Введение в машинное обучение

Лекция: Кластеризация и снижение размерности (SVD, PCA)

Лазар В. И. и Козлова Е. Р.

11 ноября 2025 г.

План занятия

1. Что такое «обучение без учителя».
2. **Кластеризация**: идеи и алгоритмы
   - K-Means
   - DBSCAN
   - Иерархическая (агломеративная) кластеризация
3. **Снижение размерности**:
   - SVD — сингулярное разложение
   - PCA — метод главных компонент
4. Практические советы, мини-упражнения, ответы на вопросы.

Обучение без учителя (интуиция)

- Даны только объекты (точки, картинки, тексты), *без меток ответов*.
- Хотим: **найти структуру** в данных.
- Две базовые задачи:
  1. **Кластеризация** — разбить объекты на «похожие группы».
  2. **Снижение размерности** — «упростить» данные, сохранив самое важное.

Кластеризация: K-Means — идея

- Выбираем число групп K.
- Каждая группа описывается центром (**центроидом**).
- Алгоритм стремится «поставить» центры так, чтобы точки были ближе к своим центрам, чем к чужим.
- Мера близости — обычно евклидово расстояние.

K-Means — пошаговый алгоритм

1. **Инициализация**: выбрать K начальных центров (лучше k-means++).
2. **Шаг присвоения**: каждую точку отнести к ближайшему центру.
3. **Шаг пересчёта**: для каждого кластера посчитать новый центр как среднее своих точек.
4. Повторять шаги 2–3, пока центры почти не меняются (**сходимость**).

$$\text{Целевая функция: } \min_{\text{кластеры}} \sum_{i=1}^{n} \left\| x_i - \mu_{c(i)} \right\|^2$$

где x_i — точка, $\mu_{c(i)}$ — центр её кластера.

K-Means — плюсы и минусы

Плюсы

- Простой и быстрый.
- Понятная геометрическая идея.
- Хорош для «круглых» кластеров схожего размера.

Минусы

- Нужно знать K заранее.
- Плохо с «некруглыми» формами и разной плотностью.
- Чувствителен к масштабам признаков и выбросам.

Практика: перед K-Means часто делают стандартизацию признаков.

Кластеризация: DBSCAN — идея

- **Density-Based Spatial Clustering of Applications with Noise**.
- Вместо «круглых» групп ищем **области высокой плотности**.
- Параметры:
  - ε (eps) — «радиус соседства»;
  - min_samples — сколько соседей нужно, чтобы точка считалась «ядром».
- Типы точек: **ядро**, **пограничная**, **шум**.
- Число кластеров получается автоматически.

DBSCAN — плюсы и минусы

Плюсы

- Находит *кластеры произвольной формы*.
- Не требует знать число кластеров.
- Устойчив к выбросам (помечает как шум).

Минусы

- Нужно подбирать ε и min_samples.
- Трудно при сильно неоднородной плотности.
- Чувствителен к масштабу признаков.

Иерархическая (агломеративная) кластеризация — идея

- Стартуем: каждый объект — свой кластер.
- На каждом шаге **сливаем** две наиболее близкие группы.
- Выбор «близости» групп — это **связь (linkage)**:
  - **single** — ближайшие точки;
  - **complete** — самые дальние точки;
  - **average** — среднее расстояние;
  - **Ward** — минимизируем рост дисперсии.
- Получаем **дендрограмму**; «отрезаем» на нужной высоте — получаем кластеры.

Иерархическая кластеризация — плюсы и минусы

Плюсы

- Дает *иерархию* — разные уровни детализации.
- Не обязательно заранее задавать K.
- Гибкость выбора метрики и связей.

Минусы

- Память/время $\sim n^2$ — трудно для очень больших данных.
- Ранние «неудачные» слияния уже не откатить.

Коротко: что выбрать?

	K-Means	**DBSCAN**	**Иерархическая**
Форма кластеров	«Круглые», схожие по размеру	Любые формы, плотностные	Любые, зависит от метрики/связи
Заранее K?	Да	Нет	Необязательно (режем дендрограмму)
Выбросы	Чувствителен	Устойчив (шум = -1)	Зависит от метрики
Гиперпараметры	K, инициализация	ε, min_samples	Связь (linkage), «высота среза»
Масштаб признаков	Важно	Важно	Важно
Сложность	Быстрый	Средний	$O(n^2)$ по памяти/времени

Снижение размерности — зачем?

- Упростить: оставить «главное», убрать шум/избыточность.
- **Визуализация**: проецировать высокие измерения на 2D/3D.
- **Скорость**: меньше признаков — быстрее обучение.
- **Сжатие**: хранить данные компактнее.

SVD — сингулярное разложение (идея)

- Для матрицы $X \in \mathbb{R}^{n \times d}$:
$$X = U \Sigma V^{\top},$$
где $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{d \times d}$ — **ортогональные** матрицы, Σ — диагональная с неотрицательными **сингулярными значениями** $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$.
- Геометрия: V — «направления» в признаковом пространстве, Σ — «силы» по этим направлениям, U — координаты точек в новых осях.
- Если оставить только первые k наибольших σ, получаем **лучшую аппроксимацию** ранга k (в смысле наименьших квадратов).

PCA — метод главных компонент (интуиция)

- Ищем новые оси, вдоль которых **разброс данных (дисперсия)** максимален.
- 1-я главная компонента — направление наибольшей дисперсии, 2-я — следующей и т.д.
- Проекция на первые k компонент сохраняет «больше всего информации» в смысле дисперсии.

PCA — пошаговый алгоритм

1. **Центрирование**: вычесть среднее из каждого признака.
2. (Часто) **Стандартизация**: привести признаки к одному масштабу.
3. Посчитать ковариационную матрицу $C = \frac{1}{n-1} X^\top X$ (для центрированного X).
4. Найти собственные векторы/значения C; отсортировать по убыванию.
5. Взять первые k векторов, спроецировать: $Z = XW_k$.

Связь с SVD: если $X = U\Sigma V^\top$ и X центрирована, то столбцы V — направления PCA, а $\Sigma^2/(n-1)$ — дисперсии компонент.

Практические советы

- **Масштабируйте** признаки (StandardScaler), особенно перед K-Means/PCA.
- Для K-Means подберите K: **локоть** (elbow) или **силуэт**.
- Для DBSCAN подберите ε по графику k-ближайших расстояний.
- В PCA смотрите на **долю объяснённой дисперсии**; часто берут 90–95%.
- Удаляйте **явные выбросы** или используйте устойчивые алгоритмы (DBSCAN).