Современные методы машинного обучения

Лекция 1

Турдаков Денис Юрьевич

План

- Вводная часть о курсе
- Задачи машинного обучения
  - Регрессия (линейная регрессия)
  - Классификация (логистическая регрессия, наивный байесовский классификатор)
  - Кластеризация (k-means)
- Обзор современных подходов и проблем

Часть 1. О курсе

О курсе

- Официальное название спецкурса (для учебной части):
  - бакалавриат — «Современные методы машинного обучения»
  - магистратура — «Математические основы и приложения нейронных сетей»
- Лекции по средам в 18.00 (Zoom)
  - предполагаются минимальные знания
    - линейной алгебры,
    - теории вероятности и математической статистики,
    - программирования
  - не все имеют одинаковые знания
    - предполагается, что студенты могут быстро учиться

О курсе

- Совместный курс ВМК МГУ, Samsung Research Russia, ИСП РАН
  - MOOC Курс от SRR на Stepik
    - https://stepik.org/course/50352 - введение в нейронные сети
    - https://stepik.org/course/54098
  - «Живые» лекции в Zoom от сотрудников МГУ и ИСП РАН
    - Опыт применения на практике
    - наиболее интересные современные направления
  - Практическое задание: Защита от состязательных атак на модели для компьютерного зрения
    - автоматическая система тестирования (реализующая атаки)
    - "хор" за практическое задание, если побил baseline 1
    - "отл" за практическое задание, если побил baseline 2
    - возможны автоматы за вхождение в top 3 с существенным/статистически значимым отрывом от 4 и последующих мест

О курсе

- Оценка результатов. Для получения оценки необходимо выполнить одно из условий

Условие	Максимальная оценка
Пройти оба курса на Stepik (сделать все задания)	хорошо
Пройти один курс на Stepik и сделать практическое задание на хорошо	хорошо
Пройти один курс на Stepik и сделать практическое задание на отл	отлично
Пройти оба курса на Stepik (сделать все задания) и сдать экзамен на отлично	отлично
Пройти один курс на Stepik, сделать практическое задание на хорошо, сдать экзамен на отлично	отлично

Часть 2. Машинное обучение

Основные понятия

- X – множество объектов
- Y – множество меток
- $y : X \to Y$ – неизвестная зависимость (целевая функция), значения которой известны на конечном подмножестве объектов $\{x_1, \ldots, x_l\} \subset X$
- $X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка

Постановка задачи

- По X^l восстановить зависимость y

Признаки объектов

- $f_j : X \to D_j,\ j = 1 \ldots n$ – признаки объектов (features)
- Множество объектов задается матрицей

$$F = ||f_j(x_i)||_{l \times n} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_l) & \ldots & f_n(x_l) \end{pmatrix}$$

Алгоритм и модель

- В общем случае зависимость y узнать невозможно, поэтому будем ее приближать некоторой функцией $a : X \to Y$
- Функция $a : X \to Y$ должна допускать эффективную компьютерную реализацию; по этой причине будем называть её **алгоритмом**
- Поиск оптимального алгоритма осуществляют из предположения, что $a \in A = \{g(x, \theta) | \theta \in \Theta\}$ – принадлежит семейству параметрических функций **модель**, где $g : X \times \Theta \to Y$ – фиксированная функция, Θ – пространство поиска

Задачи машинного обучения

Задача	На какой вопрос про входные данные пытается ответить
Классификация (Classification)	Это **A** или **B**?
Восстановление регрессии (Regression)	Сколько этого?
Кластеризация (Clustering)	Как эти данные могут быть сгруппированы?

Задача	На какой вопрос про входные данные пытается ответить
Обучение с подкреплением (Reinforsement learning)	Что мне делать сейчас?
Поиск аномалий (Anomaly detection)	Это аномалия?

Классификация: вероятностная постановка

- $X \times Y$ – вероятностное пространство с распределением $p(x,y) = p(y)p(x|y)$ из которого случайно и независимо выбирается l наблюдений $X^l = (x_i, y_i)_{i=1}^l$
- Будем аппроксимировать $p(x,y)$ через модель совместной плотности распределения объектов и ответов $\phi(x,y,\theta)$
- Определим значение параметров θ, при которых обучающая выборка данных максимально правдоподобна, то есть наилучшим образом согласуется с моделью плотности (метод максимума правдоподобия)

$$L(\theta, X^l) = \prod_{i=1}^{l} \phi(x_i, y_i, \theta) \to max$$

Наивный байесовский классификатор

- Выбор наиболее вероятного значения

$$\hat{y} = \arg\max_{y \in Y} P(y|x) = \arg\max_{y \in Y} P(y|f_1, \ldots, f_n)$$

- По правилу Байеса

$$\hat{y} = \arg\max_{y' in Y} \frac{P(f_1, \ldots, f_n|y)P(y)}{P(f_1, \ldots, f_n)} = \arg\max_{y' in Y} P(f_1, \ldots, f_n|y)P(y)$$

- «Наивное» предположение об условной независимости признаков

$$\hat{y} = \arg\max_{y' in Y} P(y) \prod_{j=1}^{n} P(f_i|y)$$

Обучение наивного байесовского классификатора

- Сделаем предположение о распределении

$$p(f_i|y,\theta) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\pi\sigma_y^2}\right)$$

- Воспользуемся методом максимального правдоподобия для оценки среднего и дисперсии распределения
- Оценка для p(y) – частоты классов в выборке D
- Получим оценку p(x,y)
  - Можем предсказывать y для произвольного x
  - Можно даже генерировать (x,y). Это генеративная модель

Пример

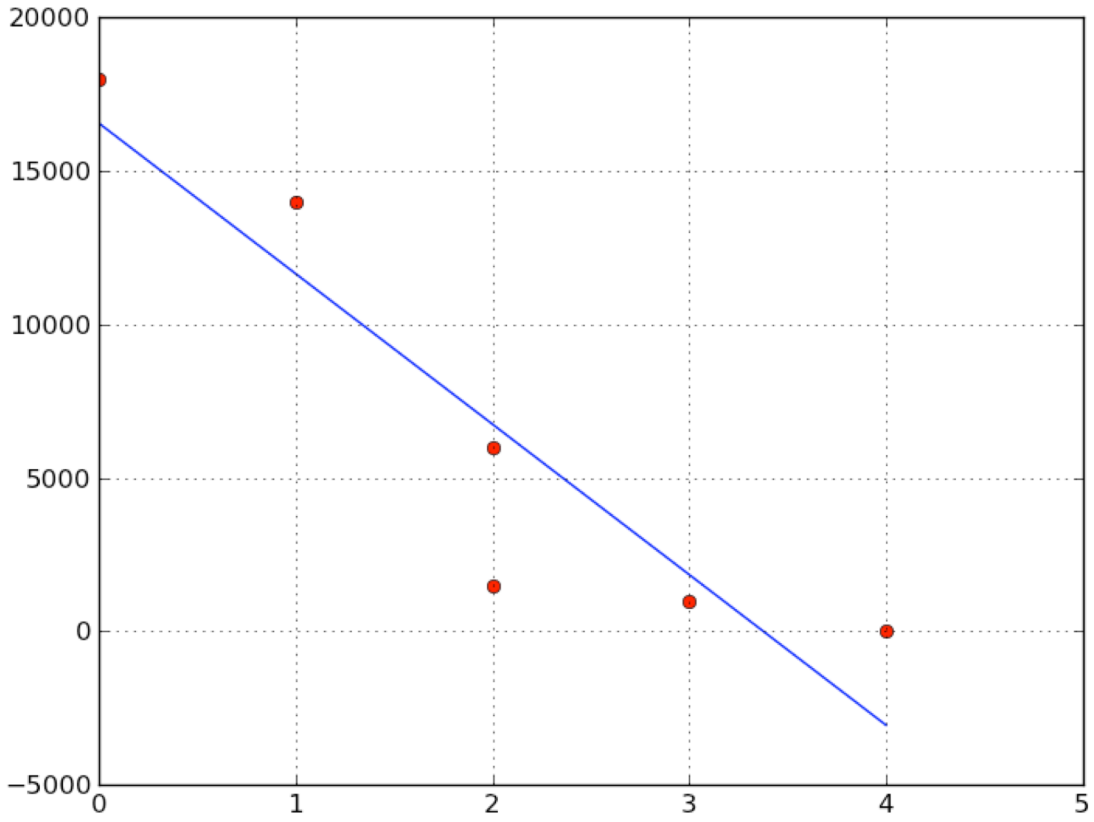
```
from sklearn.nayve_bayes import *

corpus = [['list of texts'],['classes']]

# initialize classifier
classifier = GaussianNB()

# use unigrams and bigrams as features
vectorizer = CountVectorizer(ngram_range=(1,2))
y = corpus[1]
X = vectorizer.fit_transform(corpus[0])
classifier.fit(X,y) # train classifier

#transform new texts into feature vectors
unseen_texts = ["list of unseen texts"]
feature_vectors = vectorizer.transform(unseen_texts)
answers = classifier.predict(feature_vectors)
```

Линейная регрессия

Кол-во неопределенных прилагательных	Прибыль сверх запрашиваемой
4	0
3	$1000
2	$1500
2	$6000
1	$14000
0	$18000

$$price = w_0 + w_1 * Num\_Adjectives$$

Линейная регрессия



Линейная регрессия

$$price = w_0 + w_1 * Num\_Adjectives + w_2 * Mortgage\_Rate + w_3 * Num\_Unsold\_Houses$$

- В терминах признаков

$$price = w_0 + \sum_{i=1}^{N} w_i \times f_i$$

- введем дополнительный признак $f_0 = 1$

$$y = \sum_{i=0}^{N} w_i \times f_i \quad \text{или} \quad y = w \cdot f$$

Вычисление весов признаков

- Минимизировать квадратичную погрешность

$$cost(W) = \sum_{j=0}^{M} (y_{pred}^{j} - y_{obs}^{j})^2$$

- Вычисляется по формуле

$$W = (X^T X)^{-1} X^T \overrightarrow{y}$$

Логистическая регрессия

- Перейдем к задаче классификации
- Определить вероятность, с которой наблюдение относится к классу
- Попробуем определить вероятность через линейную модель

$$P(y = true|x) = \sum_{i=0}^{N} w_i \times f_i = w \cdot f$$

Логистическая регрессия

- Попробуем определить отношение вероятности принадлежать классу к вероятности не принадлежать классу

$$\frac{P(y = true|x)}{1 - P(y = true|x)} = w \cdot f$$

Логистическая регрессия

- Проблема с несоответствием области значений решается вводом натурального логарифма

$$ln\left(\frac{P(y=true|x)}{1-P(y=true|x)}\right)=w\cdot f$$

- Логит-преобразование

$$logit(P(x))=ln\left(\frac{P(x)}{1-P(x)}\right)$$

- Определим вероятность ...

Логистическая регрессия

$$P(y = true|x) = \frac{e^{w \cdot f}}{1 + e^{w \cdot f}} \qquad P(y = false|x) = \frac{1}{1 + e^{w \cdot f}}$$

- Или

$$P(y = true|x) = \frac{1}{1 + e^{-w \cdot f}} \qquad P(y = false|x) = \frac{e^{-w \cdot f}}{1 + e^{-w \cdot f}}$$

- Логистическая функция

$$\frac{1}{1 + e^{-x}}$$

Логистическая регрессия

$$P(y = true|x) > P(y = false|x)$$

$$\frac{P(y = true|x)}{1 - P(y = true|x)} > 1$$

$$e^{w \cdot f} > 1$$

$$w \cdot f > 0$$

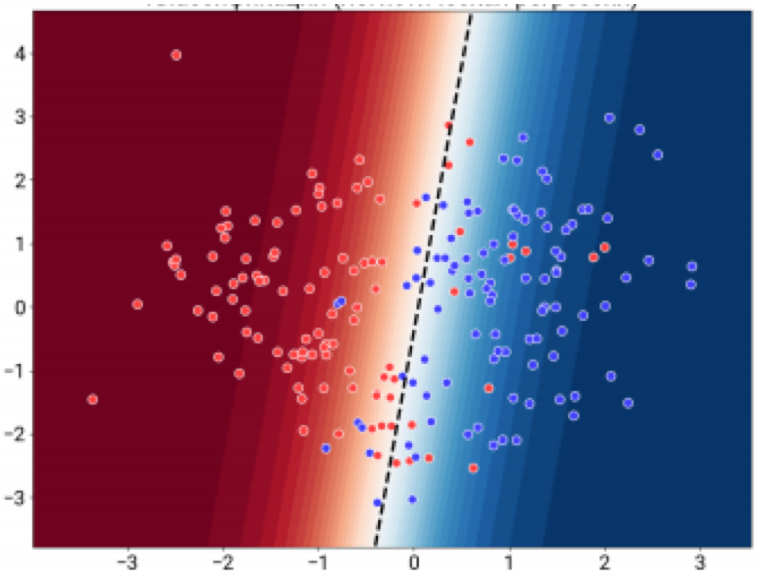
$$\sum_{i=0}^{N} w_i f_i > 0$$

разделяющая гиперплоскость

Обучение

- Оптимизация функции потерь

$$min_w \lambda ||w||^2 + \sum (log(1 + exp(-y_i w^t x_i))$$



Мультиномиальная логистическая регрессия

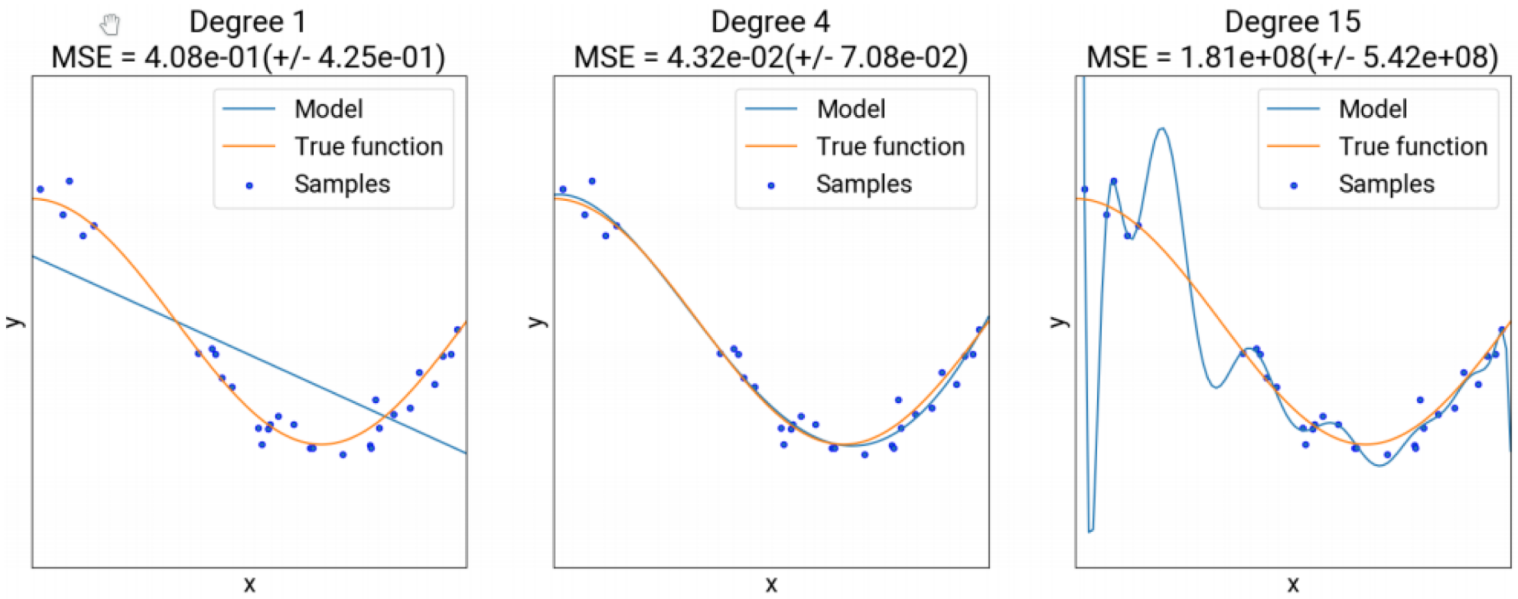
- Классификация на множество классов

$$p(c|x) = \frac{1}{Z} exp(\sum_i w_i f_i)$$

$$p(c|x) = \frac{exp\left(\sum_{i=0}^{N} w_{ci} f_i\right)}{\sum_{c' \in C} exp\left(\sum_{i=0}^{N} w_{c'i} f_i\right)}$$

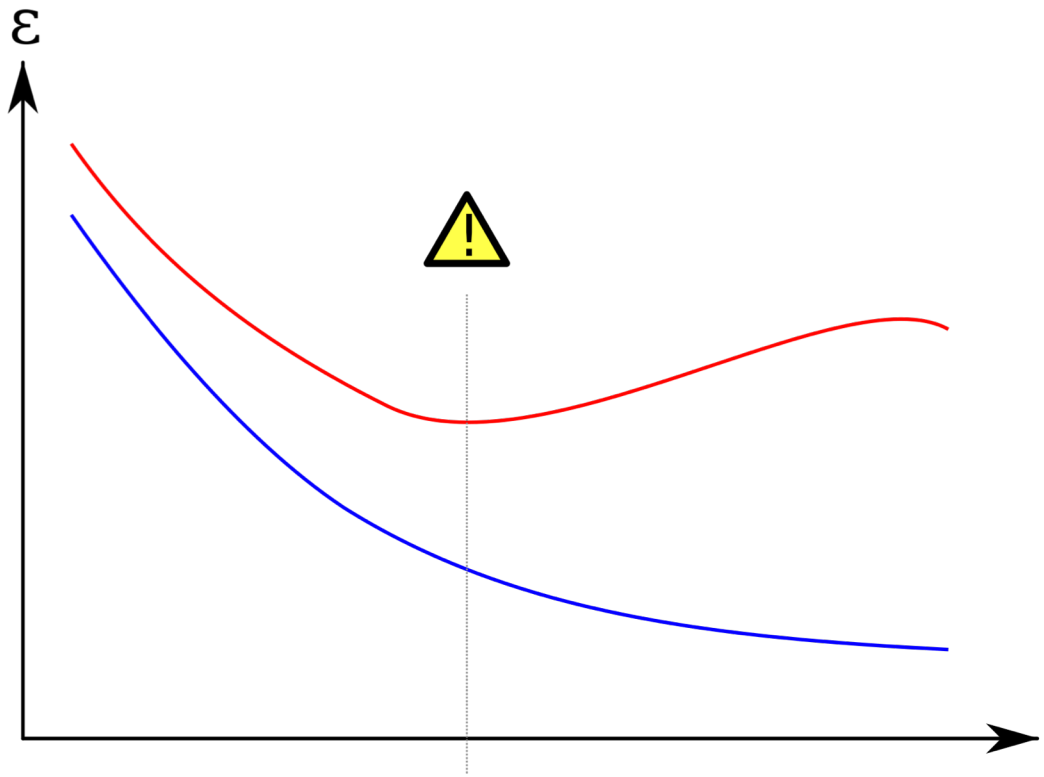
Переобучение

- А что если аппроксимировать данные полиномом с высокой степенью (в качестве признаков брать функции высоких порядков)?



- При d=15 модель получилась слишком сложной и обучилась на шуме

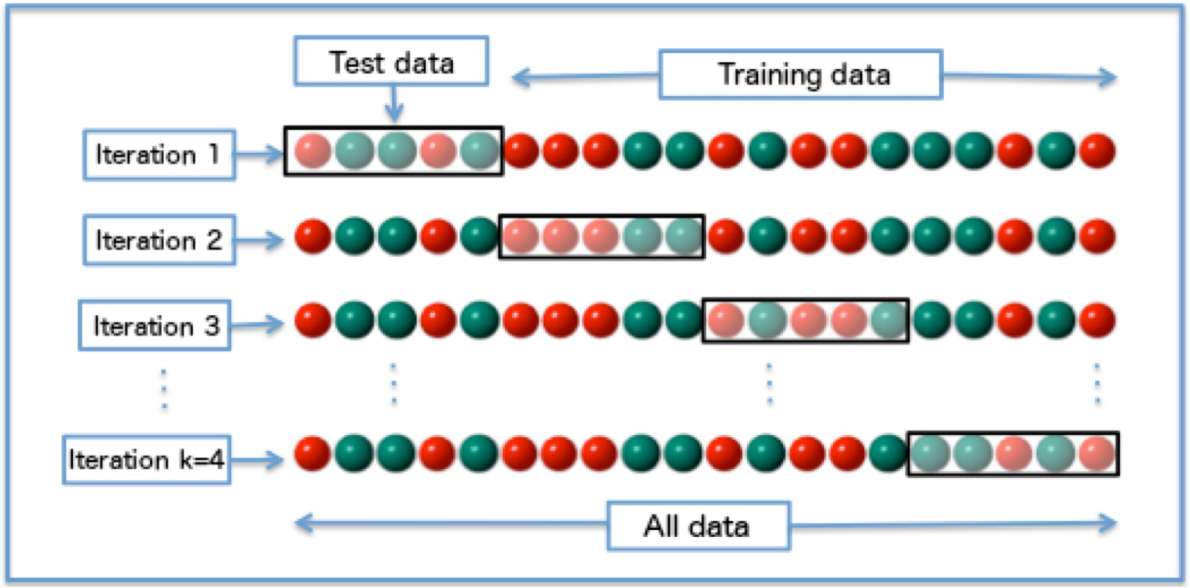
Переобучение



- (Синий) Ошибка на тренировочных данных
- (Красный) Ошибка на валидационных данных

Проведение экспериментов

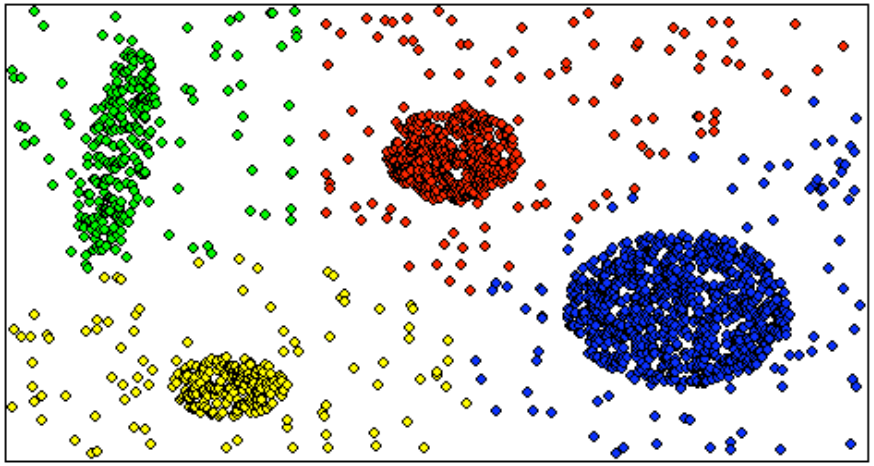
- Данные делятся на несколько частей
  - Тренировочная
  - Тестовая
  - Валидационная
- Перекрестная проверка (cross-validation)



*https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Кластеризация

- Входные элементы можно разбить на несколько групп, по принципу схожести



Вход для алгоритмов

- Пусть каждый объект $\{x_1, x_2, \ldots, x_k\}$ представлен вектором $x_i = (f_{i_1}, \ldots, f_{i_n})$ в пространстве $X \subseteq R^n$
- Задается расстояние между векторами
  - Евклидово $d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$
  - Чебышева $l_\infty(\vec{x}, \vec{y}) = \max_{i=1,\ldots,n} |x_i - y_i|$
  - Хэмминга $d_{ij} = \sum_{k=1}^{p} |x_{ik} - x_{jk}|.$
  - Минковского $\rho(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$
  - ...

Алгоритм K-средних (k-means)

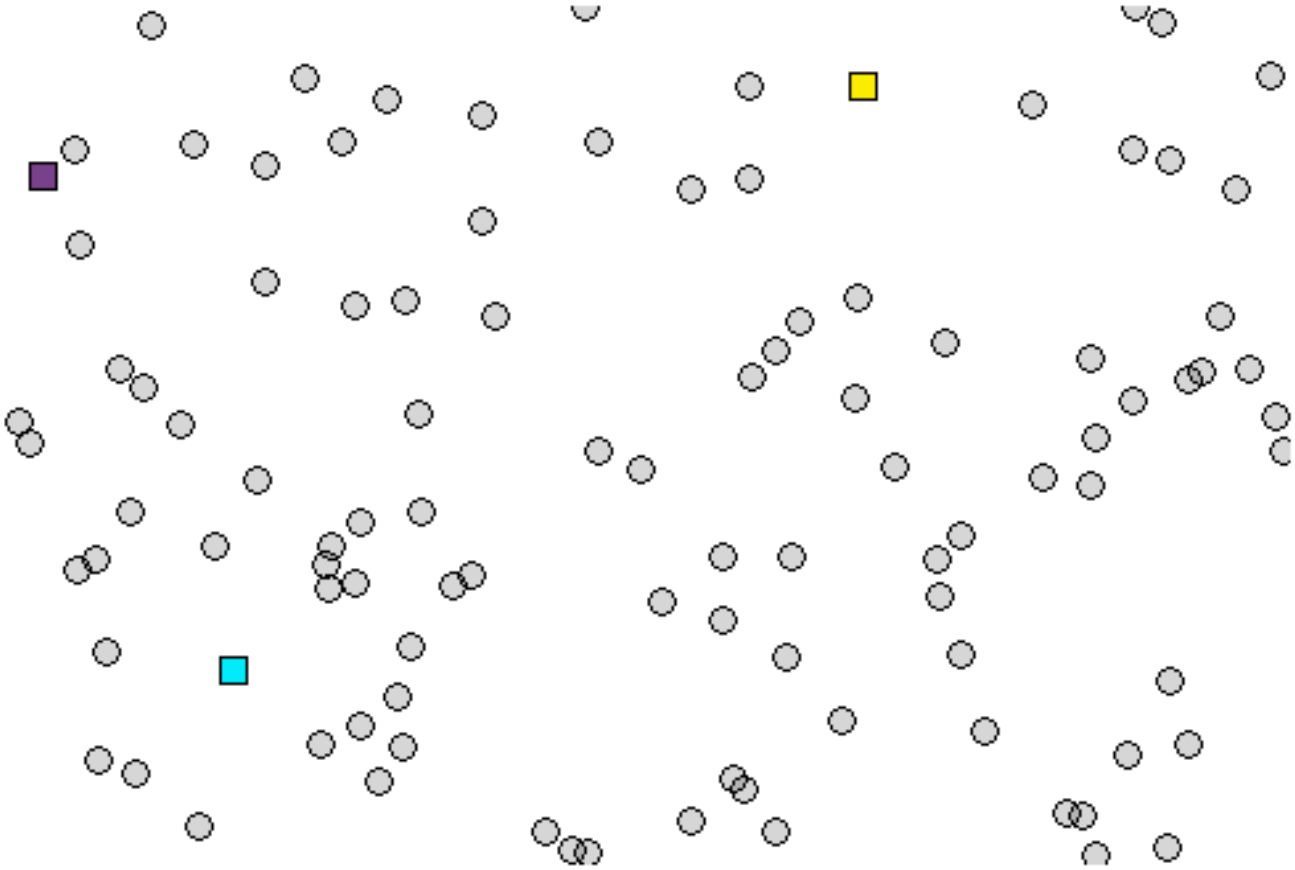
- Алгоритм k-means разбивает данные на k кластеров
  - Каждый кластер имеет центр - центроид
  - Параметр k - задается вручную
- Алгоритм
  1. Выбираются k точек в качестве начальных центроидов
  2. Сопоставить каждой точке ближайший центроид
  3. Пересчитать центроиды
  4. Если алгоритм не сошелся перейти на шаг 2

Критерий останова

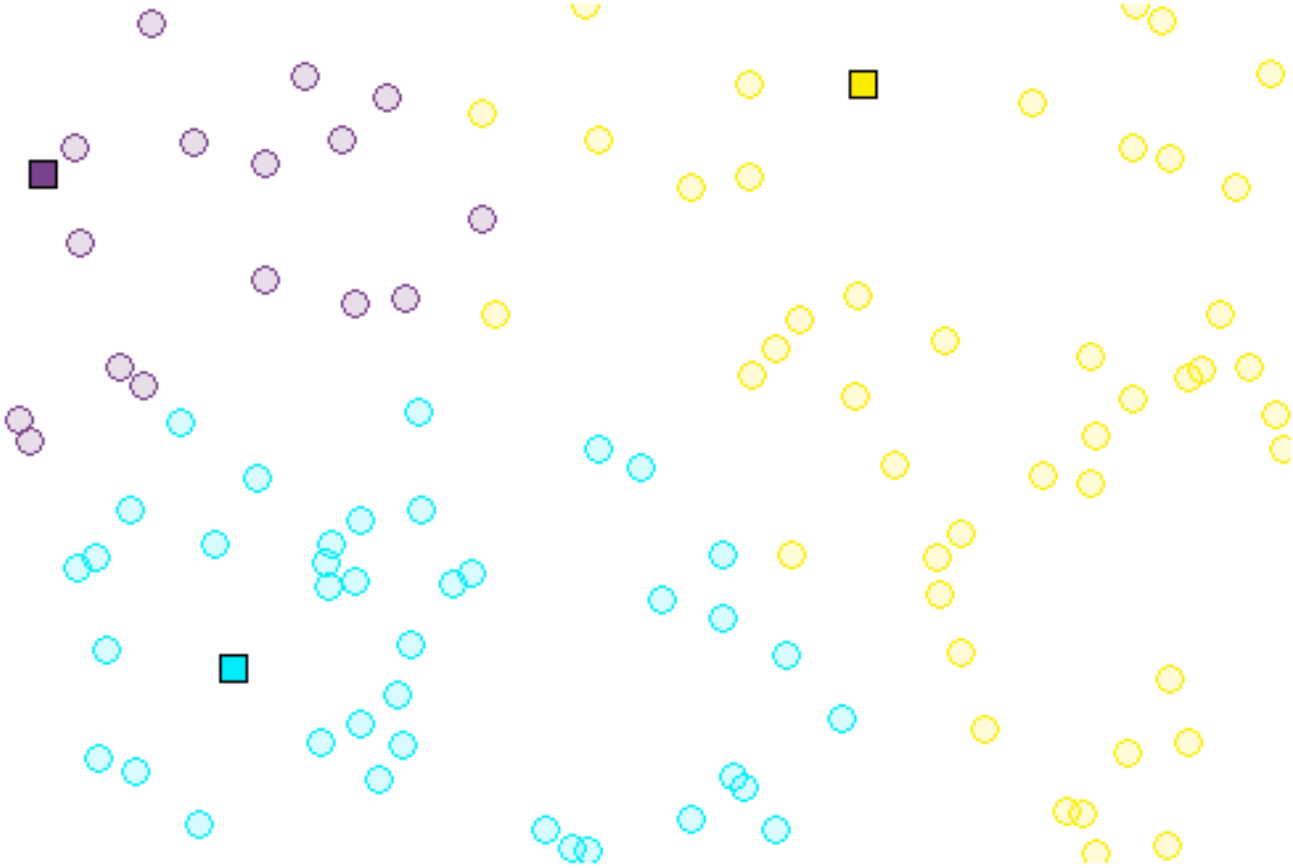
- Нет перехода точек в другой кластер
- Нет (незначительно) изменение центроидов
- Мало убывает погрешность (sum of squared error)

$$SSE = \sum_{j=1}^{k} \sum_{x \in C_j} dist(x, m_j)^2$$

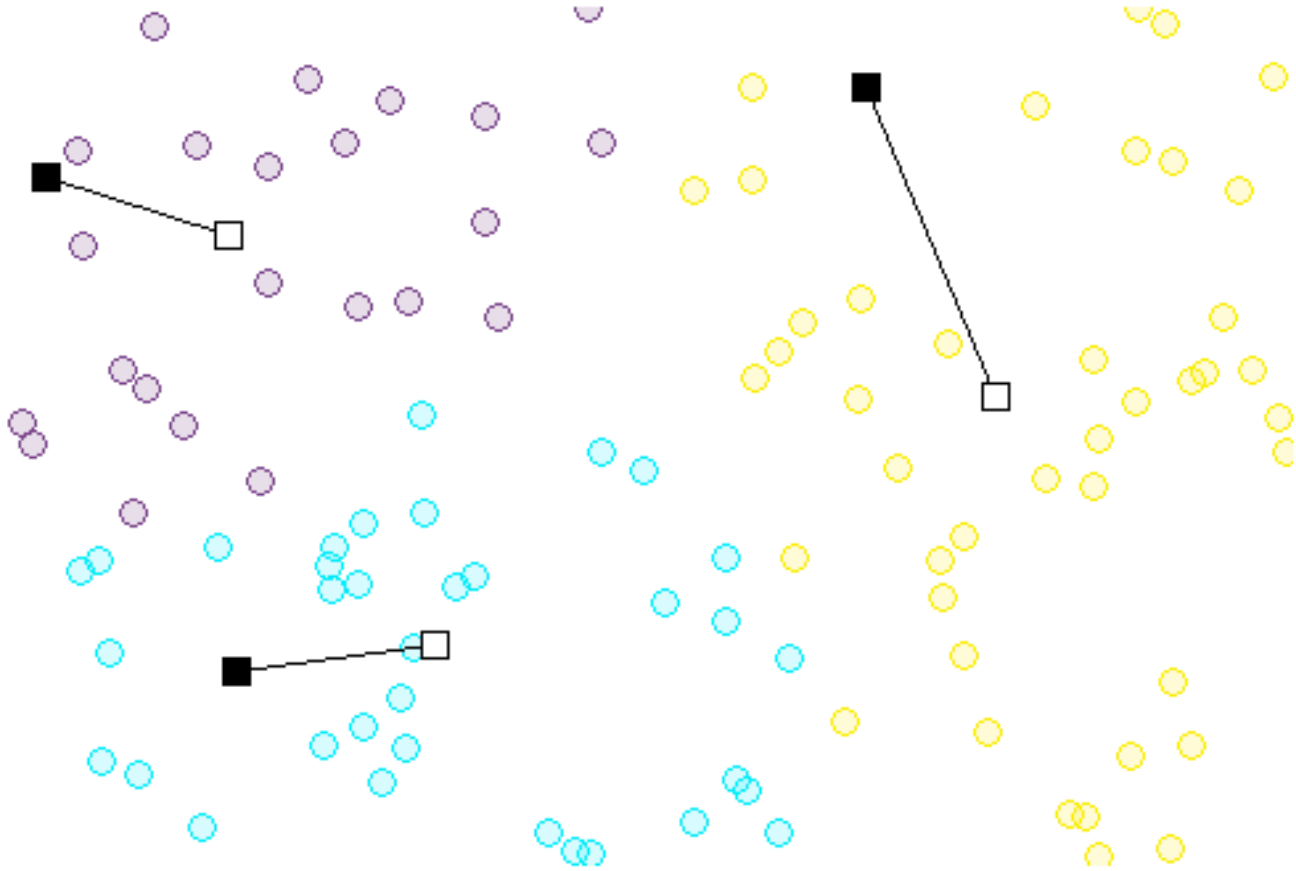
Алгоритм К-средних. Пример.



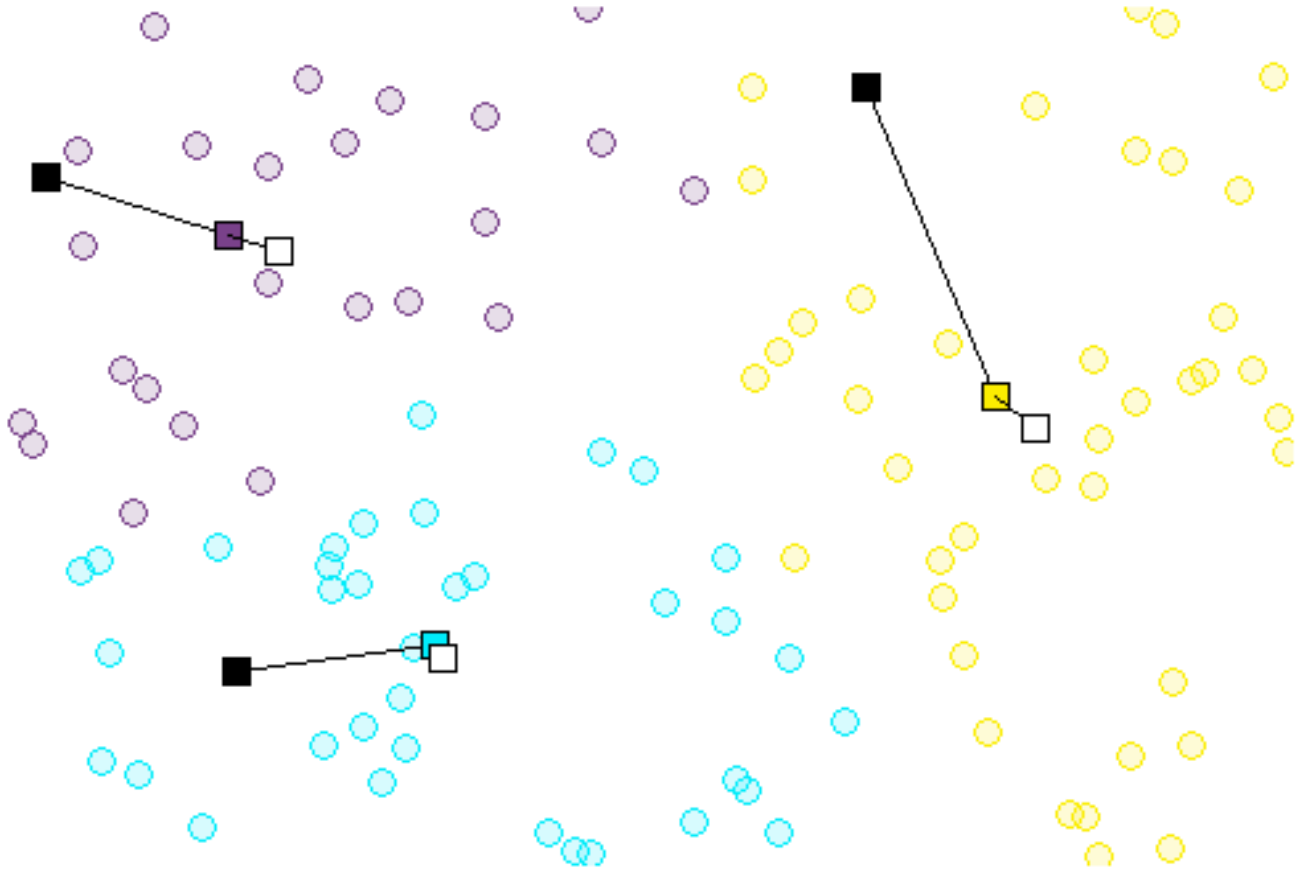
Алгоритм К-средних. Пример.



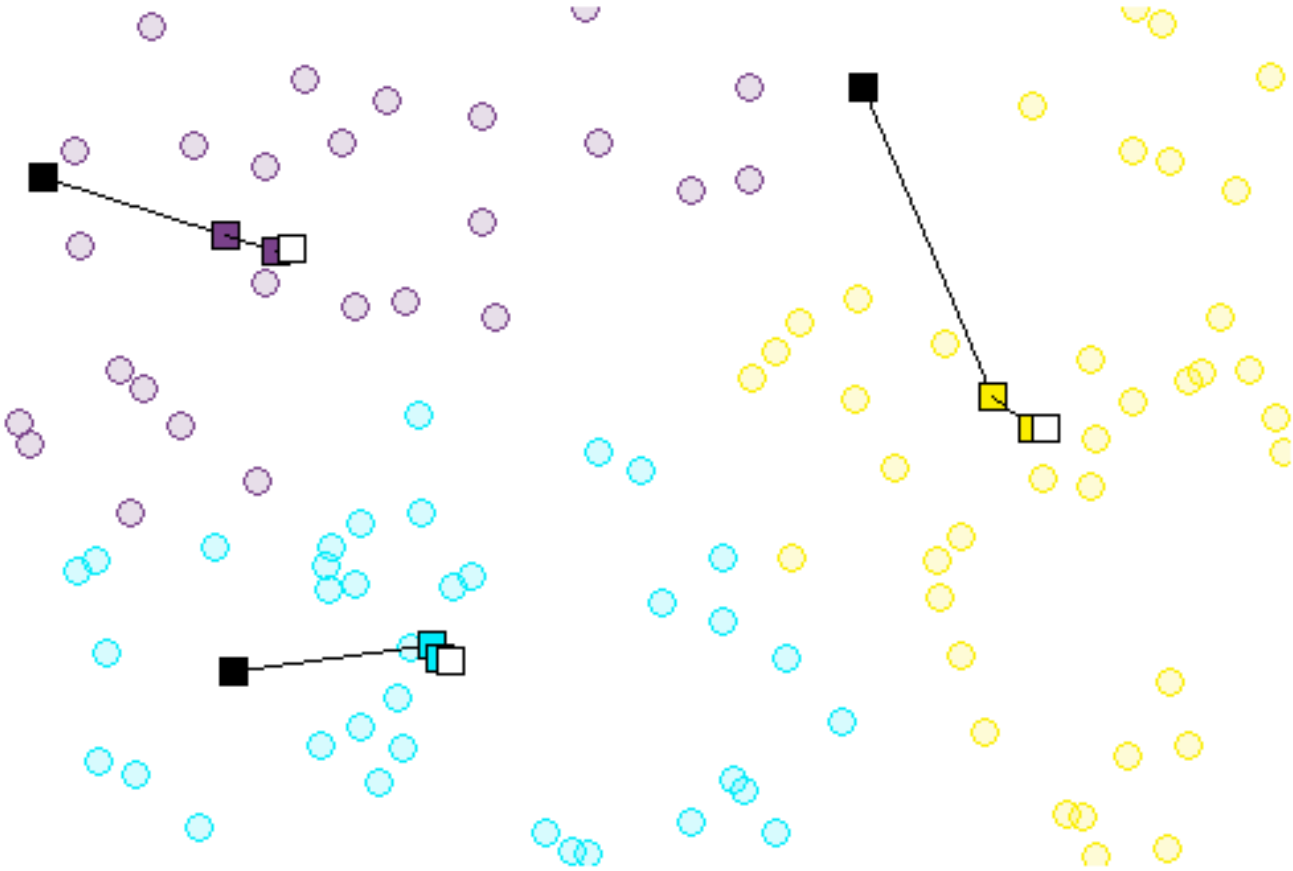
Алгоритм К-средних. Пример.



Алгоритм К-средних. Пример.



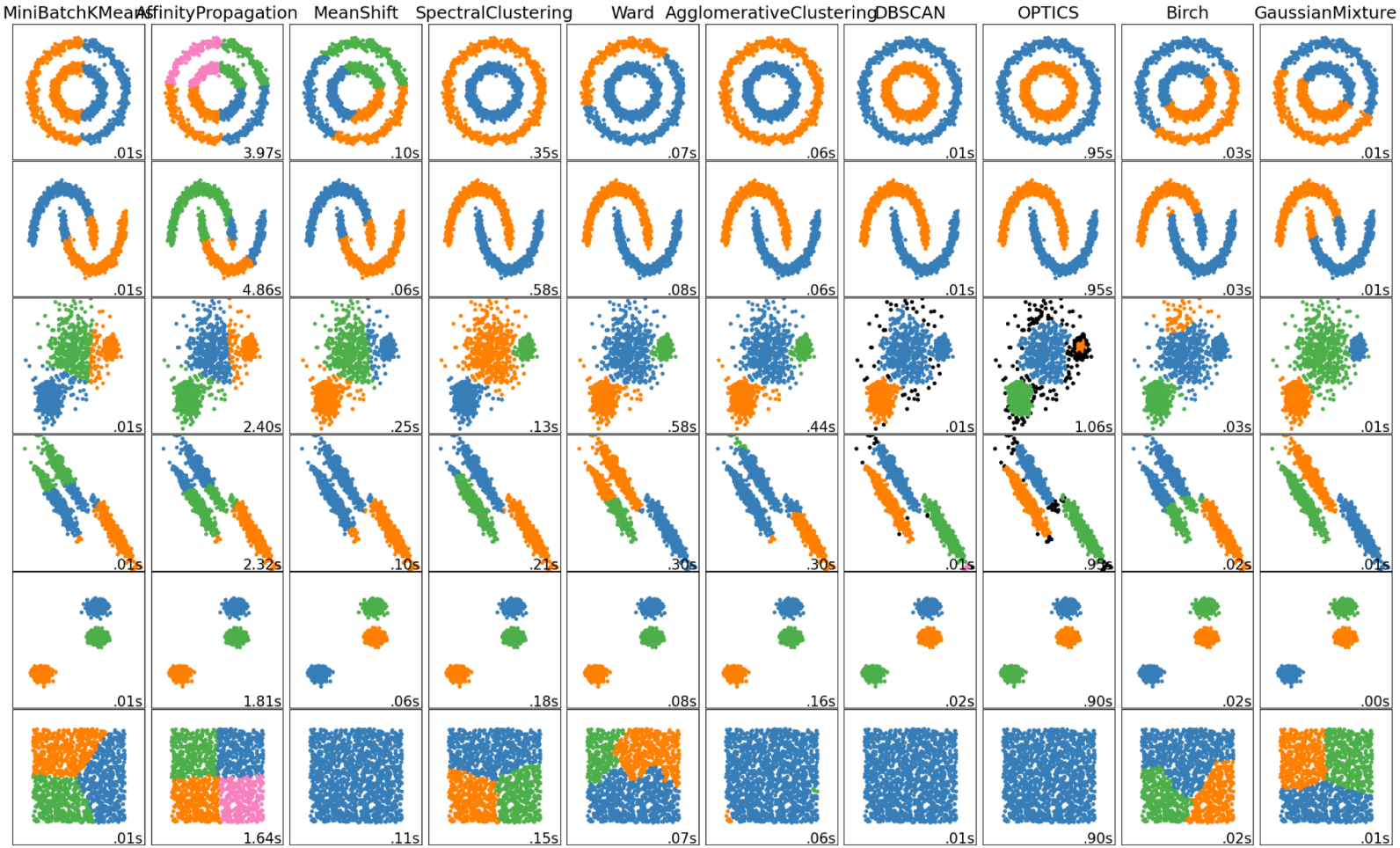
Алгоритм К-средних. Пример.



Проблемы K-means

- Алгоритм чувствителен к начальному выбору центроидов
  - запуск с различной начальной инициализацией и выбор варианта с наиболее плотными кластерами
- Чувствителен к выбросам
  - можно фильтровать выбросы
- Не подходит для нахождения кластеров, не являющихся элипсоидами
  - преобразование пространства

Какой алгоритм кластеризации выбрать

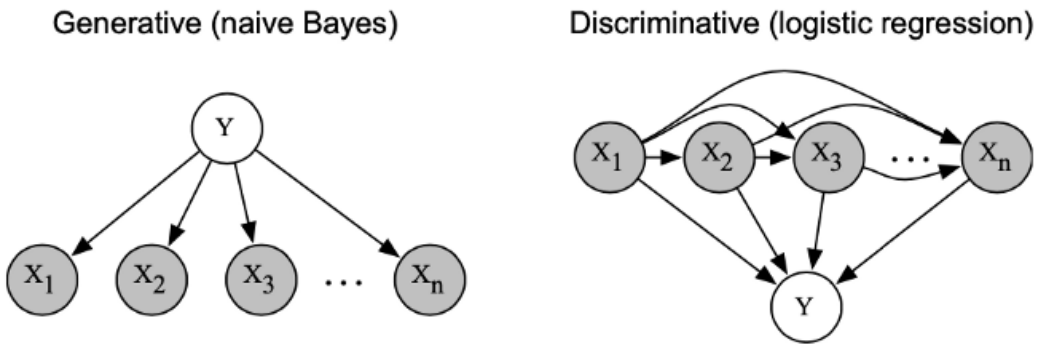


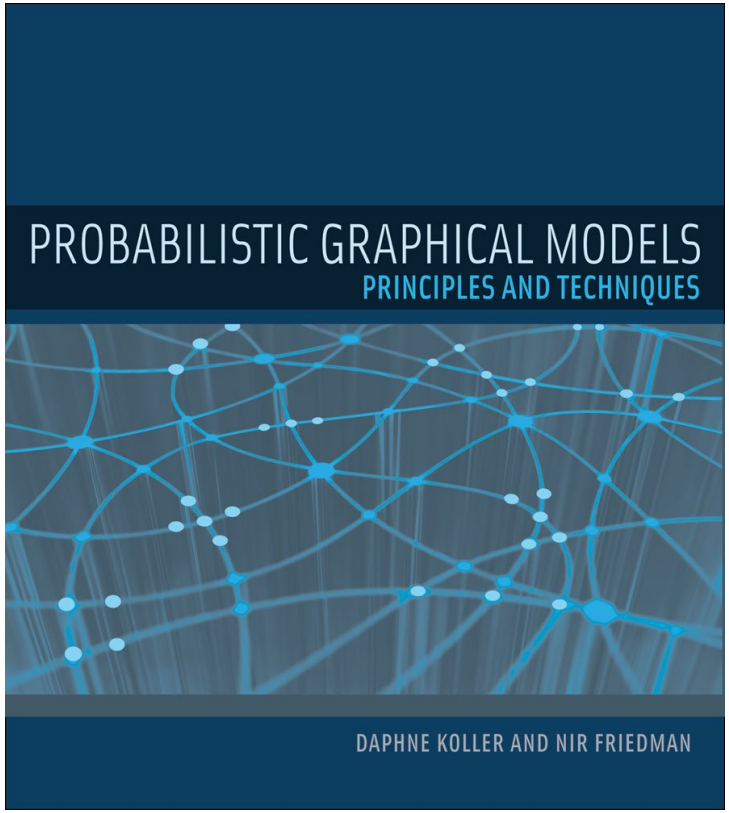
* https://scikit-learn.org/stable/modules/clustering.html

Часть 3. Подходы к машинному обучению

Вероятностные графические модели

- Вероятностные графические модели
  - Зависимости между случайными величинами представляются в виде графа
  - Разделяют наблюдаемые состояния (признаки) и скрытые состояния (классы)



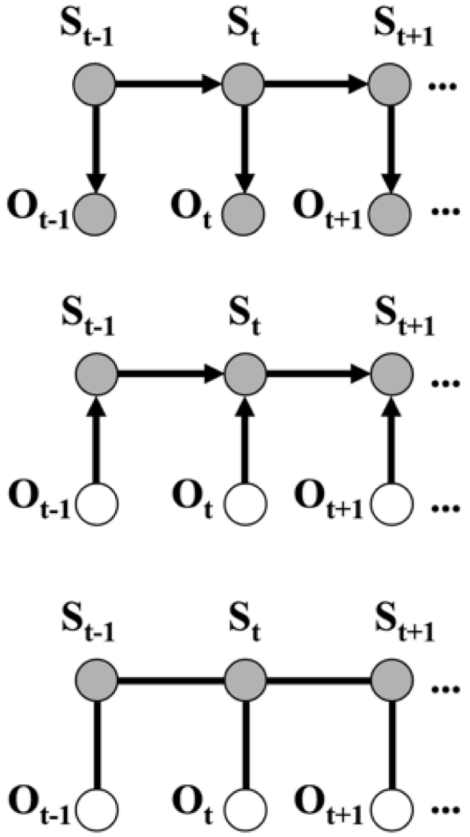


Классификация последовательностей

- Задача – поиск наиболее вероятной последовательности классов для входной последовательности
  - Например, определение частей речи в тексте

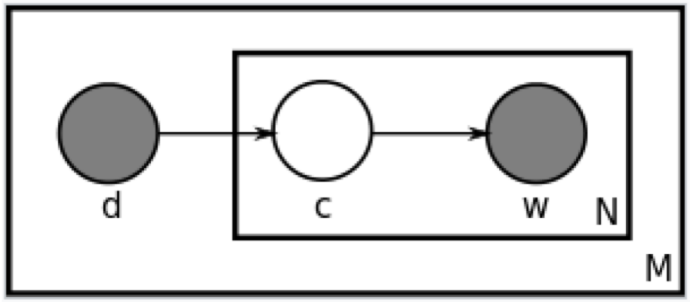
$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

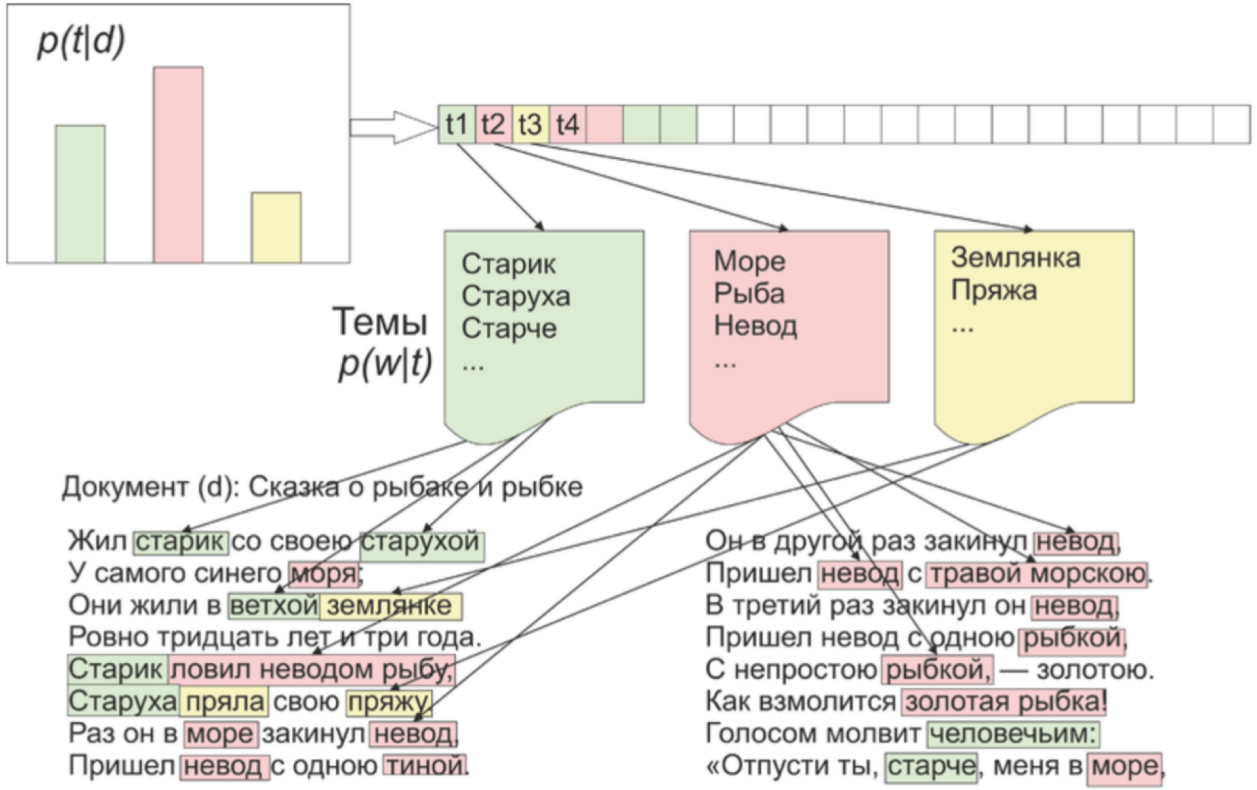
- Наиболее известные модели
  - Скрытая марковская модель (HMM)
  - Марковская модель максимальной энтропии (MEMM)
  - Условные случайные поля (CRF)



Вероятностное тематическое моделирование

- «Мягкая» кластеризация текстов
- Наиболее известные модели
  - PLSA
  - LDA
  - BigARTM



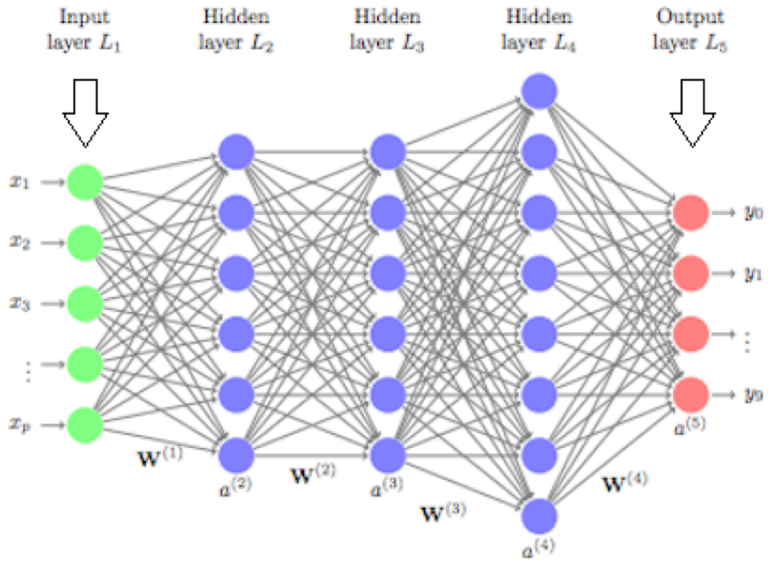


Ансамбли классификаторов

- Несколько простых (слабых) классификаторов дают более точную оценку, чем один сложный.
- **Бэггинг** – строим ансамбль независимых классификаторов, и усредняем результат (пример, Random Forest)
- **Бустинг** – строим классификаторы последовательно - учим следующий классификатор на ошибках предыдущего
  - AdaBoost
  - XGBoost, LightGBM
- **Стекинг** – выходы классификаторов могут быть входами для других классификаторов

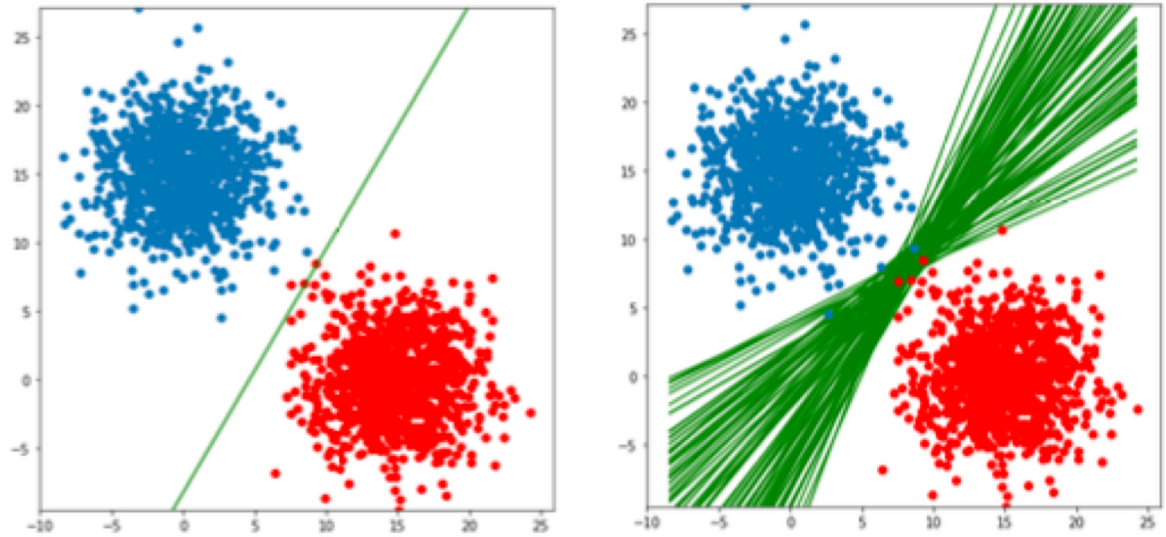
Deep learning

- Глубокое (глубинное) обучение
  - Нейронные сети со множеством слоев
  - Каждый следующий слой получает на вход результаты предыдущего



Байесовский подход к машинному обучению

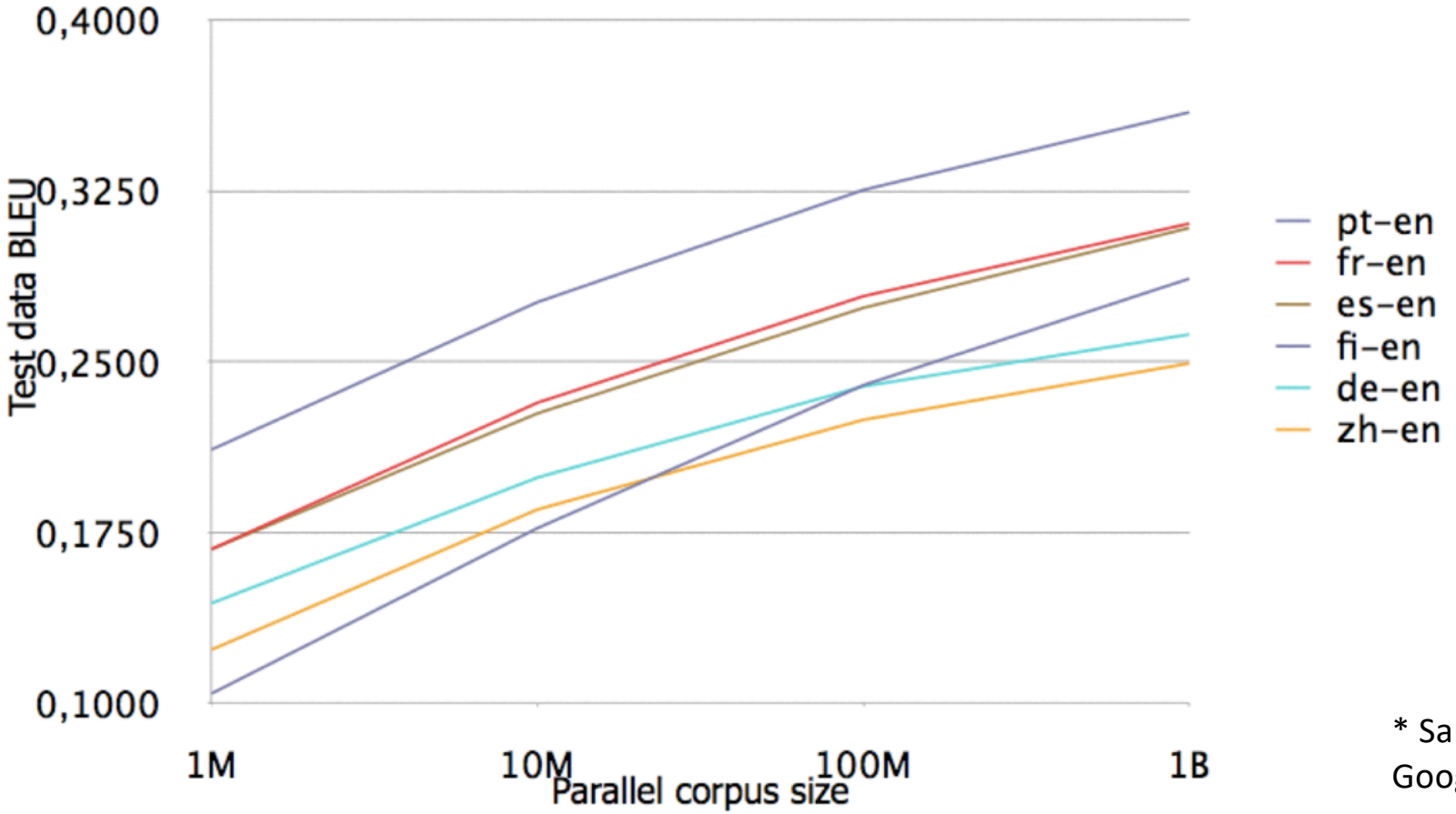
- Переход от задачи максимизации параметров к оценке их распределений



- Д.П. Ветров. Байесовские методы машинного обучения
- https://www.youtube.com/watch?v=wA8UMzhGv7o&list=PLe5rNUydzV9QHe8VDStpU0o8Yp63OecdW

Часть 4. Современные проблемы

Больше данных – лучше результат



* Salv Petrov.
Google translate

Где взять данные для обучения? (1/2)

Разметить	… долго и утомительно
Найти	Удобно, но не всегда получается
Делегировать разметку	Crowdsourcing
Размечать наиболее показательные примеры	Active Learning

Переиспользовать существующие модели	
Есть модель, решающая похожую задачу	Transfer learning
Есть модель, решающая ту же задачу, но для другой предметной области	Domain Adaptation

Где взять данные для обучения? (2/2)

Сгенерировать искусственно	
Запустить игру с нулевой суммой между двумя классификаторами: • один генерирует образцы, пытаясь обмануть второй классификатор • Другой – пытается отличить сгенерированные образцы от реальных	Generative Adversarial Networks (GAN)
Свести задачу к transfer learning, но в качестве «похожей» задачи использовать искусственно поставленную задачу на исходных данных.	Самообучение (Self-Supervision)
• Обучение представлениям (Representation learning)	Автоенкодеры Текст: word2vec и др. Графы: DeepWalk и др.
• Добавить шум и отличать зашумленные данные от реальных	Noise-contrastive estimation

Как обучить модель на доступных данных?

Есть обучающие данные для нескольких схожих задач	Многозадачное обучение (Mutitask learning)
Попытка смоделировать принципы, по которым обучаются люди	Meta-learning
• Быстрая адаптация к новым данных	Few-shot learning One-shot learning
Данные становятся доступными в последовательном порядке и используются для обновления лучшего предсказания для поступающих в будущем данных на каждом шаге	Online machine learning

Проблемы практического применения

- Интерпретируемость
- AutoML – автоматизация процесса обучения и применения алгоритмов
- Уменьшение требуемых ресурсов
  - **Прунинг** – удаление параметров, которые мало влияют на результат
  - Дистиляция – перенос знаний из сложных моделей в простые без потери качества
  - Квантизация/бинаризация – для хранения каждого параметра используем не тип double а меньшее количество бит (вплоть до одного), при этом стараемся не потерять в качестве.
- Безопасность
  - Атаки на модели
  - Воровство данных

Полезные ссылки

- https://paperswithcode.com
- http://www.machinelearning.ru
  - http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций,_К.В.Воронцов)
- https://dyakonov.org
- https://github.com/dformoso/machine-learning-mindmap/blob/master/Machine%20Learning.pdf
- https://www.coursera.org/learn/machine-learning

Следующая лекция

- Введение в нейронные сети