ИНТЕЛЛЕКТУАЛЬНОЕ МОДЕЛИРОВАНИЕ

Образовательная программа «6B06103 – Компьютерная инженерия»

Лекция 12

Валидация моделей: кросс-валидация, регуляризация, подбор гиперпараметров

Неделя 12 · Tan, Steinbach, Kumar: 4.4–4.5, 5.6.3

Цель и план лекции

Цель: объективно оценивать, сравнивать и улучшать модели, избегая переобучения и утечки данных

1. Ошибка обобщения, переобучение, смещение и разброс
2. Отложенная выборка, кросс-валидация, бутстреп
3. Регуляризация: Ridge, Lasso, Elastic Net
4. Гиперпараметры: сетка, случайный поиск
5. Схема разделения данных, утечка, важность признаков

Ошибка обучения и ошибка обобщения

Ошибка обучения

Доля ошибок на обучающих объектах

Ошибка обобщения

Ожидаемая ошибка на новых, не виденных объектах

Главная цель — **ошибка обобщения**. Модель никогда не оценивают на тех же данных, на которых она обучалась.

Недообучение и переобучение

	Недообучение	Хорошая модель	Переобучение
Ошибка обучения	Высокая	Низкая	Очень низкая
Ошибка на тесте	Высокая	Низкая, близка к обучению	Высокая
Сложность	Мала	По данным	Велика
Что делать	Усложнить, добавить признаки	—	Упростить, регуляризовать, больше данных

С ростом дерева ошибка на тесте сначала падает, затем растёт

Причины переобучения

Шум

Ошибочные метки: летучая мышь и кит как «не млекопитающие»

Мало данных

Решения по случайным совпадениям (дерево по 10 заёмщикам)

Множество сравнений

Среди тысяч разбиений одно случайно подойдёт

Бритва Оккама: при равной ошибке обобщения выбираем более простую модель

Смещение и разброс

$$\text{Ошибка} = \text{смещение} + \text{разброс} + \text{неустранимый шум}$$

- **Смещение** — систематическая ошибка слишком простых предположений
- **Разброс** — чувствительность к случайностям обучающей выборки
- Простая модель: большое смещение → недообучение
- Сложная модель: большой разброс → переобучение

Задача — компромисс: модель оптимальной сложности

K-кратная кросс-валидация

Шаг 1	тест	обучение	обучение	обучение	обучение
Шаг 2	обучение	тест	обучение	обучение	обучение
…			…		
Шаг 5	обучение	обучение	обучение	обучение	тест

- $\text{Error}_{\text{CV}} = \frac{1}{k}\sum_{i=1}^{k} \text{Error}_i$; обычно $k = 5$ или 10
- Каждый объект ровно один раз в тесте; std по блокам — устойчивость

Отложенная выборка и виды кросс-валидации

- **Отложенная выборка** (70/30, 80/20): меньше данных на обучение, оценка зависит от случая
- **Случайное разбиение** k раз: объекты попадают в тест неравномерно
- **Stratified K-Fold**: сохраняет пропорции классов
- **LOOCV** ($k = N$): N обучений, большой разброс оценки
- **Group K-Fold**: группа целиком в одной части
- **TimeSeriesSplit**: обучение на прошлом, проверка на будущем

Бутстреп

63,2 %

различных объектов попадает в выборку

- N объектов выбираются **с возвращением**
- $1 - \left(1 - \frac{1}{N}\right)^N \to 1 - e^{-1} \approx 0{,}632$
- Не попавшие объекты — тестовая выборка
- Повтор b раз

.632-бутстреп: $\text{acc} = \frac{1}{b}\sum_{i=1}^{b}(0{,}632 \cdot \varepsilon_i + 0{,}368 \cdot \text{acc}_s)$

Какой метод оценки выбрать

Метод	Когда применять
Отложенная выборка	Много данных, быстрая оценка
K-Fold	Общий случай
Stratified K-Fold	Классификация, несбалансированные классы
LOOCV	Очень мало данных
Group K-Fold	Сгруппированные данные (пациенты, клиенты)
TimeSeriesSplit	Временные ряды
Бутстреп	Малые выборки, оценка разброса

Регуляризация

Штраф за большие коэффициенты ограничивает сложность модели

L2 · Ridge

$$\text{MSE} + \alpha \sum_{j=1}^{k} a_j^2$$

Уменьшает все коэффициенты, не обнуляя

L1 · Lasso

$$\text{MSE} + \alpha \sum_{j=1}^{k} |a_j|$$

Обнуляет незначимые — отбор признаков

Elastic Net

L1 + L2

Доля L1 — параметр l1_ratio

- $\alpha = 0$ — обычная регрессия; большое α — недообучение; в LogisticRegression $C = \frac{1}{\alpha}$
- Перед регуляризацией признаки стандартизируют

Параметры и гиперпараметры

Параметры (из данных)	Гиперпараметры (до обучения)
Коэффициенты регрессии	α или C
Пороги разбиений дерева	Глубина дерева
Вероятности в Байесе	Сглаживание alpha, число кластеров

- **Grid Search** — все комбинации; надёжно, но дорого
- **Random Search** — случайные комбинации; быстрее в большом пространстве
- **Байесовская оптимизация** — Optuna, scikit-optimize

Правильная схема и утечка данных

1

Отложить тест и не трогать до конца

2

На обучающих данных — выбор модели кросс-валидацией

3

Обучить итоговую модель и **один раз** оценить на тесте

Утечка: масштабирование или заполнение пропусков по всему набору до разделения. Защита — вся предобработка внутри Pipeline

Строже: вложенная кросс-валидация (внешний цикл — оценка, внутренний — подбор)

Важность признаков

- **Коэффициенты** линейных моделей (на стандартизированных признаках)
- **feature_importances_** деревьев — уменьшение неоднородности
- **Перестановочная важность** — падение качества при перемешивании признака
- **RFE** — последовательное удаление слабых признаков

Подозрительно важный признак — сигнал возможной утечки

Кросс-валидация: набор Diabetes

```
cv = KFold(n_splits=5, shuffle=True, random_state=42)
poly = make_pipeline(PolynomialFeatures(2, include_bias=False),
                     StandardScaler(), LinearRegression())
scores = cross_val_score(model, X_train, y_train, cv=cv, scoring='r2')
```

Модель	R^2 обучение	R^2 CV
Линейная (10 признаков)	0,528	0,480 ± 0,041
Полином 2 (65 признаков)	0,606	0,339 ± 0,070

Полином лучше на обучении, хуже на CV — переобучение: 65 признаков на 353 объекта

Регуляризация полиномиальной модели

Без регуляризации

≈ 0,42

R^2 на тесте

Ridge ($\alpha = 10$)

0,50

R^2 на тесте

Lasso ($\alpha = 1$)

0,52

R^2 на тесте

Lasso обнулила **29 из 65** коэффициентов — автоматический отбор признаков

Поиск по сетке и случайный поиск

```
param_grid = {'ridge__alpha': [0.1, 1, 10, 100, 1000]}
grid = GridSearchCV(pipe, param_grid, cv=cv, scoring='r2')
grid.fit(X_train, y_train)    # лучший alpha = 100

RandomizedSearchCV(pipe,
  {'logisticregression__C': loguniform(1e-3, 1e2)},
  n_iter=20, cv=5, scoring='roc_auc')  # C ≈ 0,13, AUC ≈ 0,993
```

- Имя гиперпараметра: «шаг__параметр»
- Предобработка внутри конвейера — на каждом блоке заново, утечки нет

Кривая валидации: глубина дерева

```
depths = [1, 2, 3, 4, 5, 6, 8, 10]
train_sc, val_sc = validation_curve(
  DecisionTreeClassifier(random_state=0), Xb, yb,
  param_name='max_depth', param_range=depths,
  cv=StratifiedKFold(5, shuffle=True, random_state=0))
```

Обучение

Растёт до 1,0 при глубине 8 и больше

Валидация

Максимум ≈ 0,94 при глубине 5, затем 0,93

Выводы

- Важна ошибка обобщения, а не ошибка на обучении
- Оптимум — компромисс смещения и разброса
- Кросс-валидация и бутстреп дают надёжную оценку
- Регуляризация ограничивает сложность модели
- Тест — один раз; предобработка — внутри Pipeline

Далее: обучение без учителя — кластерный анализ

Контрольные вопросы

1. Ошибка обучения и обобщения, пере- и недообучение: как их обнаружить?
2. Причины переобучения по Tan и соавт.; принцип бритвы Оккама.
3. K-кратная кросс-валидация против отложенной выборки, Stratified, LOOCV, бутстрепа.
4. L1 и L2: функции потерь; почему Lasso обнуляет коэффициенты; роль α.
5. Параметры и гиперпараметры; правильная схема данных; два примера утечки.
6. GridSearchCV для max_depth и min_samples_leaf на Breast Cancer: лучшие значения и точность.

Литература

1. Tan P.-N., Steinbach M., Kumar V. Introduction to Data Mining. — Pearson, 2014. — 4.4; 4.5; 5.6.3.
2. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. — 2017. — Глава 5.
3. VanderPlas J. Python Data Science Handbook. — O’Reilly, 2016. — Chapter 5. Hyperparameters and Model Validation.
4. Грас Дж. Data Science. Наука о данных с нуля. — 2020. — Глава 11.
5. Документация scikit-learn: Cross-validation; Tuning the hyper-parameters.