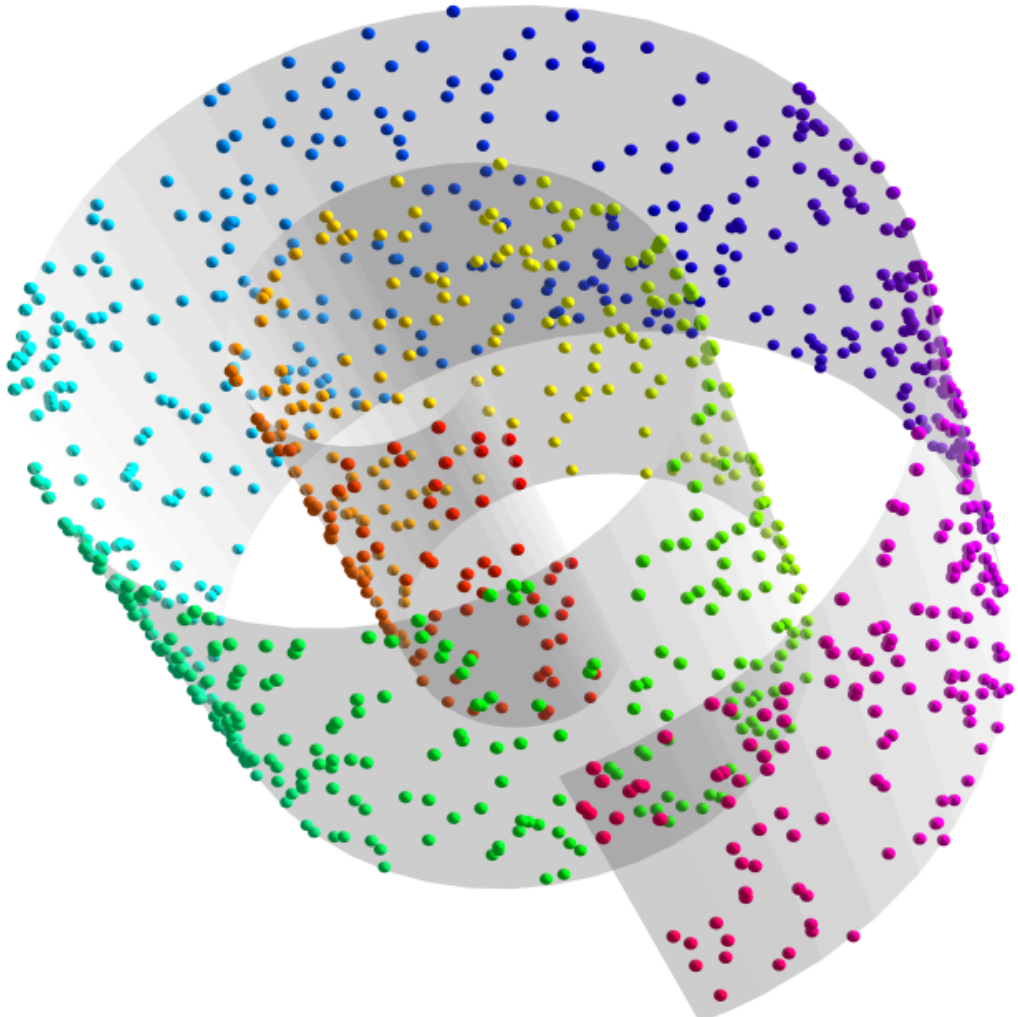
(a) Data Visualization

(b) Image procesing

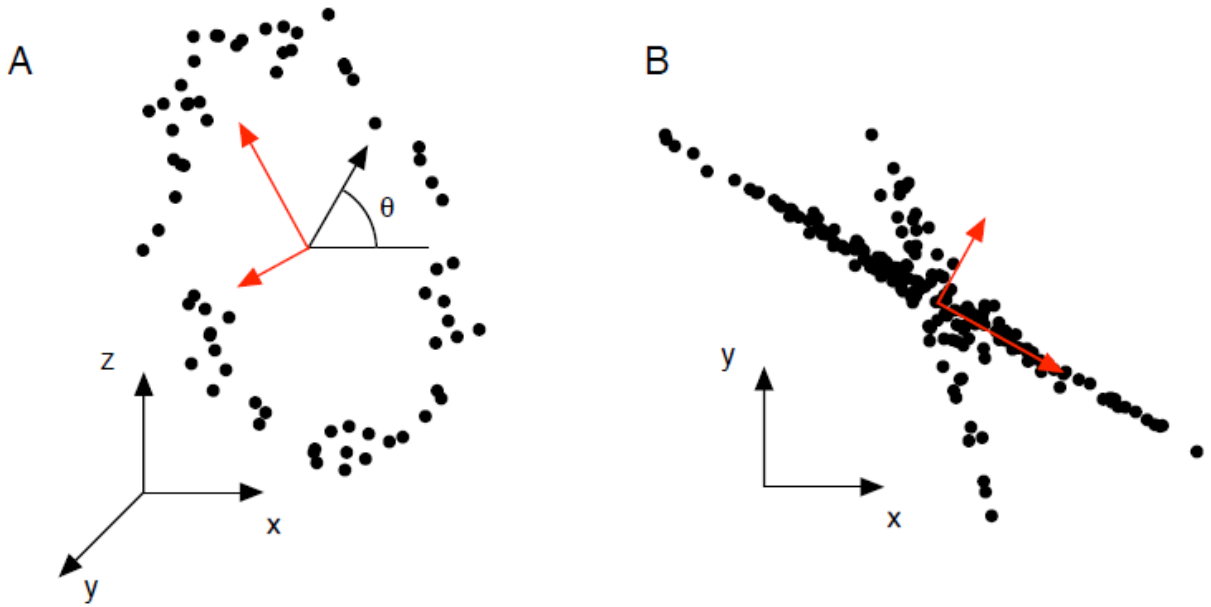
(c) Prospect

(d) Data compression

А всегда ли все хорошо?



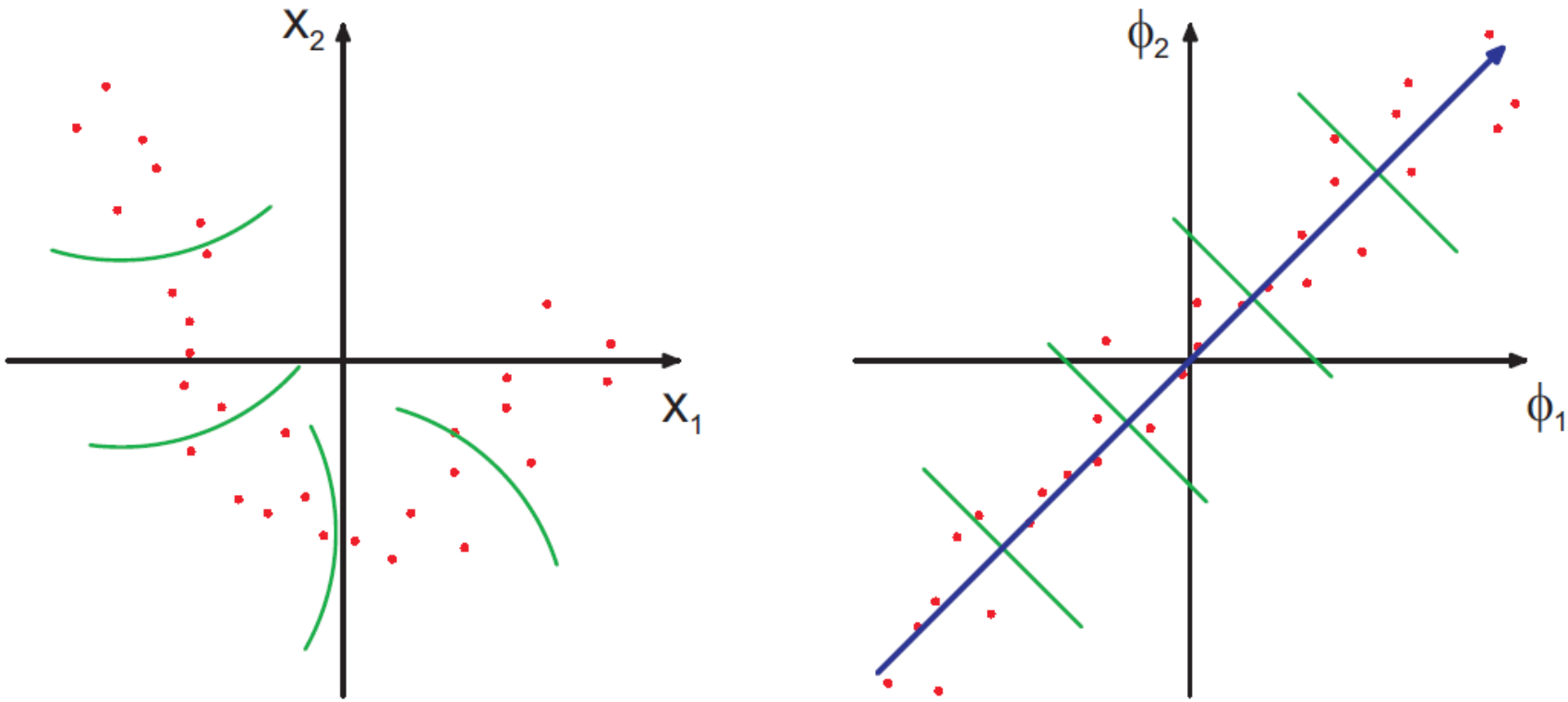
PCA Fails



Kernel PCA

Идея

Выберем некоторое нелинейное преобразование $\phi : R^D \to H$, при котором в новом пространстве нелинейное многообразие выборки переходит в гиперплоскость. (Как это сделать?)



Kernel PCA

Пусть далее известно, что скалярное произведение может быть вычислено с помощью функции от данных в исходном пространстве, тогда схему метода главных компонент можно переписать в терминах скалярных произведений

$$\phi(\mathbf{x})^T \phi(\mathbf{y}) = K(\mathbf{x}, \mathbf{y})$$

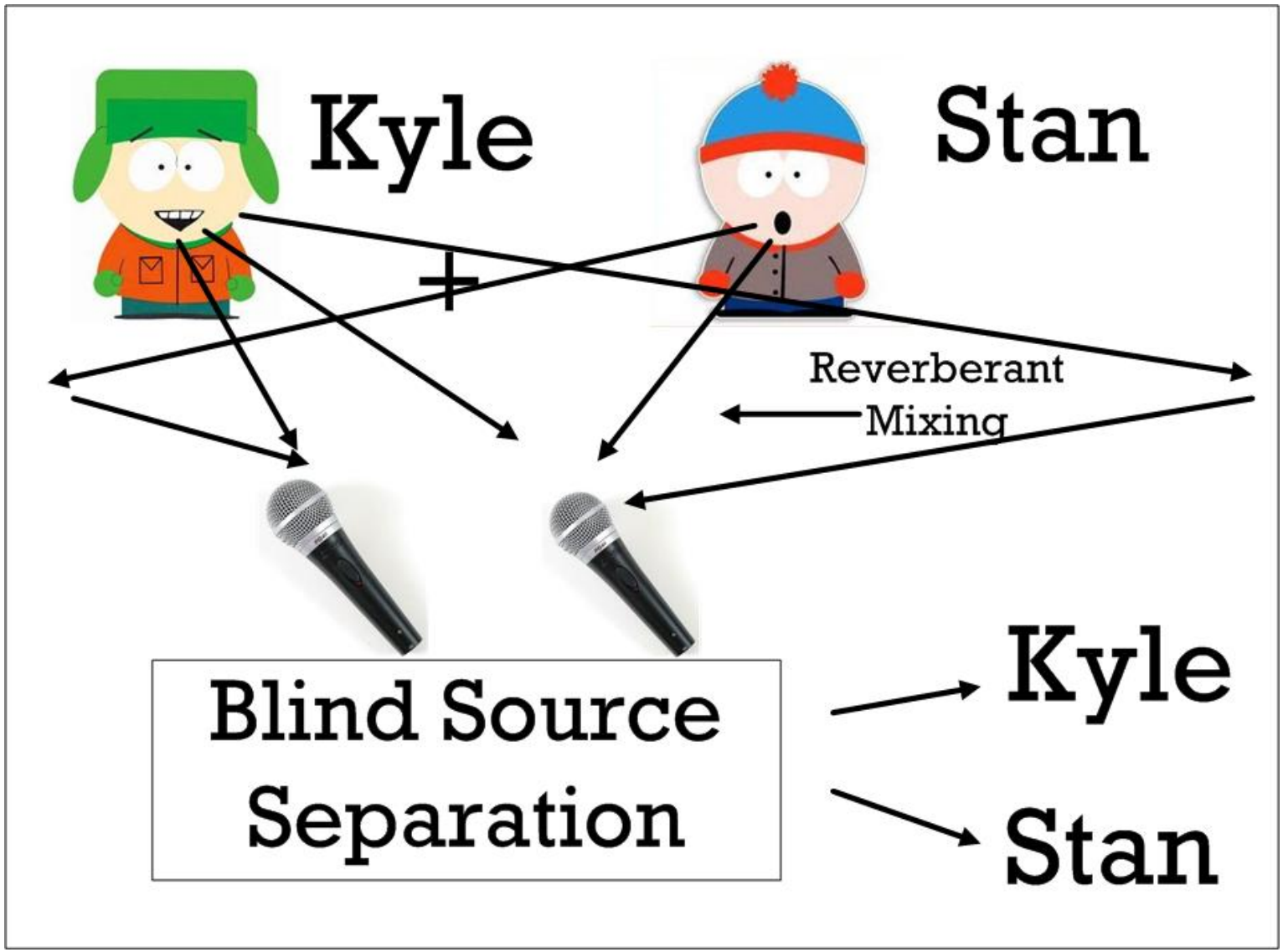
Достоиснтва и недостатки PCA

+ Алгоритм прост
+ С помощью “kernel trick” адаптируется на нелинейный случай (Kernel PCA)
— Проблема с вычислением собсвенных векторов ковариационной матрицы в случае большого количества данных
— Координаты объектов в новом пространстве определены неоднозначно

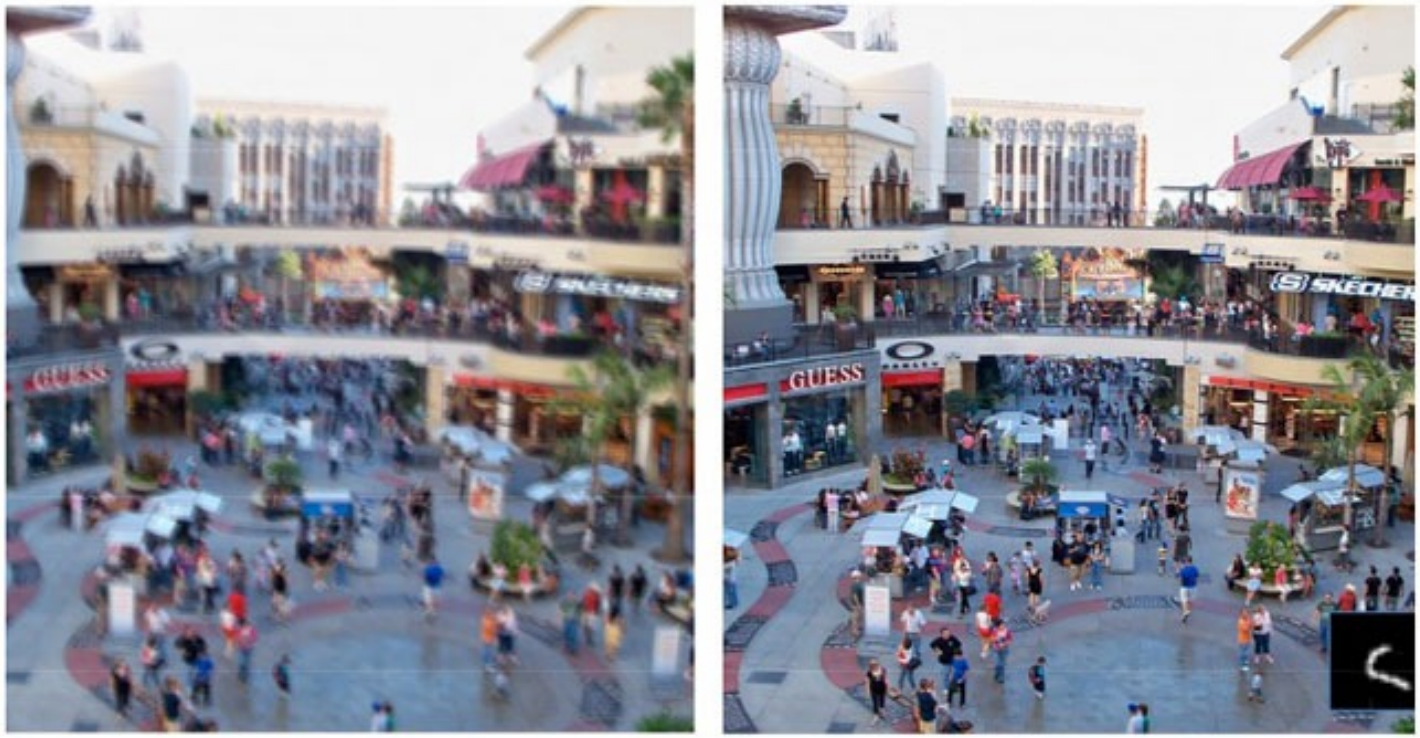
Вопрос:

- При каких условиях можно использовать представление данных в виде главных компонент для обучения?

Задача слепового разделения сигналов



Задача повышения четкости фото



blury image $=$ *original image* $*$ *motion trajectory*

Independent Component Analysis

$$\mathbf{X} = \mathbf{AS}$$

$$\mathbf{X}_j = a_{j,1}\mathbf{S}_1 + a_{j,2}\mathbf{S}_2 + \ldots + a_{j,N}\mathbf{S}_N,\ j = 1, \ldots, N$$

- $\mathbf{X}_j, \mathbf{S}_k$ - случайные величины
- $\mathbf{X}$ - наблюдаемые данные
- $\mathbf{A}$ - матрица смешивания
- $\mathbf{S}$ - неизвестный сигнал

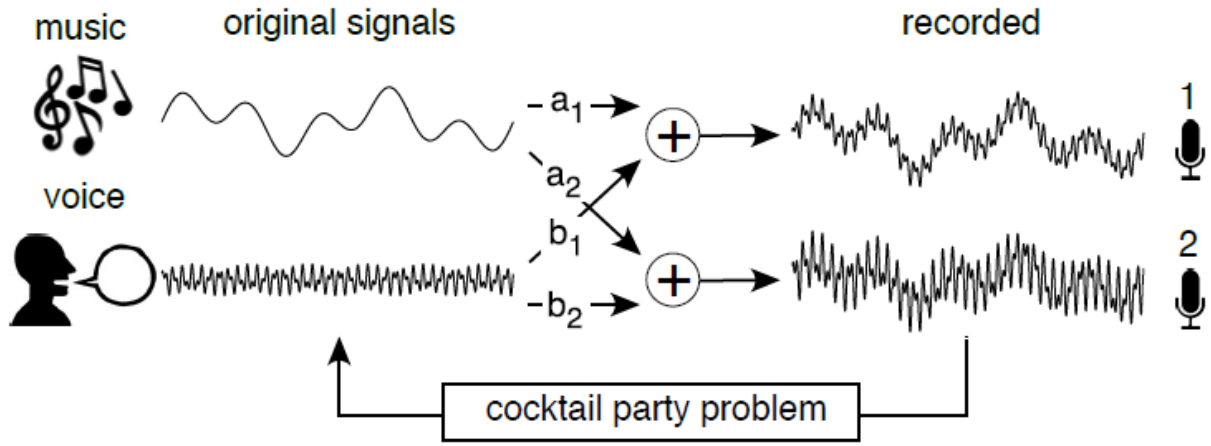
Задача:

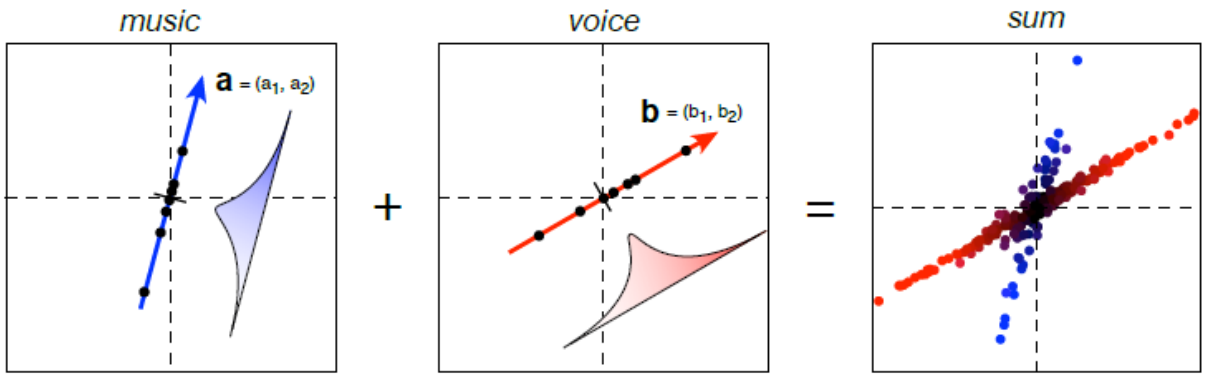
Оценить $\mathbf{A}$ и восстановить исходные сигналы $\hat{\mathbf{S}} = \mathbf{A}^{-1}\mathbf{X} = \mathbf{W}\mathbf{X}$.

Предположение:

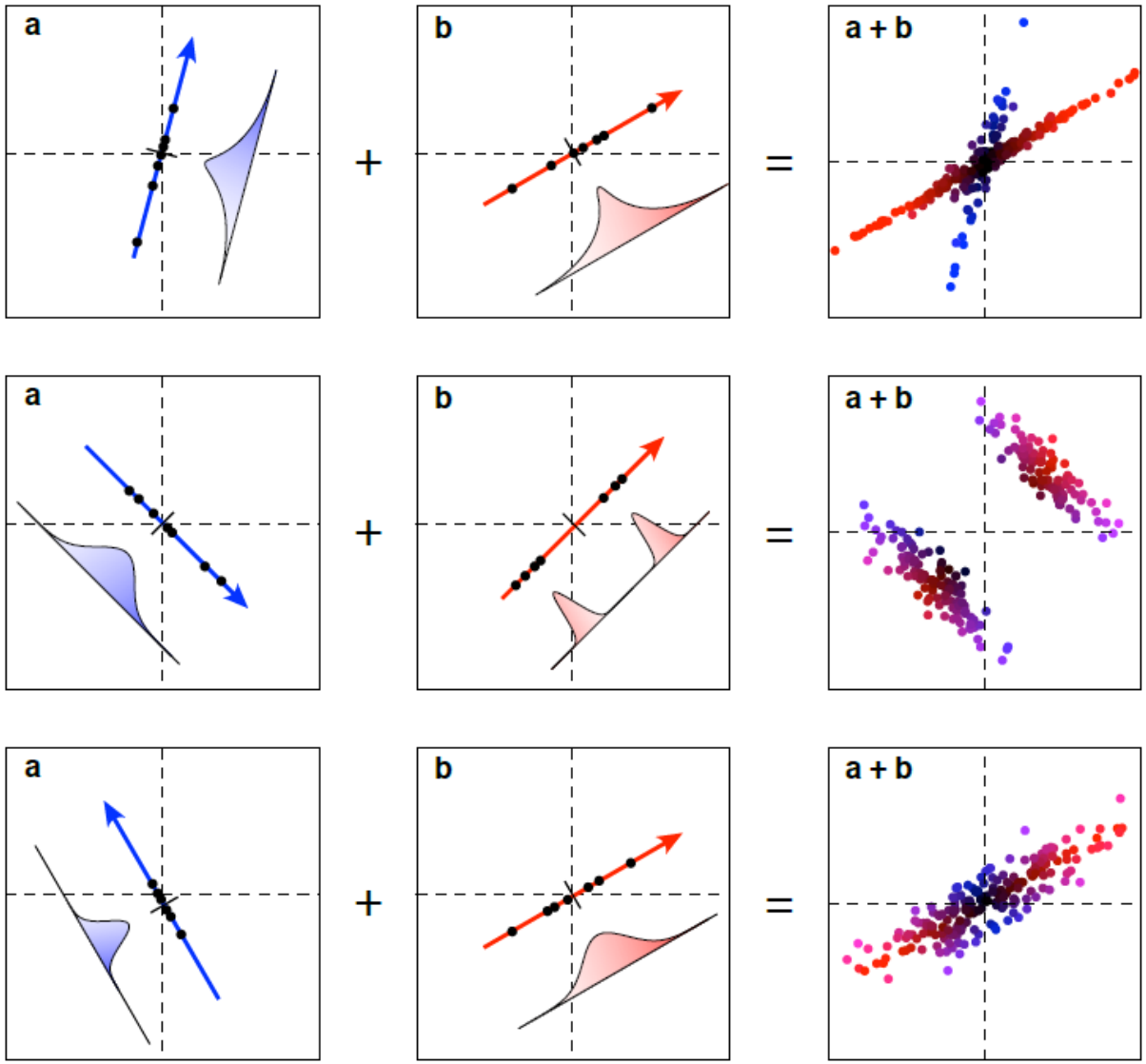
- $\mathbf{S}_i$ статистически независимы $p(\mathbf{S}_1, \mathbf{S}_2) = p(\mathbf{S}_1)p(\mathbf{S}_2)$

Cocktail party problem





Cocktail party problem



Whitening

SVD for $\mathbf{A}$

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

$$\mathbf{W} = \mathbf{A}^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$$

Предположим, что

$$\mathbf{S}\mathbf{S}^T = \mathbf{I}$$

Whitening

С одной стороны

$$\mathbf{X}\mathbf{X}^T = \mathbf{E}\mathbf{D}\mathbf{E}^T$$

С другой

$$\mathbf{X}\mathbf{X}^T = \mathbf{A}\mathbf{S}(\mathbf{A}\mathbf{S})^T =$$
$$= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{S}(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{S})^T =$$
$$= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T(\mathbf{S}\mathbf{S}^T)\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T$$
$$= \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$$

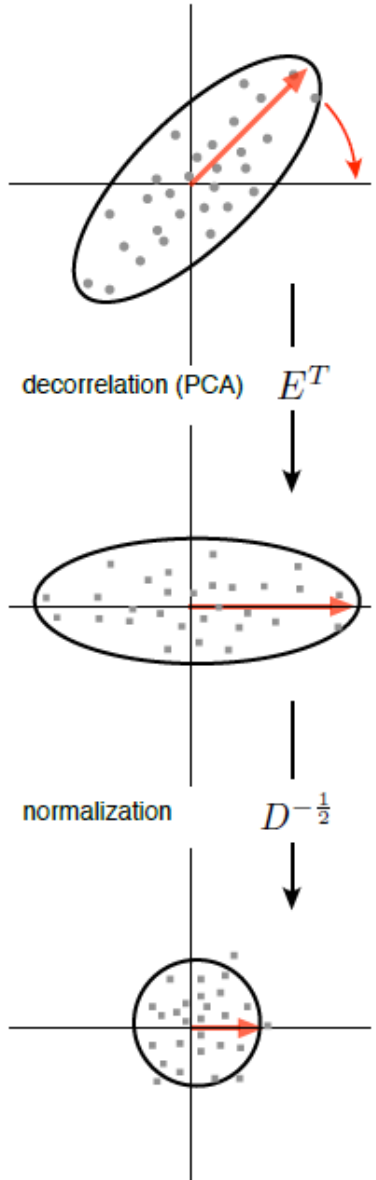
Таким образом

$$\mathbf{W} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{E}^T$$

- $\mathbf{D}$ - собственные значения выборочной ковариационной матрицы
- $\mathbf{E}$ - матрица собственных векторов ковариационной матрицы

Whitening

$$\mathbf{X}_w = \mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{X}, \quad \mathbf{X}_w\mathbf{X}_w^T = \mathbf{I}$$



The statistics of independence

Статистическая независимость

$$P(\mathbf{S}) = \prod_i P(\mathbf{S}_i)$$

Взаимная информация

$$I(\mathbf{y}) = \int P(\mathbf{y}) log_2 \frac{P(\mathbf{y})}{\prod_i P(\mathbf{y})} d\mathbf{y}$$

$$H[\mathbf{y}] = -\int P(\mathbf{y}) log_2 P(\mathbf{y}) d\mathbf{y}$$

$$I(\hat{\mathbf{S}}) = \sum_i H[(\mathbf{V}\mathbf{X}_w)_i] - H[\mathbf{V}\mathbf{X}_w] = \sum_i H[(\mathbf{V}\mathbf{X}_w)_i] - (H[\mathbf{X}_w] + log_2|\mathbf{V}|)$$

$$\mathbf{V} = arg \min_{\mathbf{V}} \sum_i H[(\mathbf{V}\mathbf{X}_w)_i]$$

Independent Component Analysis

Схема

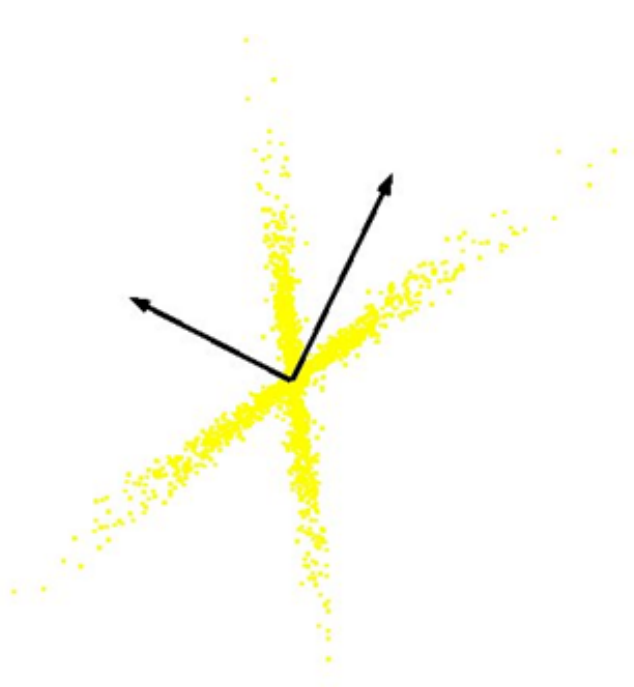
1. Центрируем данные $\mathbf{X_i} \leftarrow (\mathbf{X_i} - \bar{\mathbf{X}}) : \bar{\mathbf{X}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathbf{X_i}$
2. “Отбеливаем” данные

$$\mathbf{X}_w = \mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{X}$$

3. Находим ортогональную матрицу $\mathbf{V}$
   - Infomax
   - FastICA
   - JADE

PCA vs ICA

Геометрическая интерпретация

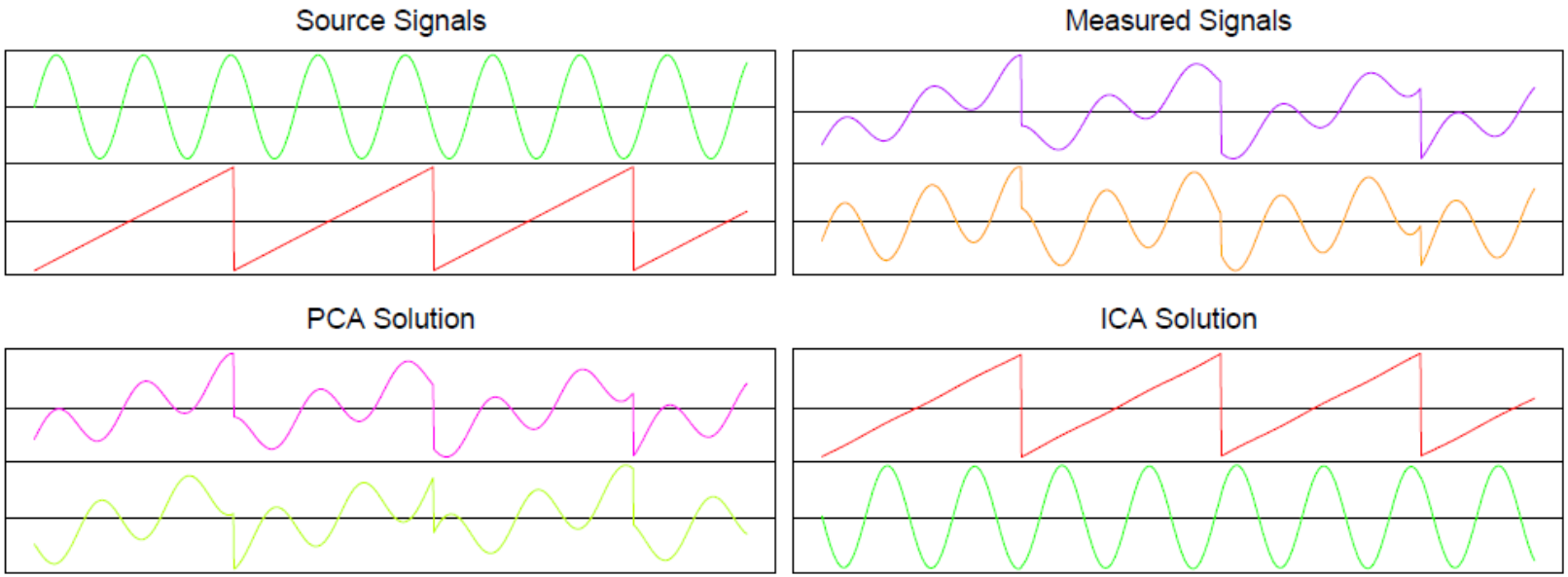


(a) PCA
(ортогональны)

(b) ICA
(не ортогональны)

PCA vs ICA

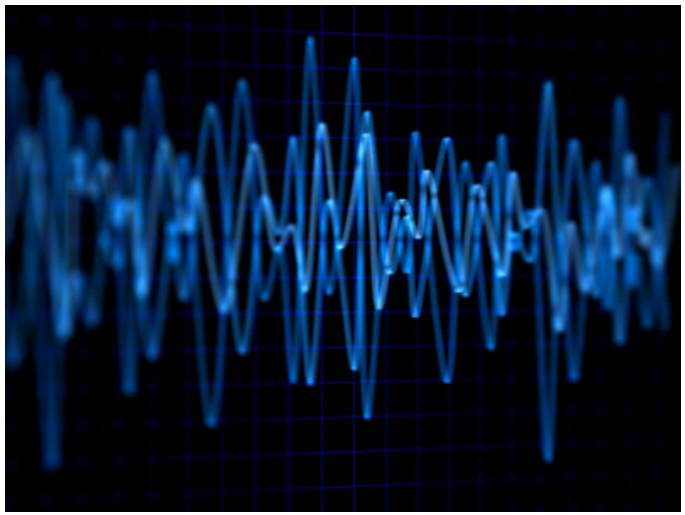
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$



- Сравнение PCA vs ICA на искуственном временном ряде, смоделированном по 1000 равномерно расспределенным точкам.

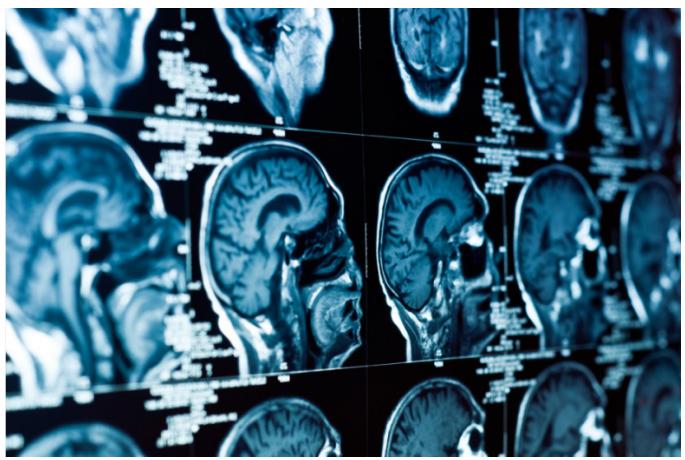
Применение ICA

(a) EEG



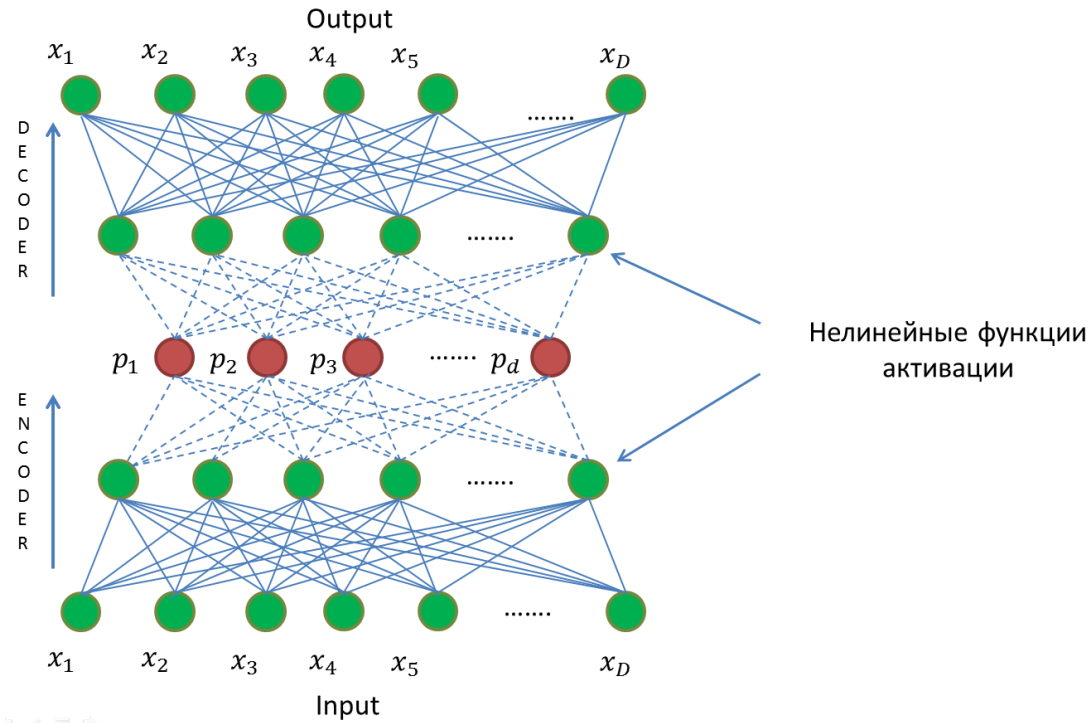
(b) Audio procesing

(c) Finance



(d) Medical data

Методы основанные на автоэнкодерах



$$J(\mathbf{w}) = \sum_{i=1}^{N} \|f(\mathbf{x}_i, \mathbf{w}) - \mathbf{x}_i\|^2 \to \min$$

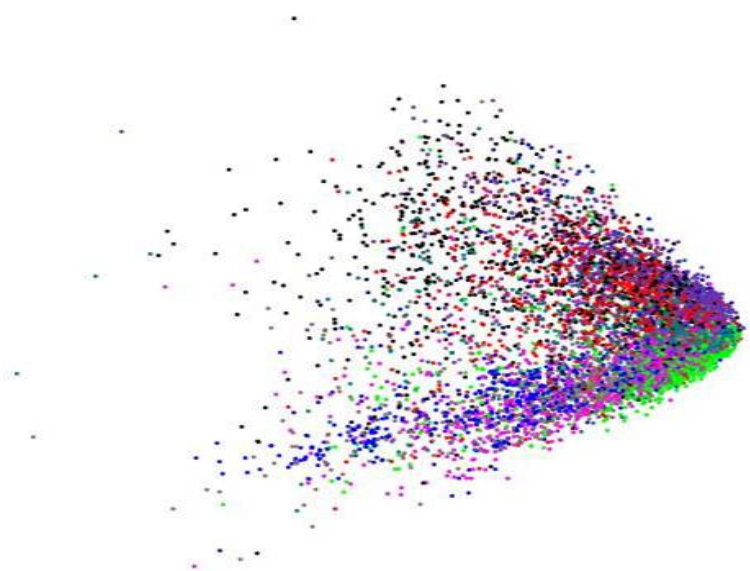
Замечание

Если в сети всего один скрытый слой, тогда результат эквивалентен PCA.

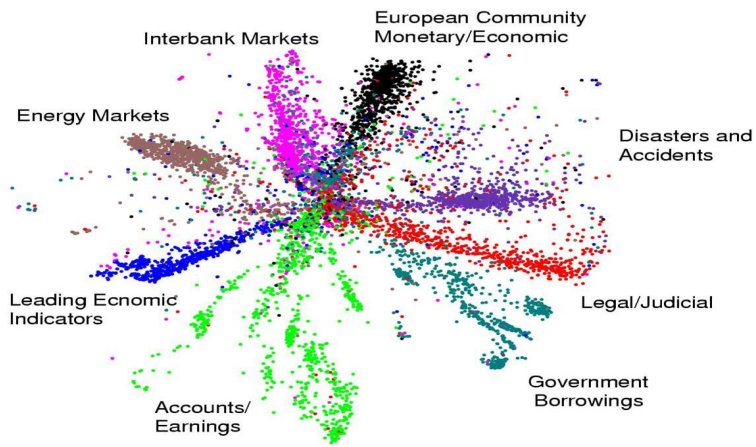
PCA vs Autoencoder

Задача визуализации тематических текстовых документов

- $D = 2000$ - “мешок слов”
- $N = 4 \cdot 10^5$ документов



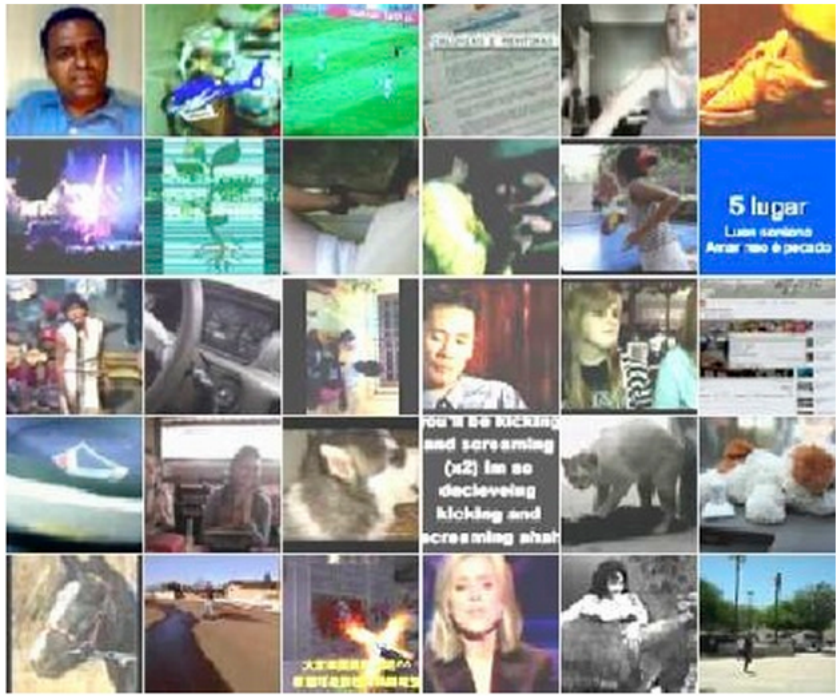
(a) PCA



(b) Deep Autoencoder

“Бабушкин” нейрон

- Andrew Ng
- 9-ти слойный разряженный автоэнкодер
- Асинхронный градиентный спуск
- 10 млн. кадров случайно взятых из роликов youtube
- Удалось найти нейрон, отвечающий за наличие лица в кадре



- http://habrahabr.ru/post/146077/

Вопросы