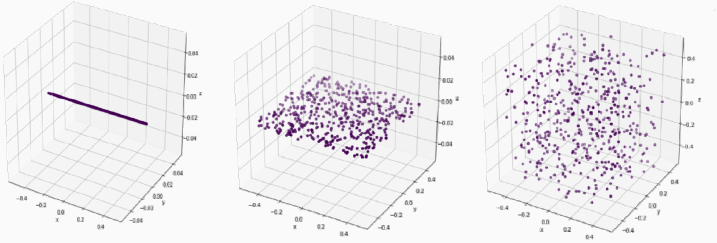
Методы понижения размерности данных

Поглазов Никита
2024

Введение

Введение

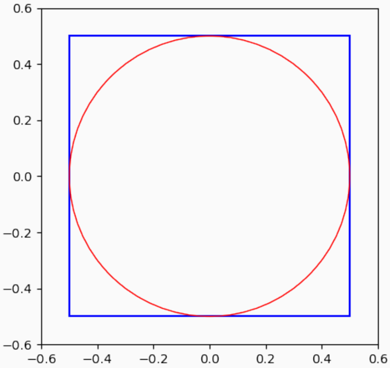
Проклятие размерности: данные высокой размерности сложны для анализа, требуют много вычислительных ресурсов и часто содержат шум.



Мотивация

Что такое "проклятие размерности"?

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

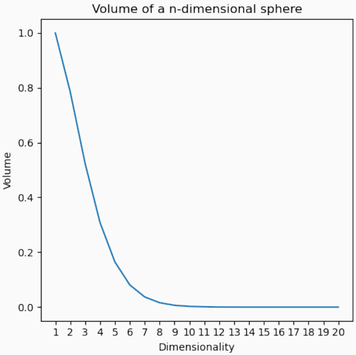


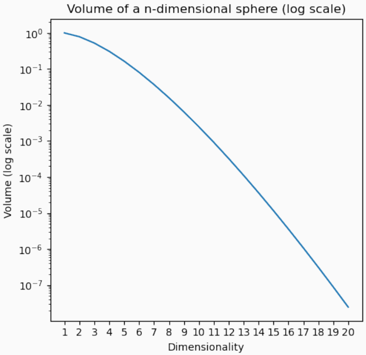
Гиперсфера и гиперкуб

- 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 $\sqrt{n}$.

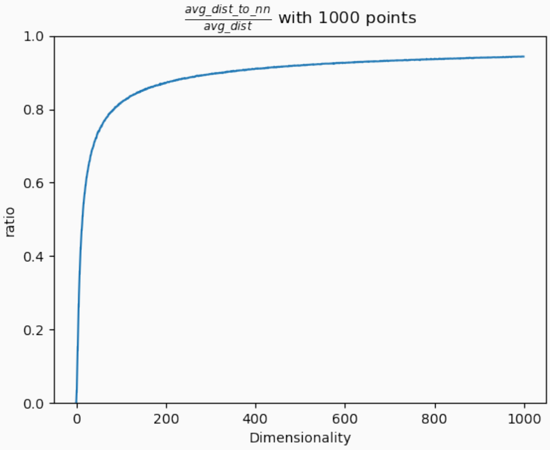




Влияние на метрические модели (1)

- Манхэттенское расстояние:
$$d(x^{(i)}, x^{(j)}) = \sum_{k=1}^{n} |x_k^{(i)} - x_k^{(j)}|$$
- Средние расстояния между точками становятся близкими:
$$\lim_{n\to\infty} \frac{d(x^{(i)}, x^{(j)})}{n} = \mu$$
- 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:
$$\mathrm{Cov}(X) = \frac{1}{k-1} X^T X$$
- 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

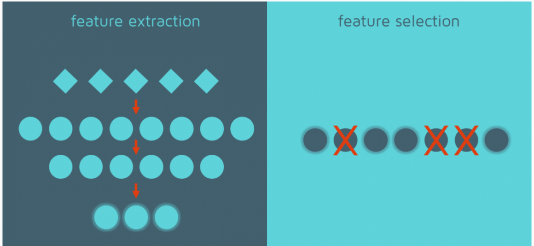
Два подхода к понижению размерности

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.

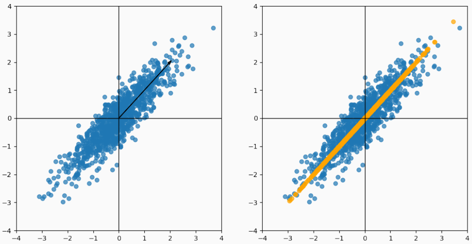


Principal Component Analysis (PCA)

Цель: Сохранить максимальную дисперсию данных.

Пример применения:

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



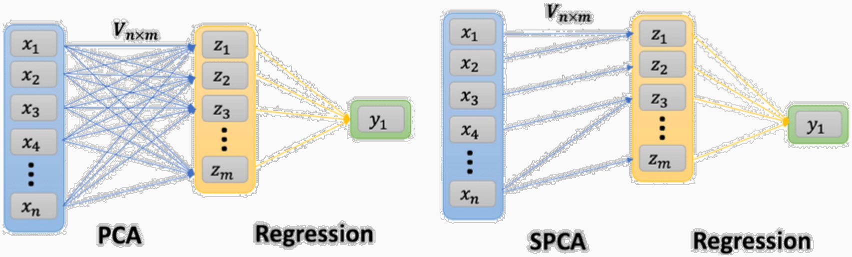
Sparse PCA (SPCA)

Отличие от PCA:

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

Применение:

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

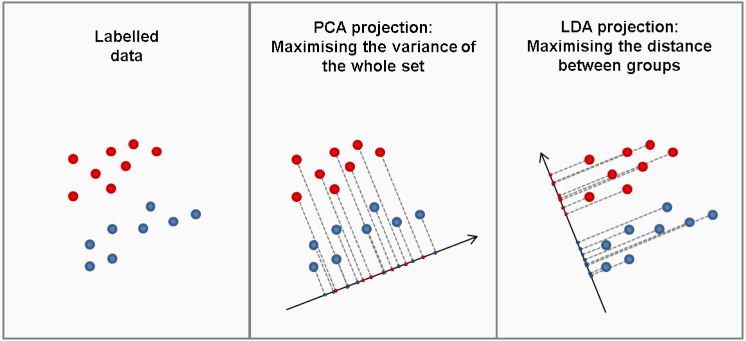


Linear Discriminant Analysis (LDA)

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.

Kernel PCA (KPCA)

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

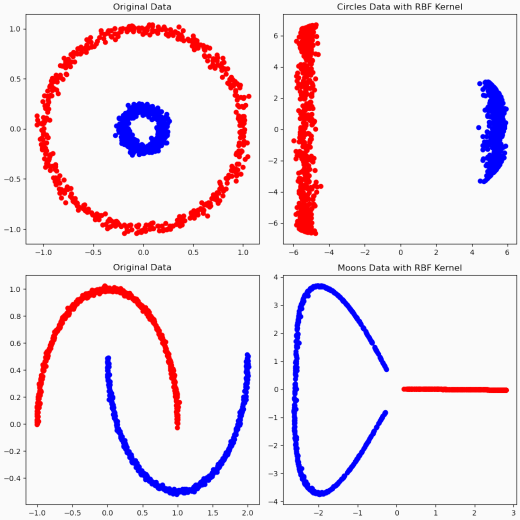
Пример ядерной функции (гауссовское ядро):

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

Пример применения:

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

Kernel PCA (KPCA)



t-SNE

Алгоритм визуализации (!) данных высокой размерности.

Цель: Локальное сохранение расстояний между точками.

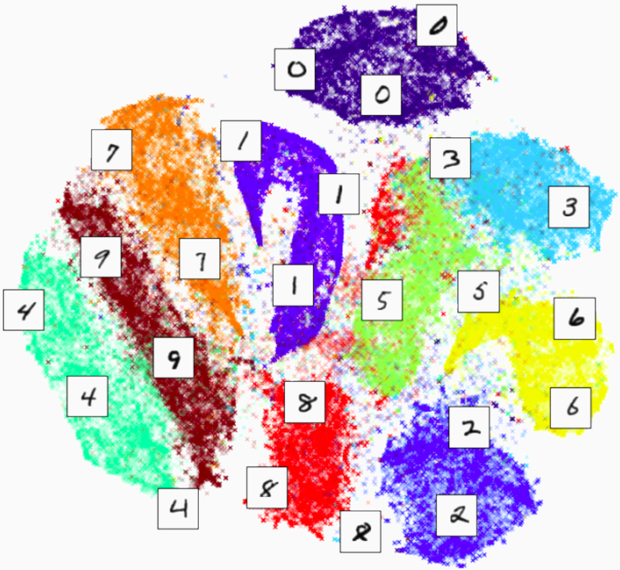
Основная идея:

- 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.

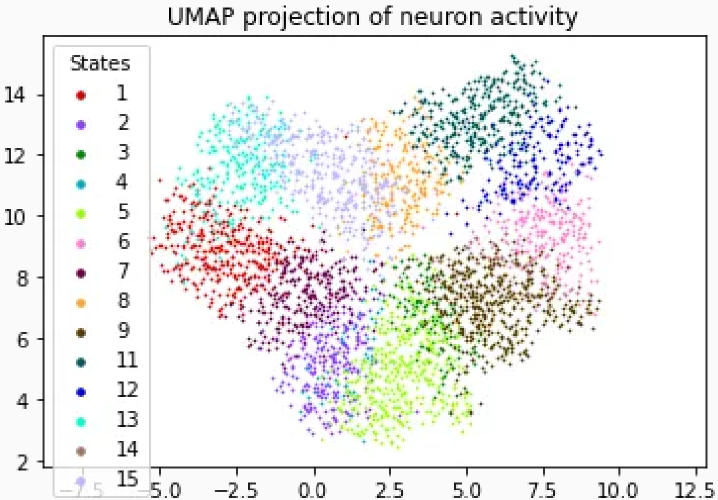
Основные этапы метода:

- 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.

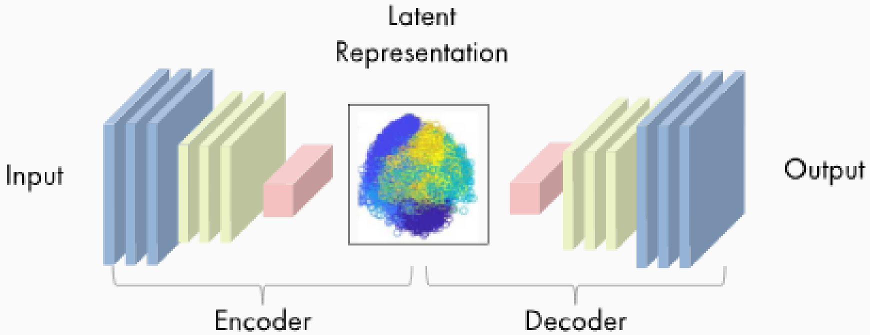
Основная структура:

- 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.

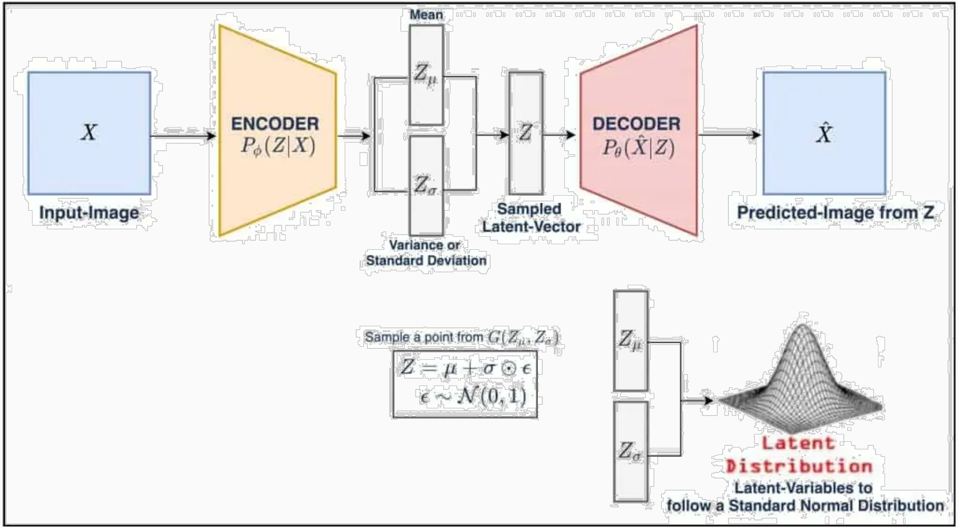
Основная идея:

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

Матрица ковариации данных:

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

Переход в новое пространство меньшей размерности (сжатие):

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

Постановка задачи (2)

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

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

Восстановление данных:

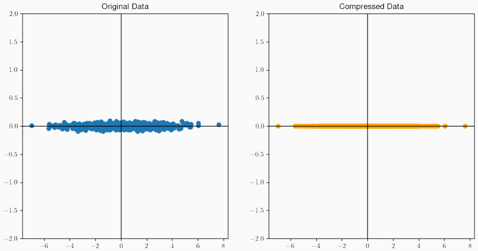
$$\tilde{\mathbf{x}}_i = \mathbf{B}\boldsymbol{z}_i.$$

Пример: 2D $\rightarrow$ 1D

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

Шаги:

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



Нахождение направления максимальной дисперсии (1)

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

Дисперсия вдоль первой координаты в новом пространстве:

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

Нахождение направления максимальной дисперсии (2)

Задача условной оптимизации:

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

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

Частные производные по $\boldsymbol{b}_1$ и λ:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}_1} = 2\boldsymbol{\Sigma} \boldsymbol{b}_1 - 2\lambda_1 \boldsymbol{b}_1 = 0,$$
$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = -\boldsymbol{b}_1^T \boldsymbol{b}_1 + 1 = 0.$$

Собственные векторы и значения

Получаем:

$$\boldsymbol{\Sigma} \boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1,$$
$$V_1 = \lambda_1.$$

Теперь можем переписать дисперсию V_1 как:

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

Интерпретация:

- $\boldsymbol{b}_1$: первое направление главной компоненты.
- λ_1: дисперсия вдоль направления $\boldsymbol{b}_1$.

Остальные компоненты (1)

Для m-й компоненты:

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

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

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

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

Остальные компоненты (2)

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

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

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

$\boldsymbol{\Sigma}$ симметрична, поэтому
$\boldsymbol{b}_j^T \boldsymbol{\Sigma} \boldsymbol{b}_m = \left\langle (\boldsymbol{b}_j^T \boldsymbol{\Sigma})^T, \boldsymbol{b}_m \right\rangle = \langle \boldsymbol{\Sigma} \boldsymbol{b}_j, \boldsymbol{b}_m \rangle = \langle \lambda_j \boldsymbol{b}_j, \boldsymbol{b}_m \rangle = \lambda_j \langle \boldsymbol{b}_j, \boldsymbol{b}_m \rangle = 0.$
Тогда $\mu_j = 0$. и, аналогично, $\forall j < m \;\; \mu_j = 0$

Остальные компоненты (3)

Таким образом:

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

Вновь, $\boldsymbol{b}_m$ - собственный вектор матрицы ковариации $\boldsymbol{\Sigma}$, а λ_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: $\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$.

Практические аспекты реализации

Формулы:

$$\boldsymbol{Z} = \mathbf{B}^T \boldsymbol{X},$$
$$\tilde{\boldsymbol{X}} = \mathbf{B}\boldsymbol{Z},$$
$$\boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T.$$

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

Практика

Kernel PCA (*KPCA*)

Постановка задачи (1)

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

Задано:

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

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 ϕ можно построить ядро k - положительно определенная функция.
- **Теорема Moore-Aronszajn:** По положительно определённому ядру k можно построить ϕ и пространство $\mathbb{H}$.
- Матрица Грама $\mathbf{K} \in \mathbb{R}^{N \times N}$:
$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

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

Наивный подход

Шаги:

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

Проблемы:

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

Kernel Trick (1)

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

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

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

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

Kernel Trick (2)

Подставим $\boldsymbol{\Sigma}$:

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

Представление компонент:

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

Kernel Trick (3)

Подставим это в уравнение для ω_p:

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\phi(\boldsymbol{x}_i)\langle\phi(\boldsymbol{x}_i),\sum_{j=1}^{N}\alpha_{p,j}\phi(\boldsymbol{x}_j)\rangle_{\mathbb{H}} = \lambda_p\sum_{i=1}^{N}\alpha_{p,i}\phi(\boldsymbol{x}_i),\\
&\frac{1}{N}\sum_{i=1}^{N}\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T\sum_{j=1}^{N}\phi(\boldsymbol{x}_j)\alpha_{p,j} = \lambda_p\sum_{j=1}^{N}\alpha_{p,j}\phi(\boldsymbol{x}_j),\\
&\frac{1}{N}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\alpha}_p = \lambda_p\boldsymbol{\Phi}^T\boldsymbol{\alpha}_p,\\
&\boldsymbol{\Phi}^T(\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\alpha}_p - N\lambda_p\boldsymbol{\alpha}_p) = 0\\
&\mathbf{K}\boldsymbol{\alpha}_p = N\lambda_p\boldsymbol{\alpha}_p, \qquad \mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T, \qquad K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j).
\end{aligned}
$$

Проекции на главные компоненты

Проекции на главные компоненты вычисляются **даже без знания** ϕ:

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

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

Центрирование образов (1)

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

Коррекция:

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

Центрирование образов (2)

Обновление ядра:

$$\tilde{k}(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x}, \boldsymbol{y}) - \frac{1}{N}\sum_{i=1}^{N}\left(k(\boldsymbol{x}, \boldsymbol{x}_i) - k(\boldsymbol{x}_i, \boldsymbol{y})\right) + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

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.

Детали реализации

Наиболее популярное ядро: Гауссово (RBF):

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

Альтернативные ядра:

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

Практика

AutoEncoders (*AEs*)

Постановка задачи

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

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

Подход: Использование нелинейных преобразований, реализованных нейронной сетью.

Архитектура автоэнкодера (1)

Encoder: Нелинейное отображение $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^M$:

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

где:

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

Decoder: Восстановление $g_{\boldsymbol{\phi}} : \mathbb{R}^M \to \mathbb{R}^D$:

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

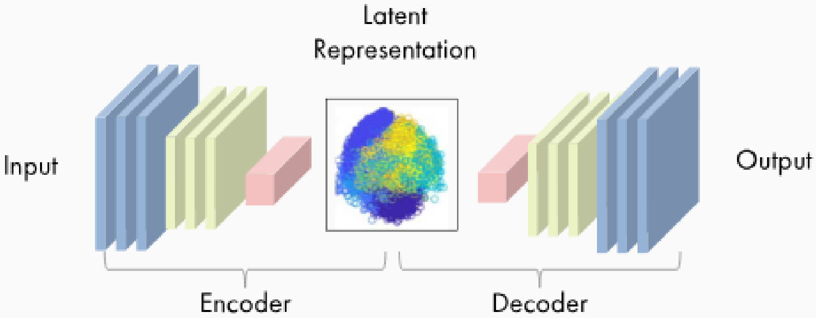
где:

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

Полная архитектура:

$$\boldsymbol{z} = f_{\boldsymbol{\theta}}(\boldsymbol{x}),$$
$$\tilde{\mathbf{x}} = g_{\boldsymbol{\phi}}(\boldsymbol{z}).$$

Архитектура автоэнкодера (2)



Обучение автоэнкодера

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

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

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

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

Аналогия с PCA

Ошибка реконструкции в PCA:

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

Если:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{W}_{\mathsf{E}}\boldsymbol{x}, \quad g_{\boldsymbol{\phi}}(\boldsymbol{z}) = \mathbf{W}_{\mathsf{D}}\boldsymbol{z},$$

то:

$$\mathbf{W}_{\mathsf{E}} = \mathbf{B}^T, \quad \mathbf{W}_{\mathsf{D}} = \mathbf{B},$$

и автоэнкодер эквивалентен PCA.

Практика

Variational AutoEncoders (*VAEs*)

Постановка задачи (1)

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

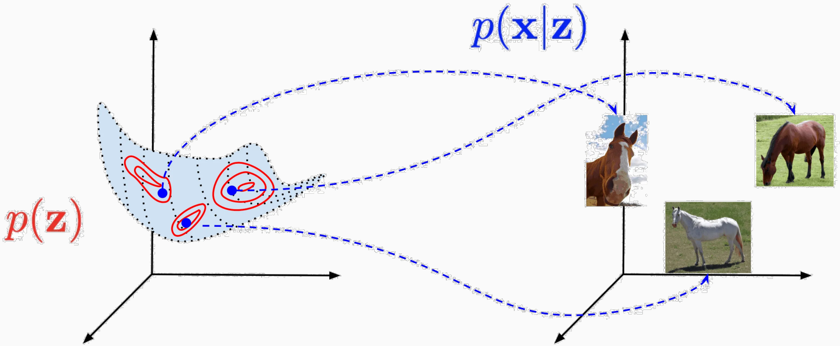
Цель:

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

Построение генеративной модели:

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

Постановка задачи (2)



Интуиция (1)

Идея:

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

Параметры:

$$\boldsymbol{\mu}_{\mathbf{x}} = \begin{bmatrix} \mu_{\boldsymbol{x}1} & \mu_{\boldsymbol{x}2} & \dots & \mu_{\boldsymbol{x}M} \end{bmatrix},$$
$$\boldsymbol{\sigma}^2_{\mathbf{x}} = \operatorname{diag}\left(\begin{bmatrix} \sigma^2_{\boldsymbol{x}1} & \sigma^2_{\boldsymbol{x}2} & \dots & \sigma^2_{\boldsymbol{x}M} \end{bmatrix}\right).$$

Интуиция (2)

Проблемы:

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

Правдоподобие $p(\boldsymbol{x})$

Совместное распределение:

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

Правдоподобие данных:

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

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

Цель: Максимизация правдоподобия:

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

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

Теорема Байеса:

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

Распределения:

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

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

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

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

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

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

Вариационное приближение (1)

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

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

Неравенства Йенсена:

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

Вариационное приближение (2)

Применим к $\log$:

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

где KL — дивергенция Кульбака-Лейблера.

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

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{z})}\left[\log p(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathrm{KL}(q(\boldsymbol{z}) \parallel p(\boldsymbol{z})) \to \max_{q(\boldsymbol{z})}.$$

Распределение $p(\boldsymbol{x} \mid \boldsymbol{z})$

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

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

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

Итоговая оптимизация

Оптимизационная задача:

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

Эквивалентная формулировка:

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

Вспомним интуицию

Получили то, чего и хотели:

$$
\begin{aligned}
\mathsf{L}_{\mathsf{rec}} &= \frac{1}{2c}\|\boldsymbol{x} - g_{\boldsymbol{\phi}}(\boldsymbol{z})\|_2^2, \\
\mathsf{L}_{\mathsf{reg}} &= \mathsf{KL}(q(\boldsymbol{z}) \parallel p(\boldsymbol{z})).
\end{aligned}
$$

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

Вычисление L_{reg} (1)

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

Вычисление L_{reg} (2)

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

Для каждой компоненты:

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

Распределение $\mathcal{N}$:

$$\log \mathcal{N}(z_j \mid \mu_{\mathbf{x}j}, \sigma^2_{\mathbf{x}j}) = -\frac{1}{2} \log(2\pi\sigma^2_{\mathbf{x}j}) - \frac{1}{2\sigma^2_{\mathbf{x}j}} (z_j - \mu_{\mathbf{x}j})^2,$$

$$\log \mathcal{N}(z_j \mid 0, 1) = -\frac{1}{2} \log(2\pi) - \frac{1}{2} z_j^2.$$

Вычисление $\mathrm{L}_{\mathrm{reg}}$ (3)

Разница логарифмов:

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

Итоговая формула:

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

Вычисление L_{reg} (4)

Математические ожидания:

$$\mathbb{E}_{q_j(z_j)}\left[z_j\right] = \mu_{\mathbf{x}j},$$
$$\mathbb{D}_{q_j(z_j)}\left[z_j\right] = \mathbb{E}_{q_j(z_j)}\left[z_j^2\right] - \mu_{\mathbf{x}j}^2 = \sigma_{\mathbf{x}j}^2 \quad \Rightarrow \quad \mathbb{E}_{q_j(z_j)}\left[z_j^2\right] = \sigma_{\mathbf{x}j}^2 + \mu_{\mathbf{x}j}^2.$$

Подстановка:

$$\begin{aligned}
\ldots &= -\frac{1}{2}\log(\sigma_{\mathbf{x}j}^2) - \frac{\mu_{\mathbf{x}j}^2}{2\sigma_{\mathbf{x}j}^2} + \frac{1}{2}\left(1 - \frac{1}{\sigma_{\mathbf{x}j}^2}\right)(\sigma_{\mathbf{x}j}^2 + \mu_{\mathbf{x}j}^2) + \frac{\mu_{\mathbf{x}j}}{\sigma_{\mathbf{x}j}^2}\mu_{\mathbf{x}j} \\
&= -\frac{1}{2}\log(\sigma_{\mathbf{x}j}^2) - \frac{\mu_{\mathbf{x}j}^2}{2\sigma_{\mathbf{x}j}^2} + \frac{1}{2}\left(\sigma_{\mathbf{x}j}^2 + \mu_{\mathbf{x}j}^2 - 1 - \frac{\mu_{\mathbf{x}j}^2}{\sigma_{\mathbf{x}j}^2}\right) + \frac{\mu_{\mathbf{x}j}^2}{\sigma_{\mathbf{x}j}^2} \\
&= -\frac{1}{2}\log(\sigma_{\mathbf{x}j}^2) + \frac{1}{2}\sigma_{\mathbf{x}j}^2 + \frac{1}{2}\mu_{\mathbf{x}j}^2 - \frac{1}{2} \\
&= \frac{1}{2}\left(\sigma_{\mathbf{x}j}^2 + \mu_{\mathbf{x}j}^2 - 1 - \log(\sigma_{\mathbf{x}j}^2)\right)
\end{aligned}$$

Функционал потерь

Итоговые формулы:

$$\mathsf{L}_{\mathsf{rec}} = \frac{1}{2c}\|\boldsymbol{x} - g_{\boldsymbol{\phi}}(\boldsymbol{z})\|_2^2,$$

$$\mathsf{L}_{\mathsf{reg}} = \sum_{j=1}^{M} \frac{1}{2}\left(\sigma_{\mathbf{x}j}^2 + \mu_{\mathbf{x}j}^2 - 1 - \log(\sigma_{\mathbf{x}j}^2)\right).$$

Дополнение: Для $\mathsf{L}_{\mathsf{rec}}$ также можно использовать кросс-энтропию.

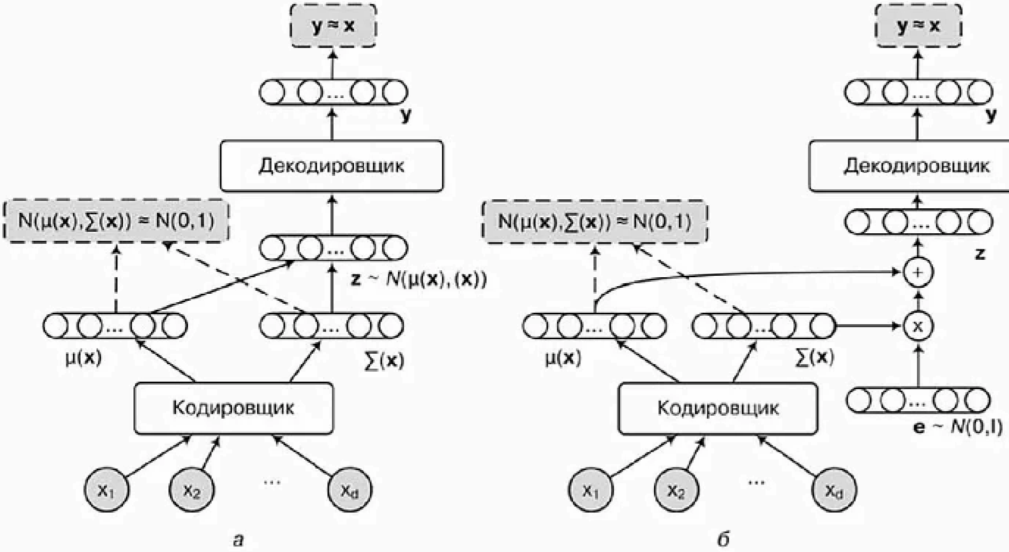
Reparametrization trick

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

Поскольку:

$$\begin{aligned} p_\varepsilon(\varepsilon) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\varepsilon^2}{2}\right), \\ \varepsilon &= \frac{z - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}, \quad \sigma_{\mathbf{x}} > 0, \\ \left|\frac{d\varepsilon}{dz}\right| &= \frac{1}{\sigma_{\mathbf{x}}}, \\ p_z(z) &= p_\varepsilon\left(\frac{z - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}\right)\left|\frac{d\varepsilon}{dz}\right| = \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}} \exp\left(-\frac{(z - \mu_{\mathbf{x}})^2}{2\sigma^2_{\mathbf{x}}}\right). \end{aligned}$$

Архитектура VAE



Проблема: $\forall i \quad 0 < \boldsymbol{\sigma}^2_{\boldsymbol{x}\boldsymbol{i}} \ll 1$ — маленькие значения могут приводить к ошибкам численных расчетов.

Решение: Использовать логарифмированное значение $\boldsymbol{\sigma}^2_{\boldsymbol{x}\boldsymbol{i}}$:

$$\log \boldsymbol{\sigma}^2_{\boldsymbol{x}\boldsymbol{i}} \in \mathbb{R}^M.$$

Репараметризация:

$$\boldsymbol{\varepsilon}_j \sim \boldsymbol{\mathcal{N}}(0, 1),$$

$$\boldsymbol{z}_j = \mu_{\mathbf{x}j} + \exp\left(\frac{1}{2}\log \boldsymbol{\sigma}^2_{\boldsymbol{x}\boldsymbol{i}}\right)\boldsymbol{\varepsilon}_j.$$

Практика

Заключение

Сравнительный анализ методов: особенности применения (1)

Ключевые отличия:

- **PCA:** базовый линейный метод, идеален для анализа небольших и линейных зависимостей. Часто используется для визуализации и как отправная точка в анализе данных.
- **KPCA:** позволяет работать с нелинейной структурой данных, но требует осторожного выбора ядерной функции и гиперпараметров. Идеален для задач распознавания образов и биоинформатики.
- **AE:** 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.

Заключение (2)

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.