ИНТЕЛЛЕКТУАЛЬНОЕ МОДЕЛИРОВАНИЕ

Образовательная программа «6B06103 – Компьютерная инженерия»

Лекция 15

Повторение лекций 9–14 и дальнейшее развитие алгоритмов

Неделя 15

Цель и план лекции

Цель: систематизировать лекции 9–14 и научиться обоснованно выбирать алгоритм

1. Классификация, метрики и валидация
2. Кластеризация и ассоциативные правила
3. Типовые задачи и сравнение моделей
4. Развитие изученных алгоритмов
5. Алгоритмическая сторона и что дальше

Логика второй половины курса

Лекция	Главный вопрос	Ключевые понятия
9	Как классифицировать по вероятностям?	Байес, сглаживание
10	Как построить границу классов?	Сигмоида, Джини, прирост
11	Как измерить качество?	MAE, F1, ROC AUC
12	Как надёжно оценить и улучшить?	Кросс-валидация, L1, L2
13	Как найти группы без меток?	Центроид, SSE, силуэт
14	Что встречается вместе?	Поддержка, лифт, Apriori

Наивный Байес

$$\hat{y} = \arg\max_{y} \; P(y) \prod_{i=1}^{d} P(x_i \mid y)$$

- Признаки считаются **условно независимыми** при известном классе
- Нулевые вероятности устраняет **сглаживание**
- Непрерывные признаки — нормальное распределение
- Быстрый метод, хорош на текстах и малых выборках

Логистическая регрессия

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-z}}, \qquad z = a_1 x_1 + \cdots + a_k x_k + b$$

- Обучается минимизацией **log-loss**
- Строит **линейную** границу решений
- Рост признака на единицу умножает шансы на множитель

$$\text{odds ratio} = e^{a_j}$$

Дерево решений

$$\mathrm{Gini}(t) = 1 - \sum_i p(i \mid t)^2, \quad \mathrm{Entropy}(t) = -\sum_i p(i \mid t) \log_2 p(i \mid t)$$

$$\Delta = I(\mathrm{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

- Строится **жадно**: разбиение с наибольшим приростом
- Правила «если — то», нелинейная граница
- Без ограничений **переобучается**: max_depth, обрезка

Какую метрику выбрать

Ситуация	Метрика
Регрессия: типичная ошибка	MAE
Регрессия: большие ошибки особенно нежелательны	RMSE
Классы сбалансированы	Accuracy
Дорог пропуск положительного объекта	Recall
Дорога ложная тревога	Precision
Несбалансированные классы, компромисс	F1
Ранжирование без выбора порога	ROC AUC

Метрики по матрице ошибок

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F_1 = \frac{2\,TP}{2\,TP + FP + FN}$$

$$\text{MAE} = \frac{1}{n}\sum_i |y_i - \hat{y}_i|, \quad \text{RMSE} = \sqrt{\frac{1}{n}\sum_i (y_i - \hat{y}_i)^2}$$

- Accuracy вводит в заблуждение при несбалансированных классах
- Порог определяется ценой ошибок

Валидация и улучшение моделей

- **Недообучение:** ошибки высоки на обучении и тесте
- **Переобучение:** на обучении мала, на тесте велика

Инструмент	Для чего нужен
Stratified K-Fold	Надёжная оценка классификатора
Ridge и Lasso	Ограничение сложности; Lasso отбирает признаки
GridSearchCV	Подбор гиперпараметров
Pipeline	Защита от утечки данных

Тест используют один раз — для итоговой оценки

Метод k-средних

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} \mathrm{dist}(c_i, x)^2, \qquad s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- Назначение точек и пересчёт центроидов уменьшают SSE
- Сходится к **локальному** минимуму: k-means++ и несколько запусков
- Признаки масштабируют; K — по методу локтя и силуэту

Ассоциативные правила

$$s(X \to Y) = \frac{\sigma(X \cup Y)}{N}, \quad c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}, \quad \text{Lift} = \frac{c(X \to Y)}{s(Y)}$$

- Если набор редкий, редки и все его надмножества
- **Apriori** — уровни и кандидаты
- **FP-Growth** — FP-дерево и два прохода
- **Eclat** — пересечение TID-списков

Сводная таблица алгоритмов

Алгоритм	Сильная сторона	Слабая сторона
Линейная регрессия	Простота, интерпретация	Линейная форма
Наивный Байес	Скорость, тексты	Независимость признаков
Логистическая регрессия	Вероятности, шансы	Линейная граница
Дерево решений	Правила, нелинейность	Переобучение
k-средних	Простота, большие данные	Шарообразные кластеры
Apriori, FP-Growth, Eclat	Понятные правила	Много неинтересных правил

Задачи 1 и 2. Вероятности и шансы

10 писем, 4 спама. «приз»: 3/4 и 1/6; «встреча»: 1/4 и 3/6

$$0{,}4 \cdot \frac{3}{4} \cdot \frac{1}{4} = 0{,}075, \quad 0{,}6 \cdot \frac{1}{6} \cdot \frac{3}{6} = 0{,}05, \quad P = \frac{0{,}075}{0{,}125} = 0{,}6$$

Модель z = −2 + 0,8x при x = 5

$$z = 2, \quad p = \frac{1}{1 + e^{-2}} \approx 0{,}881, \quad \text{odds} = e^{2} \approx 7{,}39, \quad e^{0{,}8} \approx 2{,}23$$

- Граница решений: x = 2,5

Задача 3. Выбор разбиения

Родитель: 4 «+», 6 «–». Слева 4 «+», 1 «–»; справа 0 «+», 5 «–»

Мера	Родитель	Левый	Правый	Взвешенно	Прирост
Gini	0,480	0,320	0	0,160	0,320
Энтропия	0,971	0,722	0	0,361	0,610

Правый узел чист, поэтому разбиение сильно снижает неоднородность

Задача 4. Метрики

	Предсказан +	**Предсказан –**
Фактически +	TP = 40	FN = 10
Фактически –	FP = 30	TN = 920

$$\text{Precision} = \frac{40}{70} \approx 0{,}571, \quad \text{Recall} = \frac{40}{50} = 0{,}8, \quad F_1 = \frac{80}{120} \approx 0{,}667$$

- Accuracy = 0,960; Specificity = 920/950 ≈ 0,968

Модель «всегда отрицательный»: accuracy 0,95, но recall 0

Задача 5. Диагностика модели

Обучение	CV	Диагноз	Что делать
0,99	0,80	Переобучение	Упростить, регуляризировать, добавить данные
0,65	0,63	Недообучение	Усложнить, улучшить признаки

Подбирать гиперпараметры по тестовой выборке нельзя: оценка окажется завышенной

Задача 6. Метод k-средних

(1, 1), (2, 2), (8, 8), (9, 9); K = 2; старт — первые две точки

1. Центроиды: (1; 1) и (6,33; 6,33)
2. (2, 2) переходит в первый кластер: (1,5; 1,5) и (8,5; 8,5)
3. Кластеры не меняются — алгоритм завершён

$$\mathrm{SSE} = 4\,(0{,}5^2 + 0{,}5^2) = 2, \qquad s(1,1) = 1 - \frac{1{,}414}{10{,}607} \approx 0{,}87$$

Задача 7. Ассоциативные правила

{a, b, c}, {a, b}, {a, c}, {b, d}, {a, b, d}

Правило	s	c	Lift	Вывод
{a} → {b}	0,6	0,75	0,94	Связь слабо отрицательная
{d} → {b}	0,4	1,00	1,25	Положительная связь
{c} → {a}	0,4	1,00	1,25	Положительная связь

Высокая достоверность сама по себе не означает связи

Сравнение классификаторов: код

Breast Cancer: положительный класс — злокачественная опухоль

```
X, y = data.data, 1 - data.target
cv = StratifiedKFold(5, shuffle=True, random_state=42)
scores = cross_validate(model, X_train, y_train, cv=cv,
                        scoring=['recall', 'f1', 'roc_auc'])
model.fit(X_train, y_train)
pred = model.predict(X_test)
```

Сравнение классификаторов: результат

Модель	CV recall	CV F1	CV AUC	Тест recall	Тест F1
Наивный Байес	0,893	0,916	0,986	0,849	0,918
Логистическая регрессия	0,944	0,961	0,991	0,925	0,951
Дерево глубины 3	0,874	0,892	0,923	0,736	0,830
Случайный лес	0,937	0,943	0,986	0,925	0,961
Градиентный бустинг	0,918	0,934	0,987	0,906	0,950

По кросс-валидации лучше простая логистическая регрессия: сложнее не значит лучше

Как выбирать алгоритм

1. **Есть ли целевая переменная?** Определяет задачу
2. **Нужно ли объяснять решение?** Линейные модели, небольшие деревья
3. **Какие данные?** Тексты, таблицы, изображения
4. **Какова цена ошибок?** Определяет метрику и порог
5. **Какие ресурсы?** Время, память, переобучение
6. **Что показала проверка?** Кросс-валидация и базовая модель

От изученных методов к новым

Алгоритм курса	Ограничение	Развитие
Наивный Байес	Независимость признаков	Байесовские сети
Логистическая регрессия	Линейная граница	SVM с ядрами, нейросети
Дерево решений	Большой разброс	Лес, бустинг
k-средних	Шарообразные кластеры	DBSCAN, гауссовы смеси
Apriori, FP-Growth	Нет порядка событий	Последовательные шаблоны, рекомендации
GridSearchCV	Дорогой перебор	Байесовская оптимизация, AutoML

Ансамбли: от дерева к лесу и бустингу

- **Бэггинг:** модели на бутстреп-выборках, уменьшает разброс
- **Случайный лес:** ещё и случайные признаки в разбиениях
- **Бустинг:** каждая модель исправляет ошибки предыдущих, уменьшает прежде всего смещение
- Реализации: XGBoost, LightGBM, CatBoost

Плата за качество — меньшая интерпретируемость

Нейронные сети

- Логистическая регрессия — это один **нейрон**
- Слои нейронов строят признаки автоматически
- Обучение: градиентный спуск, **обратное распространение ошибки**
- CNN — изображения; RNN и трансформеры — последовательности и тексты

Нелинейные данные: сравнение

make_moons: 300 объектов, два полумесяца с шумом

Модель	Обучение	Тест
Логистическая регрессия	0,848	0,844
Дерево глубины 4	0,938	0,867
Случайный лес	0,981	0,989
Нейронная сеть	0,981	1,000

В тесте всего 90 объектов: результат 1,000 — не гарантия безошибочности

Кластеризация за пределами k-средних

Метод	Идея
DBSCAN	Плотные области, шум, число кластеров не задаётся
Иерархическая	Дерево вложенных групп
Гауссовы смеси	Мягкое назначение с вероятностями

```
k-средних, ARI: 0.24   # полумесяцы
DBSCAN, ARI:    1.0
```

Параметры eps и min_samples нужно подбирать; универсально лучшего метода нет

От ассоциаций к рекомендациям

- Шаблоны ищут не только в наборах, но и в **последовательностях** и **графах**
- **Коллаборативная фильтрация:** то, что нравится похожим пользователям
- Меры сходства из лекции 2 здесь снова работают
- **Эмбеддинги:** пользователи и товары — числовые векторы

Алгоритмические идеи курса

Идея	Где встречается
Жадный выбор	Построение дерева
Итеративное улучшение	k-средних, градиентный спуск
Разделяй и властвуй	Деление узлов дерева, FP-Growth
Обход в ширину и глубину	Apriori, Eclat
Отсечение перебора	Принцип Apriori, обрезка дерева
Рандомизация	k-means++, случайный поиск, бутстреп
Структуры данных	FP-дерево, TID-списки, разреженные матрицы

Вычислительная сложность

m — объекты, n — признаки, I — итерации, K — кластеры, d — товары

Алгоритм	Обучение
Наивный Байес	$O(m \cdot n)$
Линейная регрессия (нормальные уравнения)	$O(m \cdot n^2 + n^3)$
Логистическая регрессия (градиентный спуск)	$O(I \cdot m \cdot n)$
Дерево решений	$O(n \cdot m \cdot \log m)$ на уровень
k-средних	$O(I \cdot K \cdot m \cdot n)$
Apriori	До 2^d кандидатов

Трудные задачи и большие данные

- Оптимальное дерево — **NP-полная** задача (Hyafil, Rivest, 1976)
- Минимизация SSE — **NP-трудная** (Aloise и соавт., 2009)
- Поэтому применяют эвристики: они быстры, но не гарантируют оптимум

- **Масштабирование:** мини-батчи и SGD, выборка, приближённые алгоритмы, распределённые вычисления

Куда двигаться дальше

Направление	Что изучается
Методы машинного обучения	SVM, ансамбли, снижение размерности
Глубокое обучение	Нейросети, CNN, трансформеры
Обработка текстов	Векторные представления, языковые модели
Временные ряды	Тренд, сезонность, прогноз
Большие данные	Распределённые вычисления, Spark, SQL
MLOps	Развёртывание, мониторинг, дрейф данных
Ответственный ИИ	Объяснимость, справедливость, приватность

Выводы

- Каждый алгоритм имеет свои сильные стороны и ограничения
- Качество оценивают метриками и кросс-валидацией
- Сложная модель не обязательно лучше
- Новые методы продолжают идеи курса
- В основе — общие алгоритмические идеи

Основа: понять задачу и данные, начать с простой модели, честно оценить результат

Контрольные вопросы

1. Сравните наивный Байес, логистическую регрессию и дерево решений.
2. TP = 25, FN = 5, FP = 15, TN = 155: найдите accuracy, precision, recall, F1.
3. Опишите схему разделения данных; что означают 0,99 на обучении и 0,78 на кросс-валидации?
4. Две итерации k-средних: (0, 0), (1, 0), (5, 5), (6, 5); лифт {a} → {b} для {a, b}, {a, c}, {a, b, c}, {b, c}.
5. Как ансамбли, нейросети и DBSCAN снимают ограничения базовых методов?
6. Какие алгоритмические идеи лежат в основе дерева, k-средних, Apriori и FP-Growth?

Литература

1. Tan P.-N., Steinbach M., Kumar V. Introduction to Data Mining. — Pearson, 2014. — Chapters 4–9.
2. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. — Springer, 2009.
3. Goodfellow I., Bengio Y., Courville A. Deep Learning. — MIT Press, 2016.
4. Levitin A. Introduction to the Design and Analysis of Algorithms. — Pearson, 2012.
5. Breiman L. Random Forests. — 2001; Friedman J. H. Greedy Function Approximation. — 2001.
6. Ester M. et al. DBSCAN. — KDD, 1996; Vaswani A. et al. Attention Is All You Need. — 2017.
7. Документация scikit-learn: Ensembles; Neural network models; Clustering.