

# Методы понижения размерности данных

---

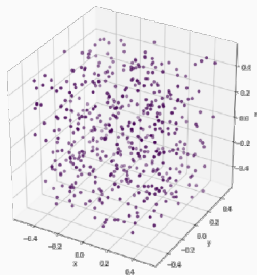
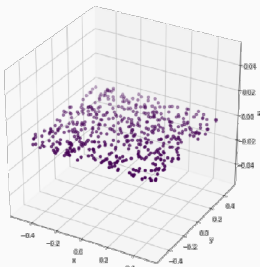
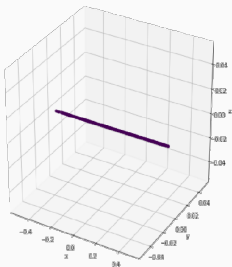
Поглазов Никита

2024

## Введение

---

**Проклятие размерности:** данные высокой размерности сложны для анализа, требуют много вычислительных ресурсов и часто содержат шум.

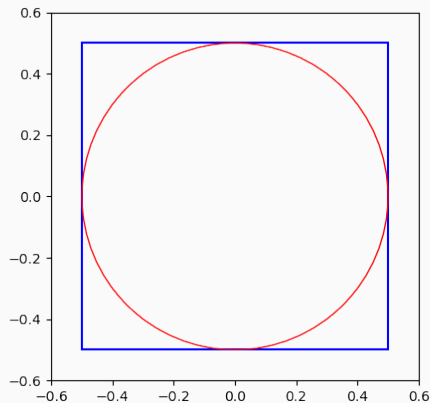


## Мотивация

---

## Что такое "проклятие размерности"?

$$S_{square} = 1 \quad S_{circle} = \pi * (0.5)^2 = \frac{\pi}{4} \approx 0.79$$

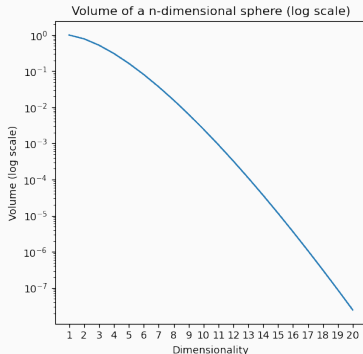
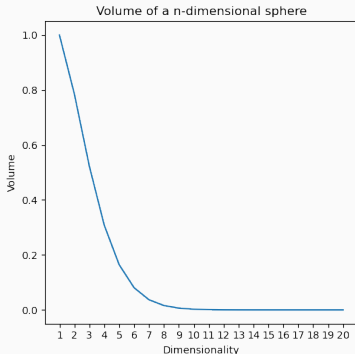


# Гиперсфера и гиперкуб

- Объем гиперсферы стремится к нулю при росте размерности:

$$V_n = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2} + 1)} R^n$$

- Диагональ гиперкуба увеличивается как  $\sqrt{n}$ .



# Влияние на метрические модели (1)

- Манхэттенское расстояние:

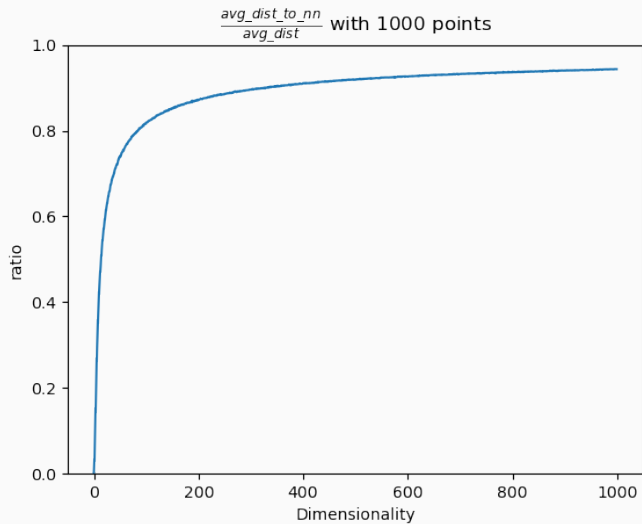
$$d(x^{(i)}, x^{(j)}) = \sum_{k=1}^n |x_k^{(i)} - x_k^{(j)}|$$

- Средние расстояния между точками становятся близкими:

$$\lim_{n \rightarrow \infty} \frac{d(x^{(i)}, x^{(j)})}{n} = \mu$$

- Сохраняется и для  $L_2$  нормы.

## Влияние на метрические модели (2)





# Линейная регрессия и мультиколлинеарность

- Решение задачи MSE:

$$(X^T X) \hat{\beta} = X^T y$$

- Матричная ковариация:

$$\text{Cov}(X) = \frac{1}{k-1} X^T X$$

- Высокая корреляция между признаками  $\Rightarrow$  нестабильные веса, переобучение.

## Влияние на "деревянные" модели

- Сложность выбора оптимального разделения при высокой размерности.
- Деревья склонны к переобучению из-за случайных разбиений.
- **Workaround:** *Random Subspace Method* (Ho)

## Влияние на глубокие нейронные сети

- Сверточные сети (*CNN*) используют локальные взаимосвязи.
- *LSTM* моделируют временные зависимости, игнорируя пространственные.
- Трансформеры извлекают только значимые зависимости.
- Проблемы: обучение на шуме, сложность оптимизации функционала потерь.

## Общее влияние "проклятия размерности"

- Увеличение времени обучения моделей.
- Сложность интерпретации табличных данных.
- Вероятность обучения на шумовых признаках  $\Rightarrow$  переобучение.

## Обзор и классификация методов

---

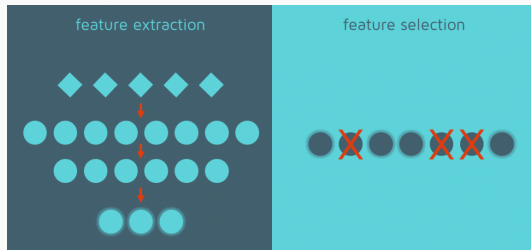
# Два подхода к понижению размерности

## Отбор признаков:

- Выбор подмножества исходных признаков.
- Сохранение информации без преобразования данных.

## Преобразование признаков:

- Трансформация данных в новое пространство меньшей размерности.
- Сохраняет наиболее значимые свойства данных.

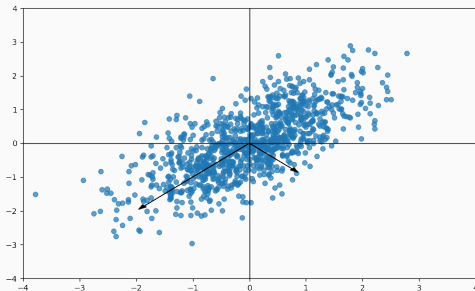


# Principal Component Analysis (PCA)

**Цель:** Сохранить максимальную дисперсию данных.

**Пример применения:**

- Визуализация данных высокой размерности (например, геномика).
- Уменьшение размерности для кластеризации образцов.



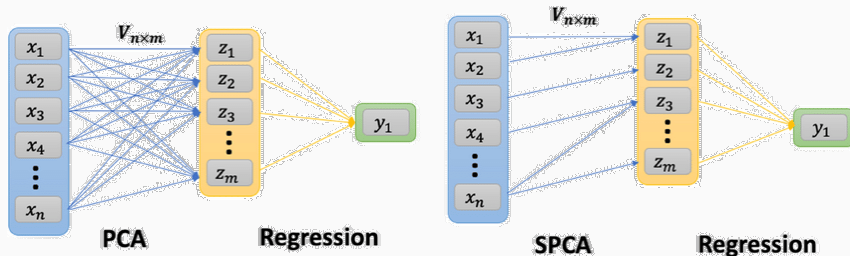
# Sparse PCA (SPCA)

## Отличие от PCA:

- Ограничение на разреженность главных компонент.
- Уменьшает сложность интерпретации данных.

## Применение:

- Анализ данных с множеством нерелевантных признаков (например, финансовые индикаторы).



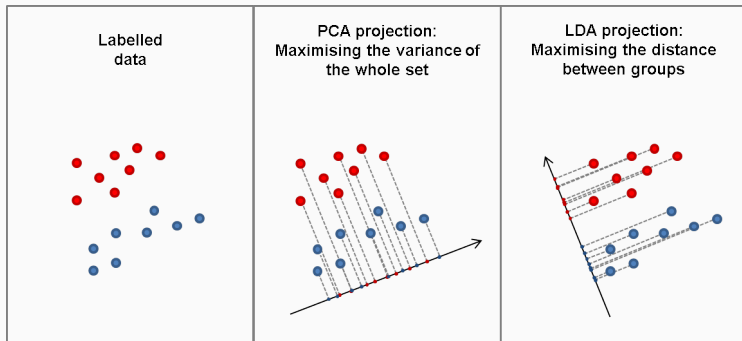


# Linear Discriminant Analysis (LDA)

**Цель:** Максимизация различий между классами.

**Пример применения:**

- Распознавание лиц в биометрии.
- Классификация текстов по категориям.



# Canonical Correlation Analysis (CCA)

**Цель:** Найти коррелирующие компоненты в двух наборах данных.

**Пример применения:**

- Связь между анкетными данными и биометрией.
- Исследование двух источников данных для выявления зависимостей.

**Ключевая идея:** *Kernel Trick* для проецирования в нелинейное пространство.

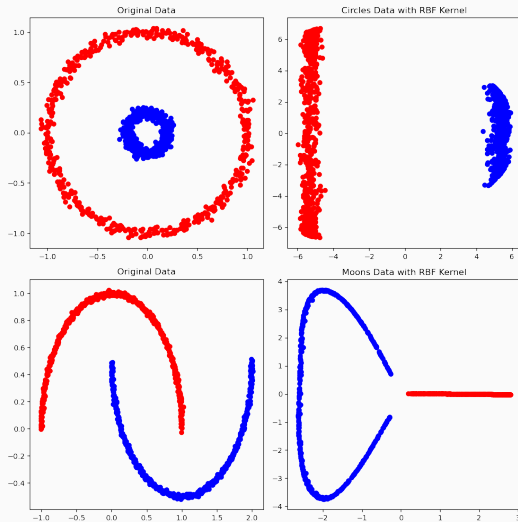
Пример ядерной функции (гауссовское ядро):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left( -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right)$$

**Пример применения:**

- Обнаружение сложных текстур на изображениях.
- Биоинформатика: анализ активности молекул.

# Kernel PCA (KPCA)



Алгоритм визуализации (!) данных высокой размерности.

**Цель:** Локальное сохранение расстояний между точками.

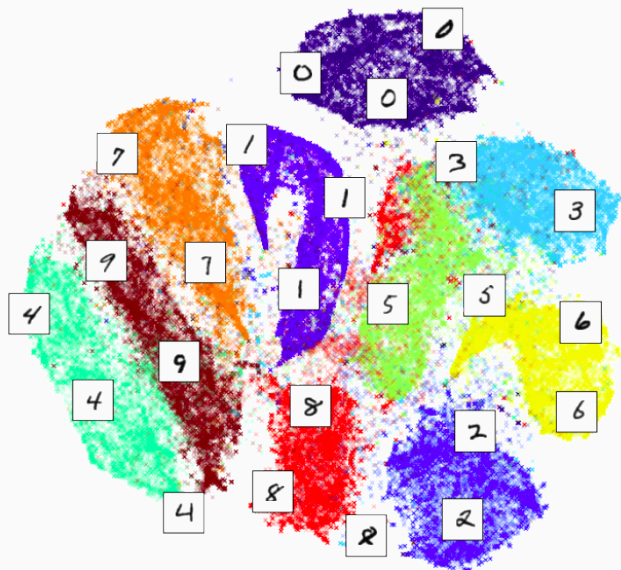
Основная идея:

- Перевод данных в вероятностное представление.
- Минимизация расстояния Кульбака-Лейблера (KL-дивергенция).

**Пример применения:**

- Визуализация эмбеддингов слов или изображений.
- Кластеризация геномных данных.

# t-SNE



# UMAP (*Uniform Manifold Approximation and Projection*)

**Цель:** Сохранение как локальных, так и глобальных структур данных.

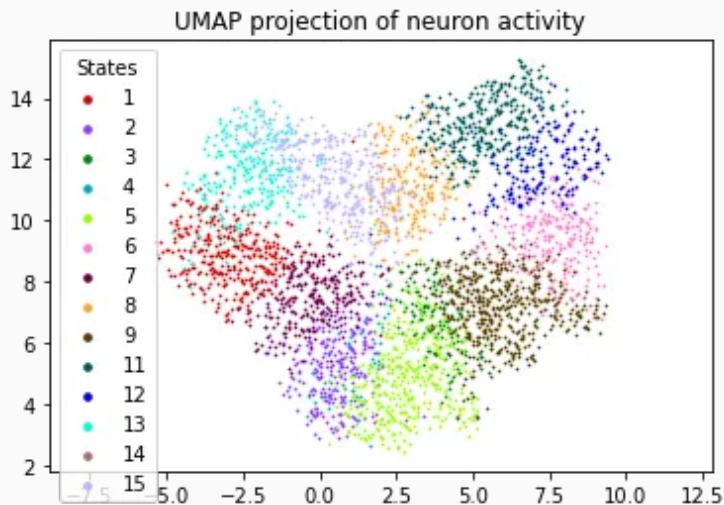
Основные этапы метода:

- Построение графа соседей данных в исходном пространстве.
- Оптимизация аппроксимации графа в пространстве меньшей размерности.

**Пример применения:**

- Визуализация паттернов активности мозга.
- Анализ биоинформационных данных.

## UMAP (Uniform Manifold Approximation and Projection)





# AutoEncoders (AEs)

**Цель:** Нахождение компактных нелинейных представлений данных.

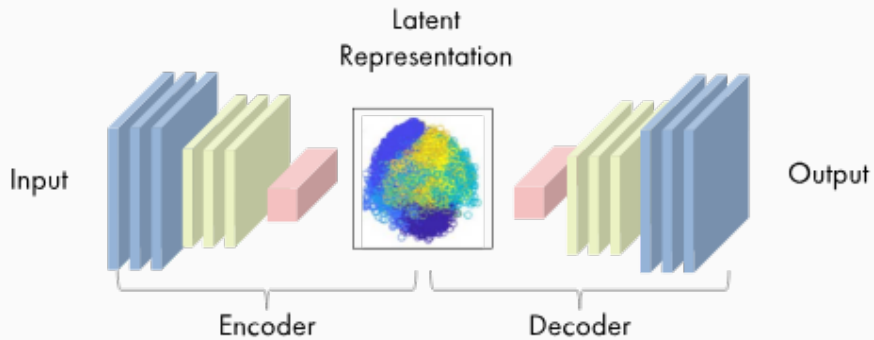
Основная структура:

- Кодировщик (encoder): преобразует входные данные в компактное представление.
- Декодировщик (decoder): восстанавливает данные из сжатого представления.

**Пример применения:**

- Удаление шума с изображений.
- Выделение особенностей для классификации.

# AutoEncoders (AEs)



# Variational AutoEncoders (VAEs)

**Расширение автоэнкодеров:** генерация данных на основе латентного пространства.

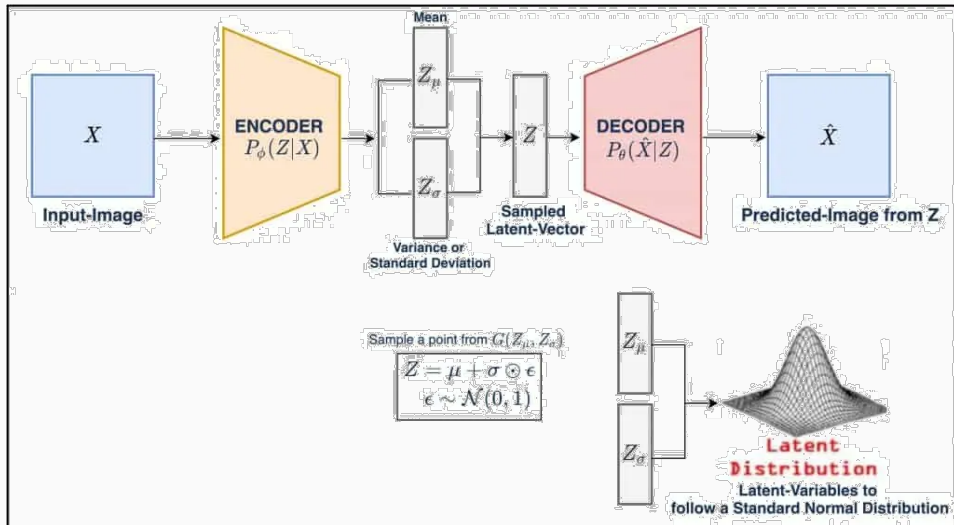
Основная идея:

- Представление латентного пространства в виде вероятностного распределения.

**Пример применения:**

- Генерация новых молекул с заданными свойствами.
- Создание искусственных изображений.

# Variational AutoEncoders (VAEs)



## Principal Component Analysis (*PCA*)

---

## Постановка задачи (1)

Дан неразмеченный датасет  $X = \{\mathbf{x}_i\}_{i=1}^N$ , где  $\mathbf{x}_i \in \mathbb{R}^D$ . Предполагаем центрированность данных:  $\mathbb{E}[\mathbf{x}_i] = 0$ .

Матрица ковариации данных:

$$\Sigma = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T.$$

Переход в новое пространство меньшей размерности (сжатие):

$$\mathbf{z}_i = \mathbf{B}^T \mathbf{x}_i \in \mathbb{R}^M, \quad M < D,$$

## Постановка задачи (2)

Базис  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_M] \in \mathbb{R}^{D \times M}$  удовлетворяет:

$$\mathbf{b}_i^T \mathbf{b}_j = \delta_{ij} = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}$$

Восстановление данных:

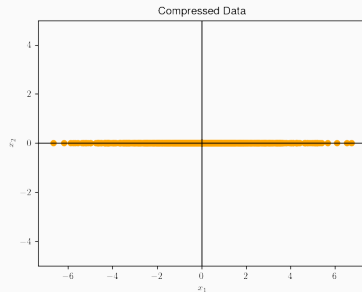
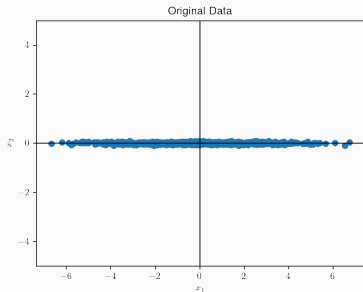
$$\tilde{\mathbf{x}}_i = \mathbf{B} \mathbf{z}_i.$$

## Пример: 2D $\rightarrow$ 1D

Исходный вектор:  $\mathbf{x}_i \in \mathbb{R}^2$ ,  $\mathbf{x}_i = \begin{bmatrix} 5 \\ \frac{1}{100} \end{bmatrix}$ . Выбираем базис  $\mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ .

Шаги:

- Координаты в новом базисе:  $\mathbf{z}_i = \mathbf{B}^T \mathbf{x}_i = 5$ .
- Восстановленный вектор:  $\tilde{\mathbf{x}}_i = \mathbf{B} \mathbf{z}_i = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ .





## Нахождение направления максимальной дисперсии (1)

**Цель:** Найти направление  $\mathbf{b}_1$ , вдоль которого дисперсия данных максимальна.

Дисперсия вдоль первой координаты в новом пространстве:

$$\begin{aligned} V_1 := \mathbb{D}[z_1] &= \frac{1}{N} \sum_{i=1}^N z_{1i}^2 = \frac{1}{N} \sum_{i=1}^N (\mathbf{b}_1^T \mathbf{x}_i)^2 = \frac{1}{N} \sum_{i=1}^N (\mathbf{b}_1^T x_i x_i^T \mathbf{b}_1) \\ &= \mathbf{b}_1^T \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right) \mathbf{b}_1 = \mathbf{b}_1^T \Sigma \mathbf{b}_1. \end{aligned}$$

## Нахождение направления максимальной дисперсии (2)

Задача условной оптимизации:

$$\max_{\mathbf{b}_1} \mathbf{b}_1^T \Sigma \mathbf{b}_1, \quad \text{s.t. } \mathbf{b}_1^T \mathbf{b}_1 = 1.$$

Функция Лагранжа:  $\mathcal{L}(\mathbf{b}_1, \lambda) = \mathbf{b}_1^T \Sigma \mathbf{b}_1 - \lambda(\mathbf{b}_1^T \mathbf{b}_1 - 1)$ .

Частные производные по  $\mathbf{b}_1$  и  $\lambda$ :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{b}_1} &= 2\Sigma \mathbf{b}_1 - 2\lambda_1 \mathbf{b}_1 = 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda_1} &= -\mathbf{b}_1^T \mathbf{b}_1 + 1 = 0. \end{aligned}$$

## Собственные векторы и значения

Получаем:

$$\begin{aligned}\Sigma \mathbf{b}_1 &= \lambda_1 \mathbf{b}_1, \\ V_1 &= \lambda_1.\end{aligned}$$

Теперь можем переписать дисперсию  $V_1$  как:

$$V_1 = \mathbf{b}_1^T \Sigma \mathbf{b}_1 = \lambda_1 \mathbf{b}_1^T \mathbf{b}_1 = \lambda_1.$$

Интерпретация:

- $\mathbf{b}_1$ : первое направление главной компоненты.
- $\lambda_1$ : дисперсия вдоль направления  $\mathbf{b}_1$ .

## Остальные компоненты (1)

Для  $m$ -й компоненты:

$$\begin{aligned} & \max_{\mathbf{b}_m} \mathbf{b}_m^T \Sigma \mathbf{b}_m, \\ & \text{s.t. } \mathbf{b}_m^T \mathbf{b}_m = 1, \quad \mathbf{b}_m^T \mathbf{b}_i = 0, \forall i < m. \end{aligned}$$

Функция Лагранжа:

$$\mathcal{L}(\mathbf{b}_m, \lambda_m, \boldsymbol{\mu}) = \mathbf{b}_m^T \Sigma \mathbf{b}_m - \lambda_m (\mathbf{b}_m^T \mathbf{b}_m - 1) - \sum_{i=1}^{m-1} \mu_i \mathbf{b}_m^T \mathbf{b}_i.$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{b}_m} &= 2\Sigma \mathbf{b}_m - 2\lambda_m \mathbf{b}_m - \sum_{i=1}^{m-1} \mu_i \mathbf{b}_i = 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda_m} &= -\mathbf{b}_m^T \mathbf{b}_m + 1 = 0, \quad \frac{\partial \mathcal{L}}{\partial \mu_i} = -\mathbf{b}_m^T \mathbf{b}_i = 0, \forall i < m. \end{aligned}$$

Домножим первое уравнение на  $\mathbf{b}_j^T$ ,  $j < m$  слева:

## Остальные компоненты (2)

$$2\mathbf{b}_j^T \Sigma \mathbf{b}_m - 2\lambda_m \mathbf{b}_j^T \mathbf{b}_m - \sum_{i=1}^{m-1} \mu_i \mathbf{b}_j^T \mathbf{b}_i = 0,$$

поскольку  $\mathbf{b}_j^T \mathbf{b}_i = \delta_{ji}$ :

$$2\mathbf{b}_j^T \Sigma \mathbf{b}_m - \mu_j = 0.$$

$\Sigma$  симметрична, поэтому

$$\mathbf{b}_j^T \Sigma \mathbf{b}_m = \langle (\mathbf{b}_j^T \Sigma)^T, \mathbf{b}_m \rangle = \langle \Sigma \mathbf{b}_j, \mathbf{b}_m \rangle = \langle \lambda_j \mathbf{b}_j, \mathbf{b}_m \rangle = \lambda_j \langle \mathbf{b}_j, \mathbf{b}_m \rangle = 0.$$

Тогда  $\mu_j = 0$ . и, аналогично,  $\forall j < m \quad \mu_j = 0$

## Остальные компоненты (3)

Таким образом:

$$\Sigma \mathbf{b}_m = \lambda_m \mathbf{b}_m,$$

Вновь,  $\mathbf{b}_m$  - собственный вектор матрицы ковариации  $\Sigma$ , а  $\lambda_m$  - собственное значение.

Общая дисперсия:  $\sum_{i=1}^N \lambda_i$ .

Объясненная дисперсия первых  $m$  главных компонент:  $\sum_{i=1}^m \lambda_i$ .

Доля объясненной дисперсии:  $\frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^N \lambda_i}$ .

Формулы:

$$\begin{aligned}\mathbf{Z} &= \mathbf{B}^T \mathbf{X}, \\ \tilde{\mathbf{X}} &= \mathbf{B} \mathbf{Z}, \\ \Sigma &= \frac{1}{N} \mathbf{X} \mathbf{X}^T.\end{aligned}$$

**Примечание:** строки  $\mathbf{X}$  — признаки, столбцы — объекты.





## Kernel PCA (KPCA)

---

## Постановка задачи (1)

Дан центрированный неразмеченный датасет  $X = \{\mathbf{x}_i\}_{i=1}^N$ ,  $\mathbf{x}_i \in \mathbb{R}^D$ .

Задано:

- Преобразование  $\phi : \mathbb{R}^D \rightarrow \mathbb{H}$ , где  $\mathbb{H}$  — гильбертово пространство.
- Функция (ядро)  $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R} : \quad k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathbb{H}}$ .

**Цель:** Найти линейное подпространство в  $\mathbb{H}$  размерности  $P$ , минимизирующее расстояние между  $x_i$  и их проекцией.

## Постановка задачи (2)

### Свойства ядерных функций:

- **Утверждение:** по произвольной функции  $\phi$  можно построить ядро  $k$  - положительно определенная функция.
- **Теорема Moore-Aronszajn:** По положительно определённом ядру  $k$  можно построить  $\phi$  и пространство  $\mathbb{H}$ .
- Матрица Грама  $\mathbf{K} \in \mathbb{R}^{N \times N}$ :

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j).$$

**Пространство:** Пусть  $\mathbb{H} = \mathbb{R}^H$ , где  $H \gg D$  (для конечномерного случая).

# Наивный подход

## Шаги:

1. Вычислить  $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ .
2. Применить PCA к  $\{\phi(\mathbf{x}_i)\}_{i=1}^N$ .

## Проблемы:

- Вычисление  $\phi(\mathbf{x}_i)$  дорого.
- $\phi$  может быть неизвестным.
- Ковариационная матрица размера  $H \times H$ , где  $H \gg D$ .

## Kernel Trick (1)

**Подход:** Составим из  $\phi(\mathbf{x}_i)$  матрицу  $\Phi$  ( $N \times H$ ). Матрица ковариации:

$$\Sigma = \frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T = \frac{1}{N} \Phi^T \Phi.$$

Главные компоненты  $\omega_p \in \mathbb{H}$ :

$$\Sigma \omega_p = \lambda_p \omega_p \quad \text{для } p = 1, 2, \dots, P.$$

## Kernel Trick (2)

Подставим  $\Sigma$ :

$$\frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \omega_p = \frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x}_i) \langle \phi(\mathbf{x}_i), \omega_p \rangle_{\mathbb{H}} = \lambda_p \omega_p.$$

Представление компонент:

$$\omega_p = \sum_{j=1}^N \alpha_{p,j} \phi(\mathbf{x}_j), \quad \alpha_{p,j} = \langle \phi(\mathbf{x}_j), \omega_p \rangle_{\mathbb{H}}.$$

## Kernel Trick (3)

Подставим это в уравнение для  $\omega_p$ :

$$\frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x}_i) \langle \phi(\mathbf{x}_i), \sum_{j=1}^N \alpha_{p,j} \phi(\mathbf{x}_j) \rangle_{\mathbb{H}} = \lambda_p \sum_{i=1}^N \alpha_{p,i} \phi(\mathbf{x}_i),$$

$$\frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \sum_{j=1}^N \phi(\mathbf{x}_j) \alpha_{p,j} = \lambda_p \sum_{j=1}^N \alpha_{p,j} \phi(\mathbf{x}_j),$$

$$\frac{1}{N} \Phi^T \Phi \Phi^T \boldsymbol{\alpha}_p = \lambda_p \Phi^T \boldsymbol{\alpha}_p,$$

$$\Phi^T (\Phi \Phi^T \boldsymbol{\alpha}_p - N \lambda_p \boldsymbol{\alpha}_p) = 0$$

$$\mathbf{K} \boldsymbol{\alpha}_p = N \lambda_p \boldsymbol{\alpha}_p, \quad \mathbf{K} = \Phi \Phi^T, \quad K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j).$$

# Проекции на главные компоненты

Проекции на главные компоненты вычисляются **даже без знания  $\phi$** :

$$\begin{aligned}\mathbf{z}_{ij} &= \langle \phi(\mathbf{x}_i), \omega_j \rangle_{\mathbb{H}} = \boldsymbol{\omega}_j^T \phi(\mathbf{x}_i) = \sum_{k=1}^N \alpha_{j,k} \phi(\mathbf{x}_k)^T \phi(\mathbf{x}_i) \\ &= \sum_{k=1}^N \alpha_{j,k} k(\mathbf{x}_k, \mathbf{x}_i) = \sum_{k=1}^N \alpha_{j,k} \mathbf{K}_{ki} = \sum_{k=1}^N \alpha_{j,k} \mathbf{K}_{ik} = \mathbf{K}_i \boldsymbol{\alpha}_j.\end{aligned}$$

$$\mathbf{Z} = \mathbf{K} \boldsymbol{\alpha}.$$



## Центрирование образов (1)

**Проблема:** Образы  $\phi(\mathbf{x}_i)$  могут быть нецентрированными, даже если  $\mathbf{x}_i$  центрированы.

**Коррекция:**

$$\tilde{\phi}(\mathbf{x}) = \phi(\mathbf{x}) - \frac{1}{N} \sum_{i=1}^N \phi(\mathbf{x}_i).$$

## Центрирование образов (2)

Обновление ядра:

$$\begin{aligned}\tilde{k}(\mathbf{x}, \mathbf{y}) = & k(\mathbf{x}, \mathbf{y}) - \frac{1}{N} \sum_{i=1}^N (k(\mathbf{x}, \mathbf{x}_i) - k(\mathbf{x}_i, \mathbf{y})) \\ & + \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N k(\mathbf{x}_i, \mathbf{x}_j).\end{aligned}$$

Центрированная матрица:

$$\tilde{\mathbf{K}} = \left( \mathbf{E} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \right) \mathbf{K} \left( \mathbf{E} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \right),$$

где  $\mathbf{1}$  — вектор из единиц.

Наиболее популярное ядро: Гауссово (RBF):

$$k(\mathbf{x}, \mathbf{y}) = \exp \left( -\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2} \right) = \exp \left( -\gamma \|\mathbf{x} - \mathbf{y}\|^2 \right) .$$

Альтернативные ядра:

- Полиномиальное:  $k(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}^T \mathbf{y} + r)^d$ .
- Сигмоидальное:  $k(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \mathbf{x}^T \mathbf{y} + r)$ .
- Линейное:  $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$ .



## AutoEncoders (AEs)

---

## Постановка задачи

Дан неразмеченный датасет  $X = \{\mathbf{x}_i\}_{i=1}^N$ , где  $\mathbf{x}_i \in \mathbb{R}^D$ .

**Цель:** Найти сжатое представление  $\mathbf{z}_i \in \mathbb{R}^M$ ,  $M < D$ , такое что восстановленные данные  $\tilde{\mathbf{x}}_i$  близки к исходным  $\mathbf{x}_i$ .

**Подход:** Использование нелинейных преобразований, реализованных нейронной сетью.

# Архитектура автоэнкодера (1)

**Encoder:** Нелинейное отображение  $f_{\theta} : \mathbb{R}^D \rightarrow \mathbb{R}^M$ :

$$\mathbf{z} = f_{\theta}(\mathbf{x}),$$

где:

$$f_{\theta} = f_L \circ f_{L-1} \circ \dots \circ f_1, \quad f_i = \sigma_{Ei}(\mathbf{W}_{Ei}\mathbf{z}_{i-1} + \mathbf{b}_{Ei}).$$

**Decoder:** Восстановление  $g_{\phi} : \mathbb{R}^M \rightarrow \mathbb{R}^D$ :

$$\tilde{\mathbf{x}} = g_{\phi}(\mathbf{z}),$$

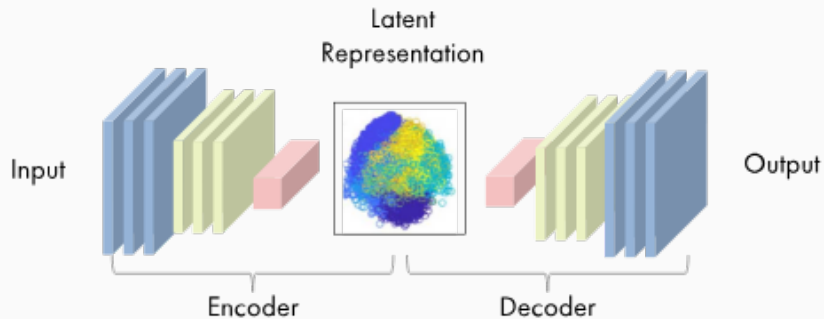
где:

$$g_{\phi} = g_L \circ g_{L-1} \circ \dots \circ g_1, \quad g_i = \sigma_{Di}(\mathbf{W}_{Di}\mathbf{z}_{i-1} + \mathbf{b}_{Di}).$$

**Полная архитектура:**

$$\begin{aligned} \mathbf{z} &= f_{\theta}(\mathbf{x}), \\ \tilde{\mathbf{x}} &= g_{\phi}(\mathbf{z}). \end{aligned}$$

## Архитектура автоэнкодера (2)





**Функция потерь:** ошибка реконструкции (*reconstruction error*):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2 = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\mathbf{x}_i))\|_2^2.$$

**Альтернативная функция потерь:** бинарная кросс-энтропия для выхода в  $[0, 1]^D$ :

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^D (x_{ij} \log \tilde{x}_{ij} + (1 - x_{ij}) \log(1 - \tilde{x}_{ij})).$$

Ошибка реконструкции в PCA:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2 = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{B}\mathbf{B}^T \mathbf{x}_i\|_2^2.$$

Если:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_E \mathbf{x}, \quad g_{\boldsymbol{\phi}}(\mathbf{z}) = \mathbf{W}_D \mathbf{z},$$

то:

$$\mathbf{W}_E = \mathbf{B}^T, \quad \mathbf{W}_D = \mathbf{B},$$

и автоэнкодер эквивалентен PCA.



## Variational AutoEncoders (VAEs)

---

## Постановка задачи (1)

Дан неразмеченный датасет  $X = \{\mathbf{x}_i\}_{i=1}^N$ , где  $\mathbf{x}_i \in \mathbb{R}^D$ .

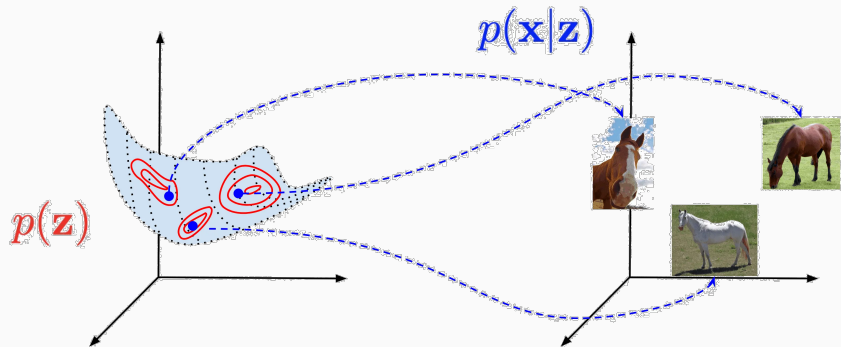
**Цель:**

- Обучить автоэнкодер так, чтобы его скрытое представление  $\mathbf{z}_i \in \mathbb{R}^M$  было распределено по заданному распределению.
- Задать латентное распределение:  $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{E})$ .

**Построение генеративной модели:**

- Способность генерировать объекты  $p(\mathbf{z})$ , близкие к объектам обучающей выборки  $X$ .

## Постановка задачи (2)



# Интуиция (1)

Идея:

- $f_{\theta}(\mathbf{x}) = \psi_{\mathbf{x}} = (\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}}^2)$  — параметры нормального распределения.
- $\mathbf{z} \sim p(\mathbf{z} \mid \psi_{\mathbf{x}}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}}^2)$ .
- $\tilde{\mathbf{x}} = g_{\phi}(\mathbf{z}) \sim p(\mathbf{x} \mid \mathbf{z})$ .

Параметры:

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{x}} &= [\mu_{\mathbf{x}1} \quad \mu_{\mathbf{x}2} \quad \dots \quad \mu_{\mathbf{x}M}] , \\ \boldsymbol{\sigma}_{\mathbf{x}}^2 &= \text{diag} \left( [\sigma_{\mathbf{x}1}^2 \quad \sigma_{\mathbf{x}2}^2 \quad \dots \quad \sigma_{\mathbf{x}M}^2] \right) .\end{aligned}$$

### Проблемы:

1. **Проблема 1:** модель стремится к  $\sigma_x^2 = 0$ , что превращает VAE в АЕ.
  - **Решение:** добавить регуляризационный член (какой?):
$$\mathcal{L}(\theta, \phi) = L_{\text{rec}} + \alpha L_{\text{reg}}.$$
2. **Проблема 2:** сэмплирование не дифференцируемо.



# Правдоподобие $p(\mathbf{x})$

Совместное распределение:

$$p_{\phi}(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z} \mid \phi).$$

Правдоподобие данных:

$$L(\phi) = \prod_{i=1}^N p_{\phi}(\mathbf{x}_i),$$

$$\log L(\phi) = \sum_{i=1}^N \log p_{\phi}(\mathbf{x}_i).$$

Цель: Максимизация правдоподобия:

$$p_{\phi}(\mathbf{x}) = \int_{\mathbf{z}} p_{\phi}(\mathbf{x}, \mathbf{z}) d\mathbf{z} \rightarrow \max_{\phi \in \Phi}.$$

# Аппроксимация $p(\mathbf{z} \mid \mathbf{x})$ (1)

Теорема Байеса:

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}.$$

Распределения:

- $p(\mathbf{x})$ : априорное распределение данных.
- $p(\mathbf{z} \mid \mathbf{x})$ : распределение энкодера.
- $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{E})$ : априорное распределение латентного пространства.
- $p(\mathbf{x} \mid \mathbf{z}) (= \mathcal{N}(\mathbf{x} \mid g_{\phi}(\mathbf{z}), c\mathbf{I}))$ : распределение декодера.

## Аппроксимация $p(\mathbf{z} \mid \mathbf{x})$ (2)

Проблема:  $p(\mathbf{z} \mid \mathbf{x})$  имеет сложную форму:

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{\mathcal{N}(\mathbf{x} \mid g_{\phi}(\mathbf{z}), c\mathbf{I})\mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{E})}{p(\mathbf{x})}.$$

Workaround: Аппроксимация через простое распределение  $q(\mathbf{z})$ :

$$p(\mathbf{z} \mid \mathbf{x}) \approx q(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}}^2).$$

# Вариационное приближение (1)

**Цель:** Максимизировать правдоподобие  $p(\mathbf{x})$ :

$$\begin{aligned}\log p(\mathbf{x}) &= \log \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int_{\mathbf{z}} p(\mathbf{x} | \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \\ &= \log \mathbb{E}_{q(\mathbf{z})} \left[ \frac{p(\mathbf{z}) p(\mathbf{x} | \mathbf{z})}{q(\mathbf{z})} \right].\end{aligned}$$

**Неравенства Йенсена:**

$$\begin{aligned}g(\mathbb{E}[\xi]) &\leq \mathbb{E}[g(\xi)], & g(x) &\text{— вогнутая функция,} \\ g(\mathbb{E}[\xi]) &\geq \mathbb{E}[g(\xi)], & g(x) &\text{— выпуклая функция.}\end{aligned}$$

## Вариационное приближение (2)

Применим к  $\log$ :

$$\begin{aligned}\log p(\mathbf{x}) &= \log \mathbb{E}_{q(\mathbf{z})} \left[ \frac{p(\mathbf{z})p(\mathbf{x} | \mathbf{z})}{q(\mathbf{z})} \right] \geq \mathbb{E}_{q(\mathbf{z})} \left[ \log \frac{p(\mathbf{z})p(\mathbf{x} | \mathbf{z})}{q(\mathbf{z})} \right] \\ &= \mathbb{E}_{q(\mathbf{z})} [\log p(\mathbf{x} | \mathbf{z})] - \mathbb{E}_{q(\mathbf{z})} \left[ \log \frac{q(\mathbf{z})}{p(\mathbf{z})} \right] \\ &= \mathbb{E}_{q(\mathbf{z})} [\log p(\mathbf{x} | \mathbf{z})] - \text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})),\end{aligned}$$

где KL — дивергенция Кульбака-Лейблера.

**Итог:** Нижняя граница на  $\log p(\mathbf{x})$  (Evidence Lower Bound, *ELBO*):

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z})} [\log p(\mathbf{x} | \mathbf{z})] - \text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})) \rightarrow \max_{q(\mathbf{z})}.$$

## Распределение $p(\mathbf{x} \mid \mathbf{z})$

$p(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mathbf{x} \mid g_\phi(\mathbf{z}), c\mathbf{I})$ ,  $c \neq 0$ , поэтому матрица ковариации невырождена.

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{z}) &= \frac{1}{(2\pi)^{D/2} |c\mathbf{I}|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - g_\phi(\mathbf{z}))^T (c\mathbf{I})^{-1} (\mathbf{x} - g_\phi(\mathbf{z})) \right) \\ &= \frac{1}{(2\pi)^{D/2} c^{D/2}} \exp \left( -\frac{1}{2c} \|\mathbf{x} - g_\phi(\mathbf{z})\|_2^2 \right) \end{aligned}$$

$$\begin{aligned} \log(p(\mathbf{x} \mid \mathbf{z})) &= -\frac{D}{2} \log(2\pi) - \frac{D}{2} \log(c) - \frac{1}{2c} \|\mathbf{x} - g_\phi(\mathbf{z})\|_2^2 \\ &= \text{const} - \frac{1}{2c} \|\mathbf{x} - g_\phi(\mathbf{z})\|_2^2. \end{aligned}$$

Оптимизационная задача:

$$\begin{aligned} & \mathbb{E}_{q(\mathbf{z})} [\log p(\mathbf{x} \mid \mathbf{z})] - \text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})) \\ &= -\mathbb{E}_{q(\mathbf{z})} \left[ \frac{1}{2c} \|\mathbf{x} - g_{\phi}(\mathbf{z})\|_2^2 \right] - \text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})) \rightarrow \max_{q(\mathbf{z})}. \end{aligned}$$

Эквивалентная формулировка:

$$\mathbb{E}_{q(\mathbf{z})} \left[ \frac{1}{2c} \|\mathbf{x} - g_{\phi}(\mathbf{z})\|_2^2 \right] + \text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})) \rightarrow \min_{q(\mathbf{z})}.$$

Получили то, чего и хотели:

$$\begin{aligned}L_{\text{rec}} &= \frac{1}{2c} \|\mathbf{x} - g\phi(\mathbf{z})\|_2^2, \\L_{\text{reg}} &= \text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})).\end{aligned}$$

**Итог:** Баланс между реконструкцией данных и отклонением от априорного распределения в латентном пространстве.



## Вычисление $L_{\text{reg}}(1)$

$$\begin{aligned} \text{KL}(q(\mathbf{z}) \parallel p(\mathbf{z})) &= \text{KL}(\mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}}^2) \parallel \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{E})) \\ &= \text{KL}\left(\prod_{i=1}^M \mathcal{N}(z_i \mid \mu_{\mathbf{x}i}, \sigma_{\mathbf{x}i}^2) \parallel \prod_{i=1}^M \mathcal{N}(z_i \mid 0, 1)\right) = \text{KL}\left(\prod_{i=1}^M q_i(\mathbf{z}_i) \parallel \prod_{i=1}^M p_i(\mathbf{z}_i)\right) \\ &= \int_{\mathbf{z}} \prod_{i=1}^M q_i(\mathbf{z}_i) \log \frac{\prod_{i=1}^M q_i(\mathbf{z}_i)}{\prod_{i=1}^M p_i(\mathbf{z}_i)} d\mathbf{z} = \int_{\mathbf{z}} \prod_{i=1}^M q_i(\mathbf{z}_i) \left(\sum_{i=1}^M \log \frac{q_i(\mathbf{z}_i)}{p_i(\mathbf{z}_i)}\right) d\mathbf{z} \\ &= \sum_{j=1}^M \int_{\mathbf{z}} \log \frac{q_j(\mathbf{z}_j)}{p_j(\mathbf{z}_j)} \prod_{i=1}^M q_i(\mathbf{z}_i) d\mathbf{z}_1 \dots d\mathbf{z}_M \\ &= \sum_{j=1}^M \left( \left( \int_{\mathbf{z}_j} \log \frac{q_j(\mathbf{z}_j)}{p_j(\mathbf{z}_j)} q_j(\mathbf{z}_j) d\mathbf{z}_j \right) \prod_{i \neq j} \int_{\mathbf{z}_i} q_i(\mathbf{z}_i) d\mathbf{z}_i \right) = \dots \end{aligned}$$

## Вычисление $L_{\text{reg}}$ (2)

$$= \sum_{j=1}^M \left( \int_{\mathbf{z}_j} \log \frac{q_j(\mathbf{z}_j)}{p_j(\mathbf{z}_j)} q_j(\mathbf{z}_j) d\mathbf{z}_j \right) = \sum_{j=1}^M \text{KL}(q_j(\mathbf{z}_j) \parallel p_j(\mathbf{z}_j))$$

Для каждой компоненты:

$$\begin{aligned} \text{KL}(q_j(\mathbf{z}_j) \parallel p_j(\mathbf{z}_j)) &= \int_{z_j} \mathcal{N}(z_j \mid \mu_{\mathbf{x}j}, \sigma_{\mathbf{x}j}^2) \log \frac{\mathcal{N}(z_j \mid \mu_{\mathbf{x}j}, \sigma_{\mathbf{x}j}^2)}{\mathcal{N}(z_j \mid 0, 1)} dz_j. \\ &= \mathbb{E}_{q_j(z_j)} [\log \mathcal{N}(z_j \mid \mu_{\mathbf{x}j}, \sigma_{\mathbf{x}j}^2) - \log \mathcal{N}(z_j \mid 0, 1)] . \end{aligned}$$

Распределение  $\mathcal{N}$ :

$$\begin{aligned} \log \mathcal{N}(z_j \mid \mu_{\mathbf{x}j}, \sigma_{\mathbf{x}j}^2) &= -\frac{1}{2} \log(2\pi\sigma_{\mathbf{x}j}^2) - \frac{1}{2\sigma_{\mathbf{x}j}^2} (z_j - \mu_{\mathbf{x}j})^2, \\ \log \mathcal{N}(z_j \mid 0, 1) &= -\frac{1}{2} \log(2\pi) - \frac{1}{2} z_j^2. \end{aligned}$$

## Вычисление $L_{\text{reg}}$ (3)

Разница логарифмов:

$$\log \mathcal{N}(z_j \mid \mu_{\mathbf{x}j}, \sigma_{\mathbf{x}j}^2) - \log \mathcal{N}(z_j \mid 0, 1) = -\frac{1}{2} \log(\sigma_{\mathbf{x}j}^2) - \frac{1}{2\sigma_{\mathbf{x}j}^2} (z_j - \mu_{\mathbf{x}j})^2 + \frac{1}{2} z_j^2.$$

Итоговая формула:

$$\begin{aligned} \text{KL}(q_j(\mathbf{z}_j) \parallel p_j(\mathbf{z}_j)) &= \mathbb{E}_{q_j(z_j)} \left[ -\frac{1}{2} \log(\sigma_{\mathbf{x}j}^2) - \frac{1}{2\sigma_{\mathbf{x}j}^2} (z_j - \mu_{\mathbf{x}j})^2 + \frac{1}{2} z_j^2 \right] \\ &= \mathbb{E}_{q_j(z_j)} \left[ -\left( \frac{1}{2} \log(\sigma_{\mathbf{x}j}^2) + \frac{\mu_{\mathbf{x}j}^2}{2\sigma_{\mathbf{x}j}^2} \right) + \left( \frac{1}{2} - \frac{1}{2\sigma_{\mathbf{x}j}^2} \right) z_j^2 + \frac{\mu_{\mathbf{x}j}}{\sigma_{\mathbf{x}j}^2} z_j \right] \\ &= -\left( \frac{1}{2} \log(\sigma_{\mathbf{x}j}^2) + \frac{\mu_{\mathbf{x}j}^2}{2\sigma_{\mathbf{x}j}^2} \right) + \left( \frac{1}{2} - \frac{1}{2\sigma_{\mathbf{x}j}^2} \right) \mathbb{E}_{q_j(z_j)} [z_j^2] + \frac{\mu_{\mathbf{x}j}}{\sigma_{\mathbf{x}j}^2} \mathbb{E}_{q_j(z_j)} [z_j] = \dots \end{aligned}$$

## Вычисление $L_{\text{reg}}$ (4)

Математические ожидания:

$$\mathbb{E}_{q_j(z_j)} [z_j] = \mu_{xj},$$

$$\mathbb{D}_{q_j(z_j)} [z_j] = \mathbb{E}_{q_j(z_j)} [z_j^2] - \mu_{xj}^2 = \sigma_{xj}^2 \quad \Rightarrow \quad \mathbb{E}_{q_j(z_j)} [z_j^2] = \sigma_{xj}^2 + \mu_{xj}^2.$$

Подстановка:

$$\begin{aligned} \dots &= -\frac{1}{2} \log(\sigma_{xj}^2) - \frac{\mu_{xj}^2}{2\sigma_{xj}^2} + \frac{1}{2} \left( 1 - \frac{1}{\sigma_{xj}^2} \right) (\sigma_{xj}^2 + \mu_{xj}^2) + \frac{\mu_{xj}}{\sigma_{xj}^2} \mu_{xj} \\ &= -\frac{1}{2} \log(\sigma_{xj}^2) - \frac{\mu_{xj}^2}{2\sigma_{xj}^2} + \frac{1}{2} \left( \sigma_{xj}^2 + \mu_{xj}^2 - 1 - \frac{\mu_{xj}^2}{\sigma_{xj}^2} \right) + \frac{\mu_{xj}^2}{\sigma_{xj}^2} \\ &= -\frac{1}{2} \log(\sigma_{xj}^2) + \frac{1}{2} \sigma_{xj}^2 + \frac{1}{2} \mu_{xj}^2 - \frac{1}{2} \\ &= \frac{1}{2} (\sigma_{xj}^2 + \mu_{xj}^2 - 1 - \log(\sigma_{xj}^2)) \end{aligned}$$

Итоговые формулы:

$$L_{\text{rec}} = \frac{1}{2c} \|\mathbf{x} - g_{\phi}(\mathbf{z})\|_2^2,$$

$$L_{\text{reg}} = \sum_{j=1}^M \frac{1}{2} (\sigma_{\mathbf{x}j}^2 + \mu_{\mathbf{x}j}^2 - 1 - \log(\sigma_{\mathbf{x}j}^2)) .$$

**Дополнение:** Для  $L_{\text{rec}}$  также можно использовать кросс-энтропию.

## Reparametrization trick

$$\mathbf{z}_j \sim \mathcal{N}(\mathbf{z}_j \mid \mu_{\mathbf{x}j}, \sigma_{\mathbf{x}j}^2) \Leftrightarrow \begin{cases} \epsilon_j \sim \mathcal{N}(0, 1), \\ \mathbf{z}_j = \mu_{\mathbf{x}j} + \sigma_{\mathbf{x}j} \epsilon_j. \end{cases}$$

Поскольку:

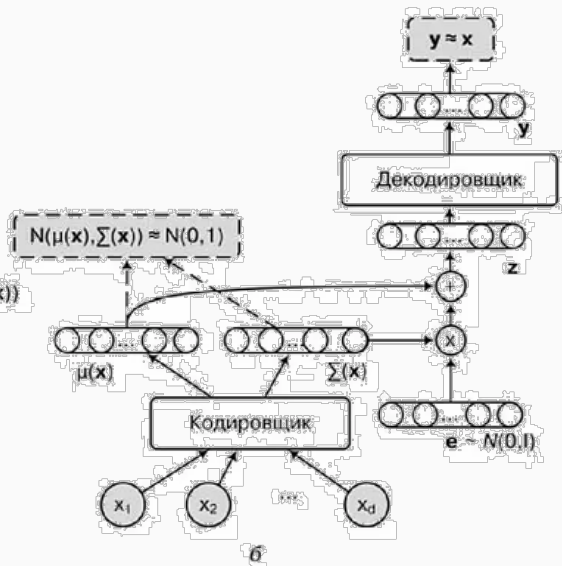
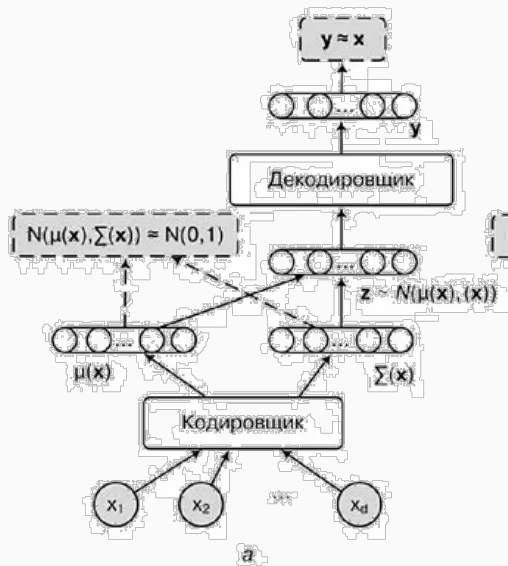
$$p_\epsilon(\epsilon) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\epsilon^2}{2}\right),$$

$$\epsilon = \frac{z - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}, \quad \sigma_{\mathbf{x}} > 0,$$

$$\left| \frac{d\epsilon}{dz} \right| = \frac{1}{\sigma_{\mathbf{x}}},$$

$$p_z(z) = p_\epsilon\left(\frac{z - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}\right) \left| \frac{d\epsilon}{dz} \right| = \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}} \exp\left(-\frac{(z - \mu_{\mathbf{x}})^2}{2\sigma_{\mathbf{x}}^2}\right).$$

# Архитектура VAE



**Проблема:**  $\forall i \quad 0 < \sigma_{xi}^2 \ll 1$  — маленькие значения могут приводить к ошибкам численных расчетов.

**Решение:** Использовать логарифмированное значение  $\sigma_{xi}^2$ :

$$\log \sigma_{xi}^2 \in \mathbb{R}^M.$$

**Репараметризация:**

$$\begin{aligned}\epsilon_j &\sim \mathcal{N}(0, 1), \\ \mathbf{z}_j &= \mu_{xj} + \exp\left(\frac{1}{2} \log \sigma_{xi}^2\right) \epsilon_j.\end{aligned}$$





## Заключение

---

# Сравнительный анализ методов: особенности применения (1)

## Ключевые отличия:

- **РСА:** базовый линейный метод, идеален для анализа небольших и линейных зависимостей. Часто используется для визуализации и как отправная точка в анализе данных.
- **КРСА:** позволяет работать с нелинейной структурой данных, но требует осторожного выбора ядерной функции и гиперпараметров. Идеален для задач распознавания образов и биоинформатики.
- **АЕ:** предоставляет большую гибкость благодаря нейронным сетям. Находит применение в обработке данных и сложных задачах анализа.

## Сравнительный анализ методов: особенности применения (2)

- **VAE:** расширяет автоэнкодеры за счет вероятностной модели, идеально подходит для задач генерации данных и анализа латентных переменных.

### Дополнительно:

- Все методы обладают уникальными преимуществами и ограничениями, что делает их подходящими для разных классов задач.
- Выбор метода зависит от структуры данных, целей анализа и доступных вычислительных ресурсов.

## Заключение (1)

**Современные вызовы:** Работа с высокоразмерными данными требует гибких и мощных инструментов.

**Основные итоги:**

- **Линейные методы** (например, PCA) остаются незаменимыми благодаря своей простоте и эффективности.
- **Нелинейные подходы**, такие как KPCA, открывают возможности работы с более сложными структурами данных.
- **Автоэнкодеры** обеспечивают исключительную гибкость для задач генерации данных и анализа скрытых зависимостей.

### Рекомендации по выбору:

- Для интерпретируемости и быстродействия – линейные методы.
- Для работы с нелинейными структурами – Kernel PCA.
- Для генерации данных – автоэнкодеры и их вариации.

**Вывод:** Методы понижения размерности предоставляют исследователям мощный арсенал для анализа данных, повышая информативность, эффективность и удобство визуализации.