Введение в обработку текстов

Лекция 5

Методы классификации и кластеризации

План

- Наивный байесовский классификатор
- Линейная регрессия
- Логистическая регрессия
- Модель максимальной энтропии
- Марковская модель максимальной энтропии

Задача классификации

- Есть множество классов и множество объектов, которые могут относиться к одному или более классам.
- Задача состоит в отнесении объектов с неизвестным классом к одному или более классов
- Факторы, на основе которых делается предсказание класса, называются **признаками** (feature)
- Пример, классификация людей по расам на основе цвета кожи и формы глаз.

Модели классификации

- Производящие (наивная байесовская модель, скрытые марковские модели)
  – предполагают независимость наблюдаемых переменных
- Разделяющие (логистическая регрессия, модель максимальной энтропии, марковские модели максимальной энтропии)

Наивный байесовский классификатор

- Выбор наиболее вероятного значения

$$\hat{s} = \underset{s \in S}{\arg\max}\, P(s|f)$$

- По правилу Байеса

$$\hat{s} = \underset{s \in S}{\arg\max} \frac{P(s)P(f|s)}{P(f)} = \underset{s \in S}{\arg\max}\, P(s)P(f|s)$$

- Наивное предположение об условной независимости признаков

$$\hat{s} = \underset{s \in S}{\arg\max}\, P(s) \prod_{j=1}^{n} P(f_i|s)$$

Обучение наивного байесовского классификатора

- Метод максимального правдоподобия
- Другими словам, просто считаем

$$P(s) = \frac{count(s)}{\sum_{s \in S} count(s_i)} \quad P(f_j|s) = \frac{count(f_j, s)}{count(s)}$$

- Алгоритм прост в реализации, но
  - Исчезновение значащих цифр → использовать сумму логарифмов вместо произведения
  - Нулевые вероятность → сглаживание или предположение о распределении $P(f_j|s)$

Пример

```
from sklearn.nayve_bayes import *

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

# initialize classifier
classifier = MultinominalNB()
```

```
# 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)
```

Модель максимальной энтропии

- Мультиномиальная логистическая регрессия
- Модель классификации вида

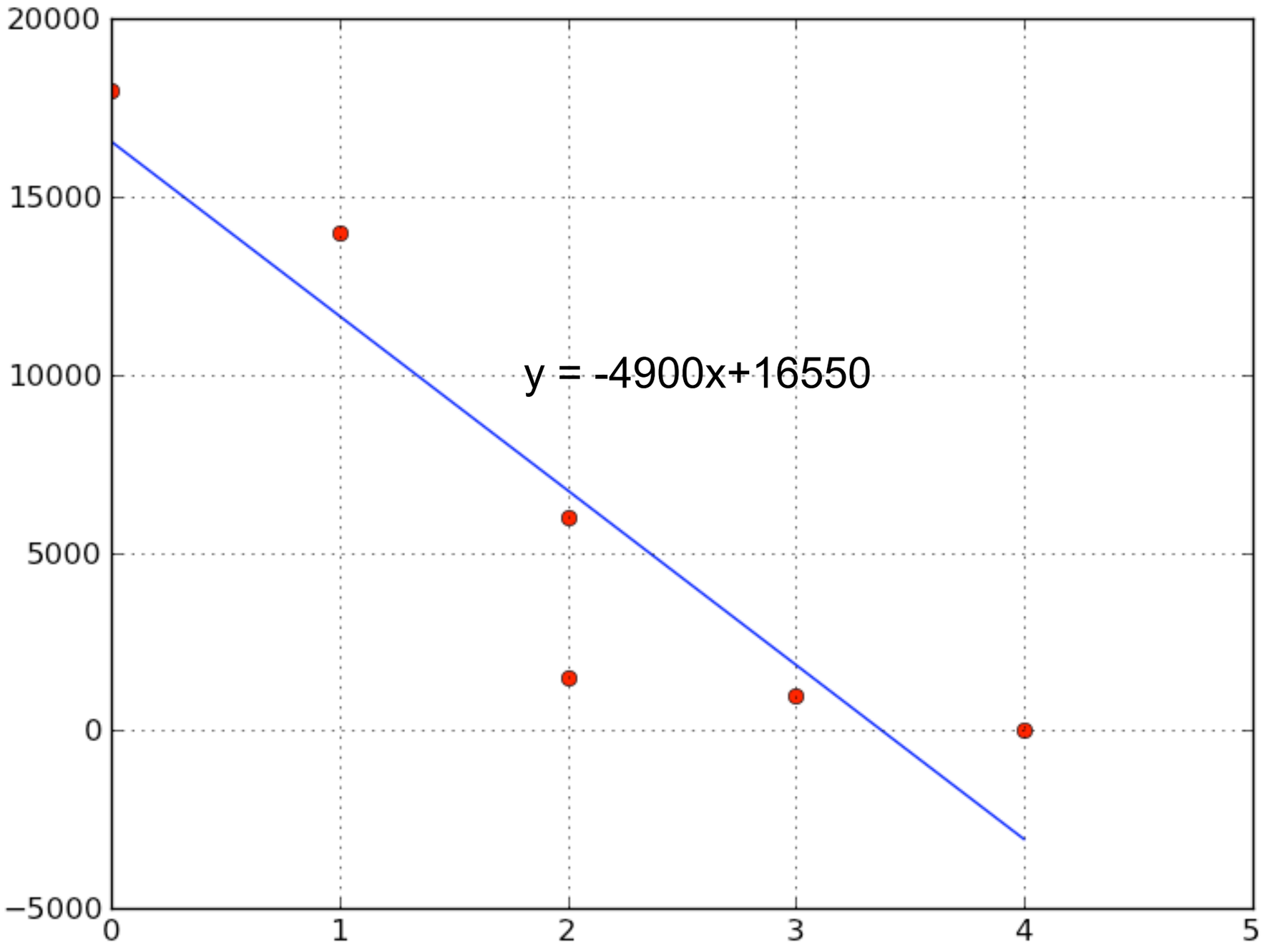
$$p(c|x) = \frac{1}{Z} exp(\sum_i w_i f_i)$$

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 $f_0 = 1$

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

Вычисление коэффициентов

- Минимизировать квадратичную погрешность

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

- 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

Мультиномиальная логистическая регрессия

- Классификация на множество классов

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

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

Признаки

- Принято использовать бинарные признаки
- Индикаторная функция зависящая от класса и наблюдения
- Пример

$$f_1(c, x) = \begin{cases} 1 \text{ if suffix}(word_i) = \text{”ing”} \ \& \ \text{c=VBG} \\ 0 \end{cases}$$

$$f_2(c, x) = \begin{cases} 1 \text{ if } word_i = \text{”race”} \ \& \ \text{c=NN} \\ 0 \end{cases}$$

Пример

		f1	f2	f3	f4	f5	f6
VB	f	0	1	0	1	1	0
	w		0.8		0.01	0.1	
NN	f	1	0	0	0	0	1
	w	0.8					-1.3

$$p(NN|x) = \frac{e^{0.8}e^{-1.3}}{e^{0.8}e^{-1.3} + e^{0.8}e^{0.01}e^{0.1}} = 0.2$$

$$p(VB|x) = \frac{e^{0.8}e^{0.01}e^{0.1}}{e^{0.8}e^{-1.3} + e^{0.8}e^{0.01}e^{0.1}} = 0.8$$

Обучение модели

- Найти параметры, которые максимизируют логарифмическое правдоподобие на тренировочном наборе

$$\hat{w} = \arg\max_{w} \sum_{i} logP(y^i|x^i) - \sum_{j=1}^{N} \frac{w_j^2}{2\sigma_j^2}$$

- Используются методы выпуклой оптимизации
- Такой способ позволяет из всех моделей, удовлетворяющих ограничениям тестовой выборки, выбрать модель с максимальной энтропией (Berger et. al. 1996)

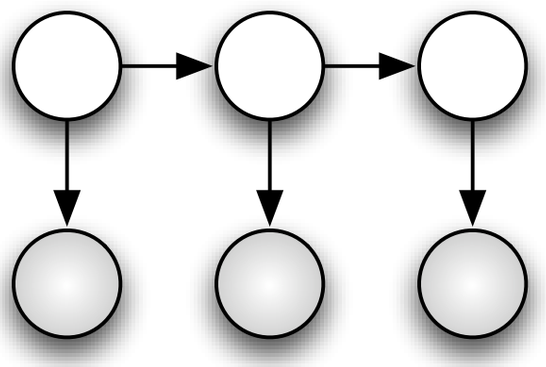
Марковская модель максимальной энтропии

- Позволяет смоделировать сложные признаки (например для определения части речи)

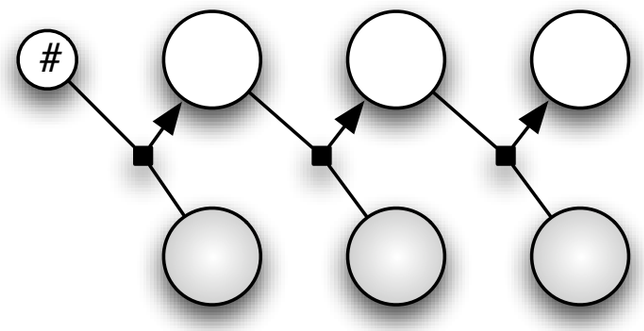
$$\hat{T} = \arg\max_t P(T|W) = \arg\max_T \prod_i P(tag_i|word_i, tag_{i-1})$$

- Сравнить с марковской моделью

$$\hat{T} = \arg\max_t P(T|W) = \arg\max_T \prod_i P(word_i|tag_i)P(tag_i, tag_{i-1})$$

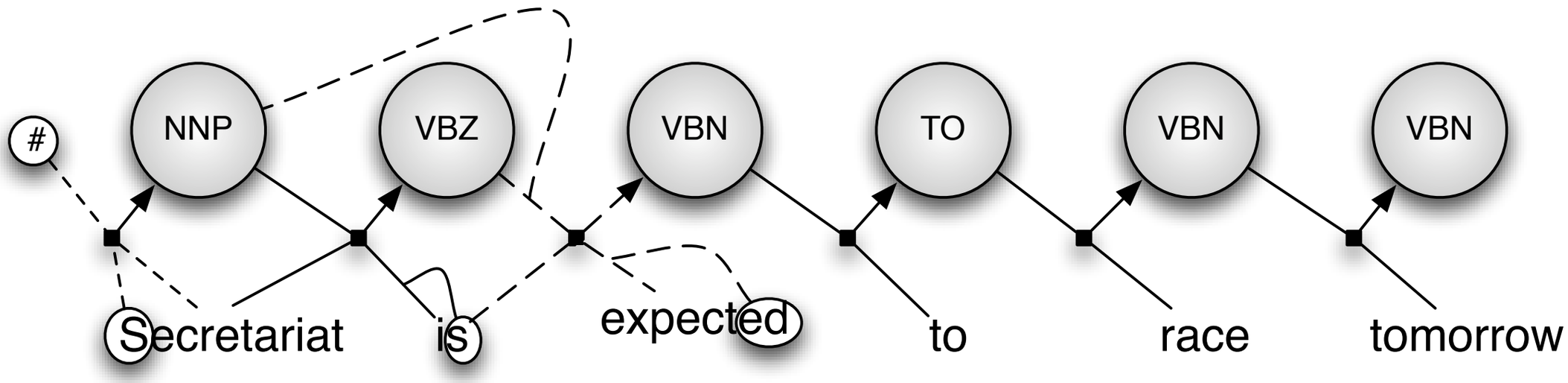


Скрытые марковские модели



Скрытые марковские модели максимальной энтропии

Признаки в MEMM



$$P(q|q', o) = \frac{1}{Z(o, q')} exp\left(\sum_i w_i f_i(o, q)\right)$$

Декодирование и обучение

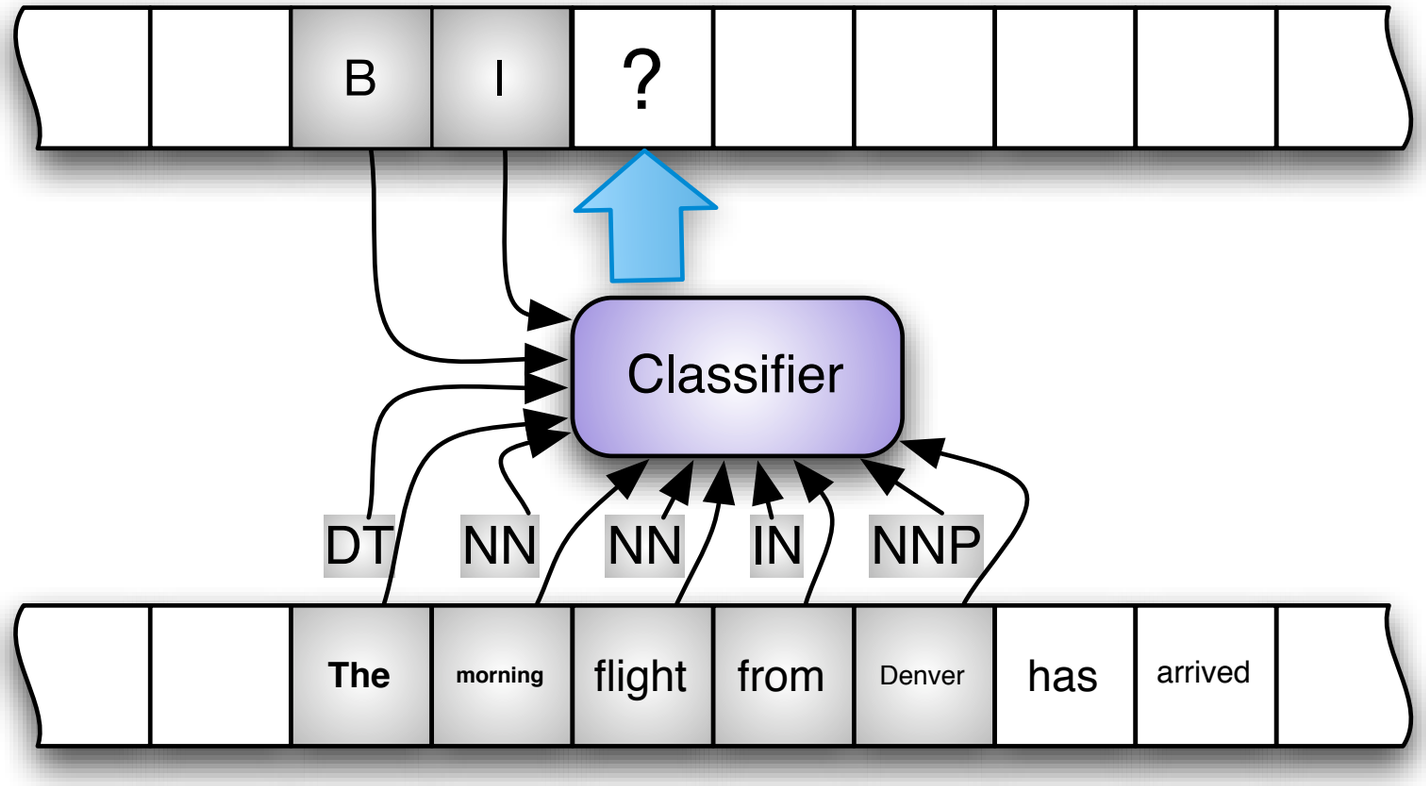
- Декодирование - алгоритм Витерби, где на каждом шаге вычисляется

$$v_t(j) = max_{i=1}^{N} v_{t-1}(i) P(s_j|s_i, o_t), 1 \leq j \leq N, 1 < t \leq T$$

- Обучение аналогично логистической регрессии

Группировка при классификации последовательностей

- Классы + метки
  - IO (inside, outside)
  - BIO (begin, inside, outside) - стандарт
  - BMEWO (begin, middle, end, whole, outside)

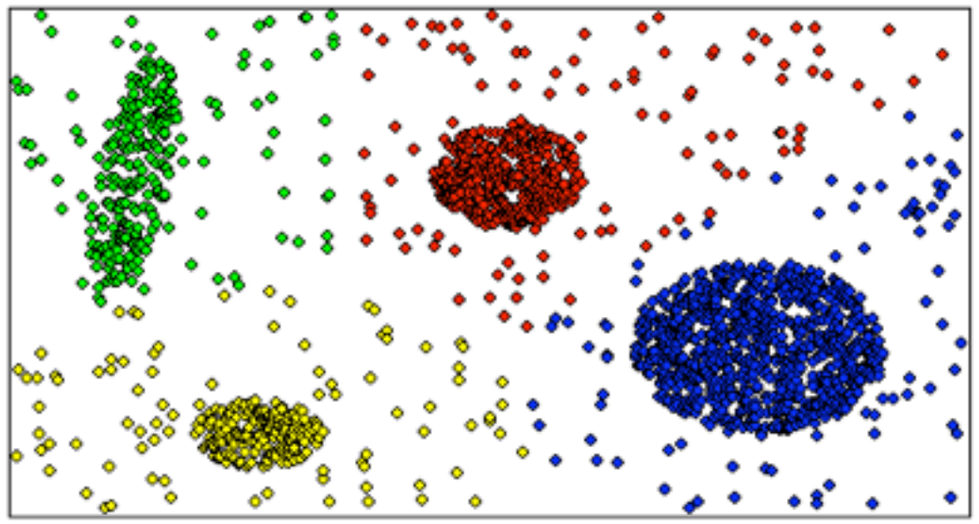


Кластеризация

обучение без учителя

Мотивация

- Данные можно разбить на несколько групп по принципу схожести



- Поиск схожих документов
- Поиск схожих слов и терминов
- Реферирование документов
- Для задач обучения с учителем
  - –Кластер, как признак для обучения
  - –Кластер, как набор данных для обучения

Вход для алгоритмов

- Пусть каждый документ $\{x_1, x_2, \ldots, x_k\}$ представлен вектором $x_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ 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 $\rho(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$
  - ...

Векторное представление документа

- Модель мешка слов (bag of words)
- Вес слова
  - Частота слова в документе (tf)
  - TF-IDF

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

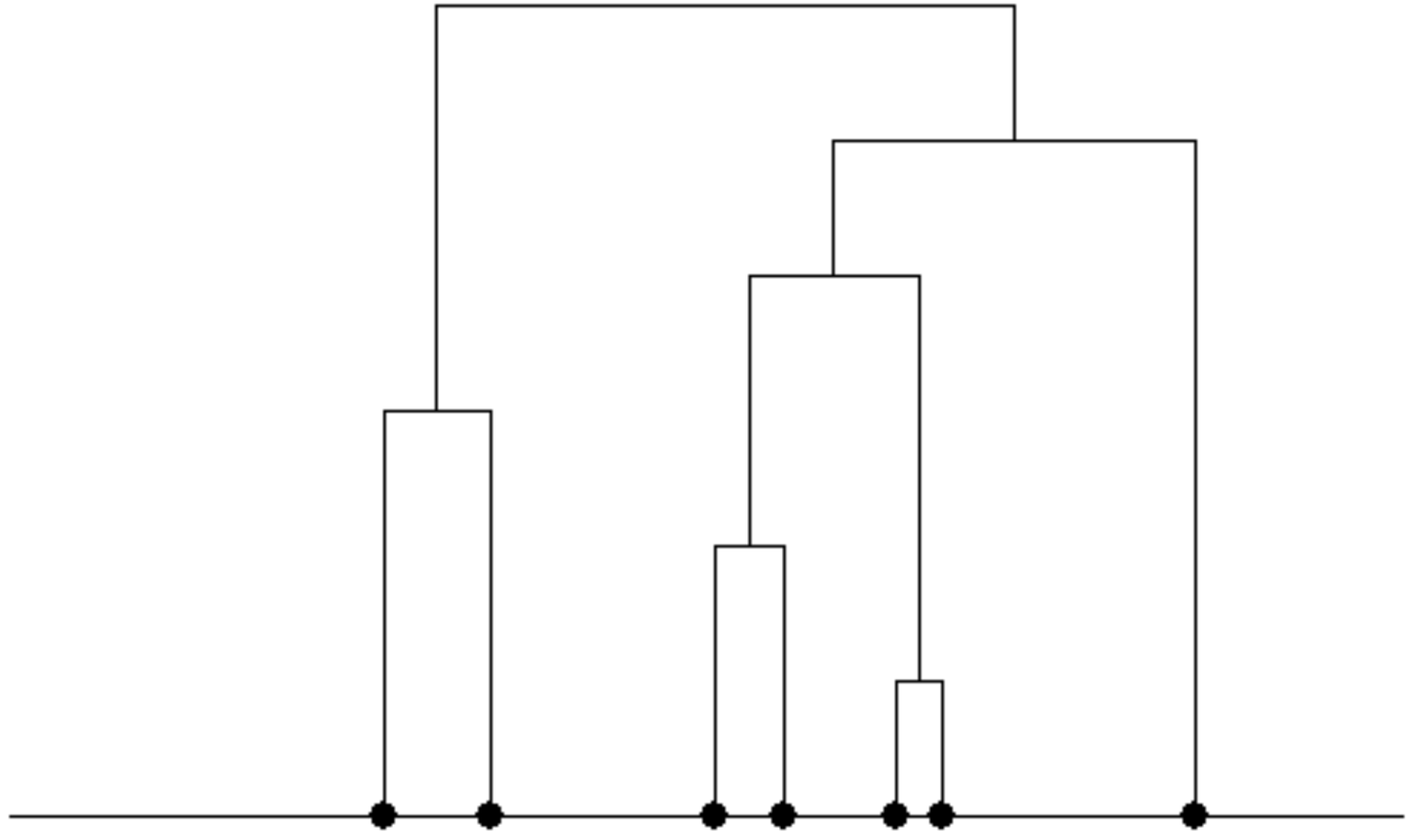
  - Вложение слов (Word Embedding): word2vec, GloVe и т. д.

План

- Иерархическая кластеризация
- k-means
- Affinity propogation
- MeanShift
- Спектральная кластеризация
- WARD
- DBSCAN

Иерархическая кластеризация

- Строится дендрограмма - дерево обозначающее вложенную последовательность кластеров

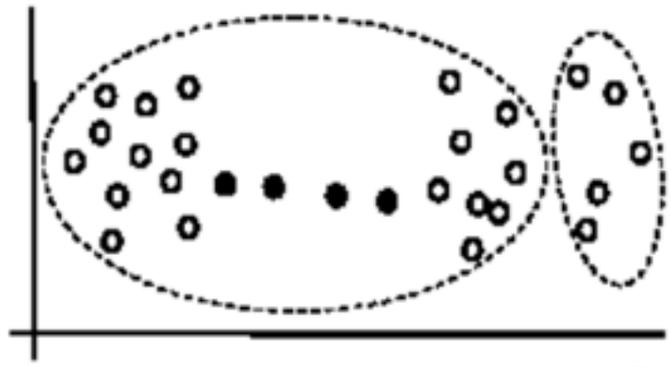


Типы иерархической кластеризации

- Агломеративная
  - каждая точка - кластер
  - объединяем два наиболее близких кластера в один
  - останавливаемся, когда все данные объединены в один кластер
- Дивизимная
  - все данные - один кластер
  - разделяем наименее плотный кластер на два
  - останавливаемся, когда достигли минимального допустимого размера

Расстояние между кластерами

- Между двумя ближайшими точками
  - –Можно получить кластеры произвольной формы
  - –“Эффект цепи”



- Между двумя самыми дальними точками
  - –Чувствителен к выбросам
- Среднее расстояние

K-средних

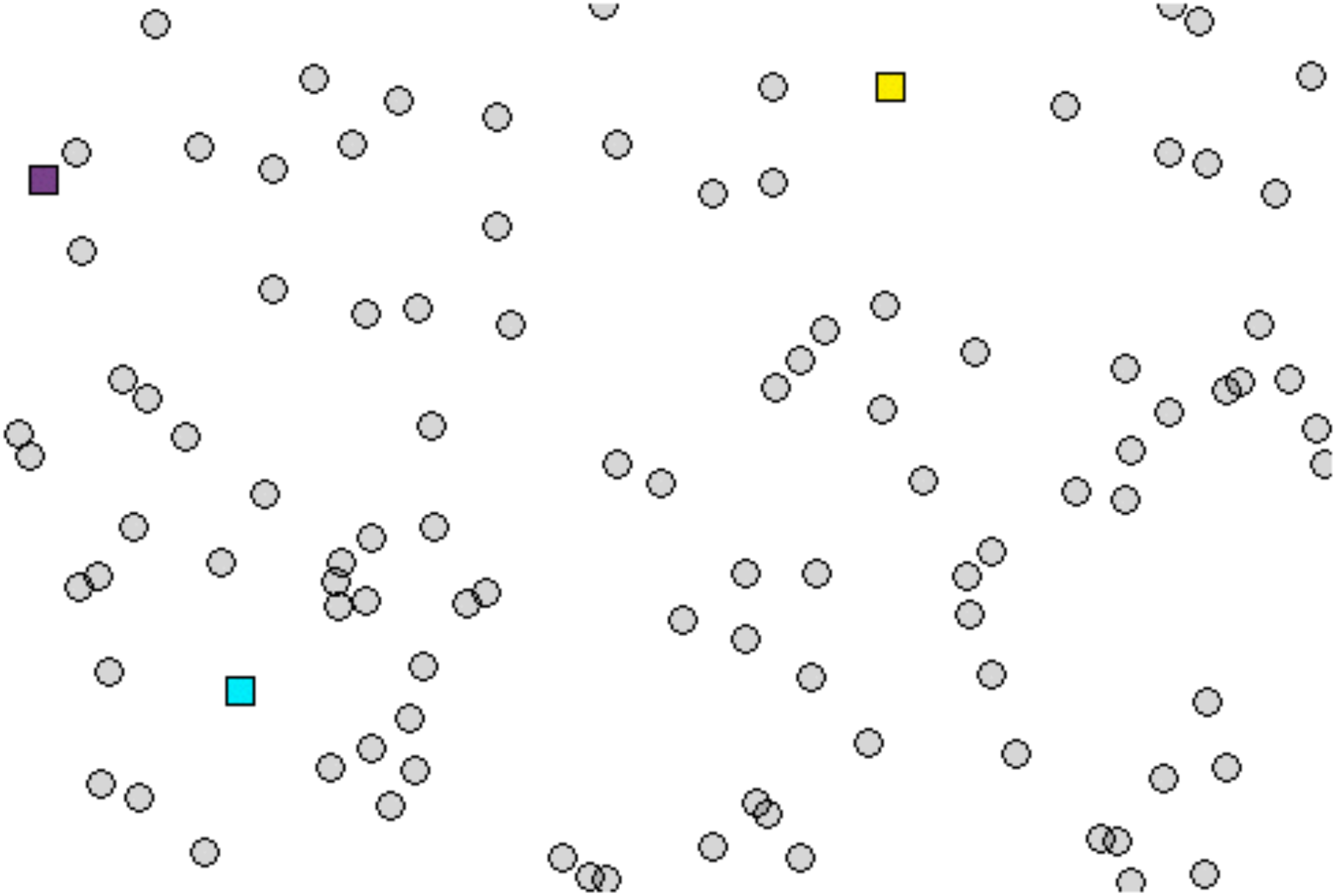
- Алгоритм k-means разбивает данные на k кластеров
  - –Каждый кластер имеет центр - центроид
  - –Параметр k - задается вручную
- Алгоритм
  1. Выбираются k точек в качестве начальных центроидов
  2. Сопоставить каждой точке ближайший центроид
  3. Пересчитать центроиды
  4. Если алгоритм не сошелся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

Критерий останова

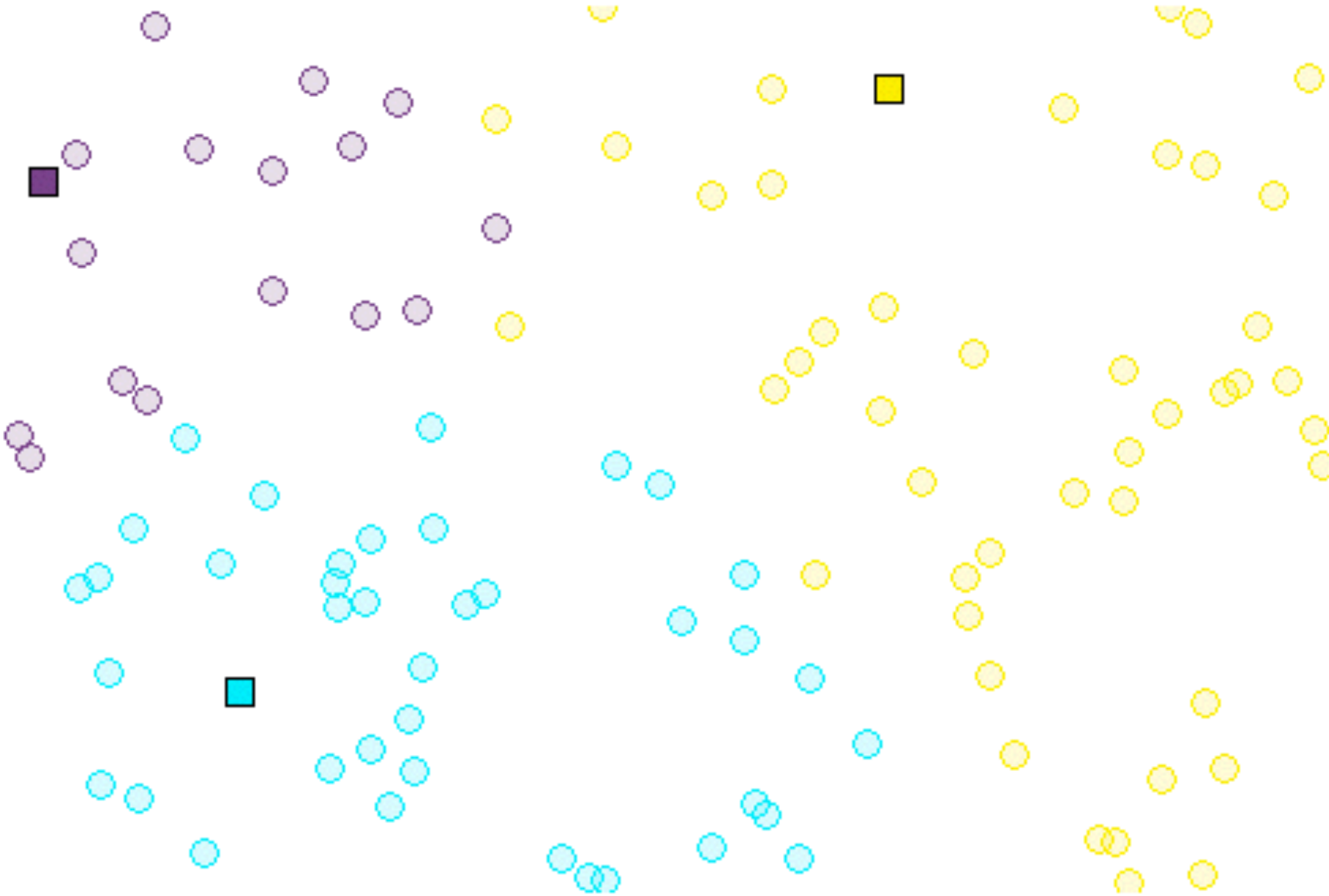
- Нет перехода точек в другой кластер
- Нет (незначительно) изменение центроидов
- Мало убывает погрешность (sum of squared error)

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

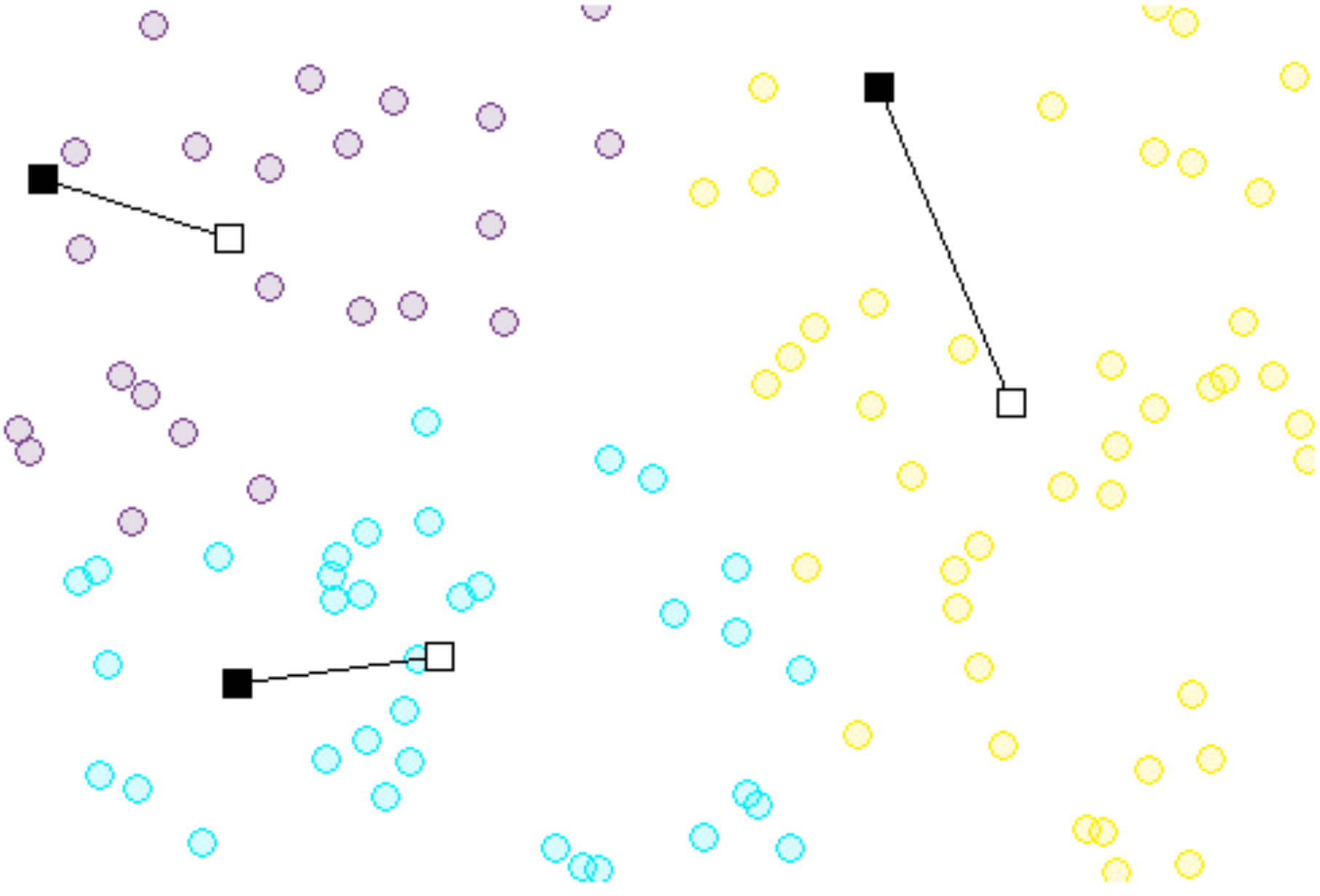
K-средних. Пример



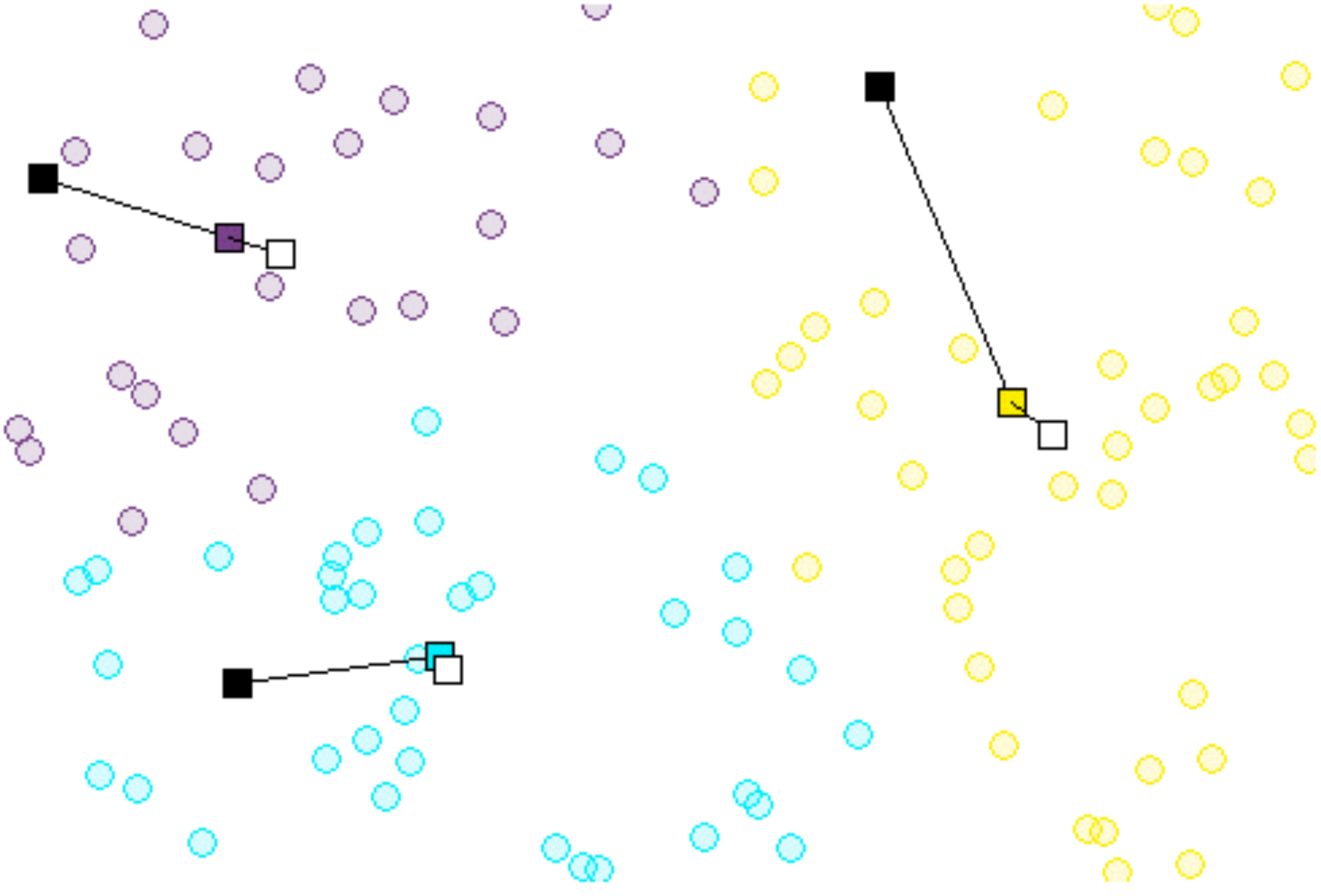
K-средних. Пример



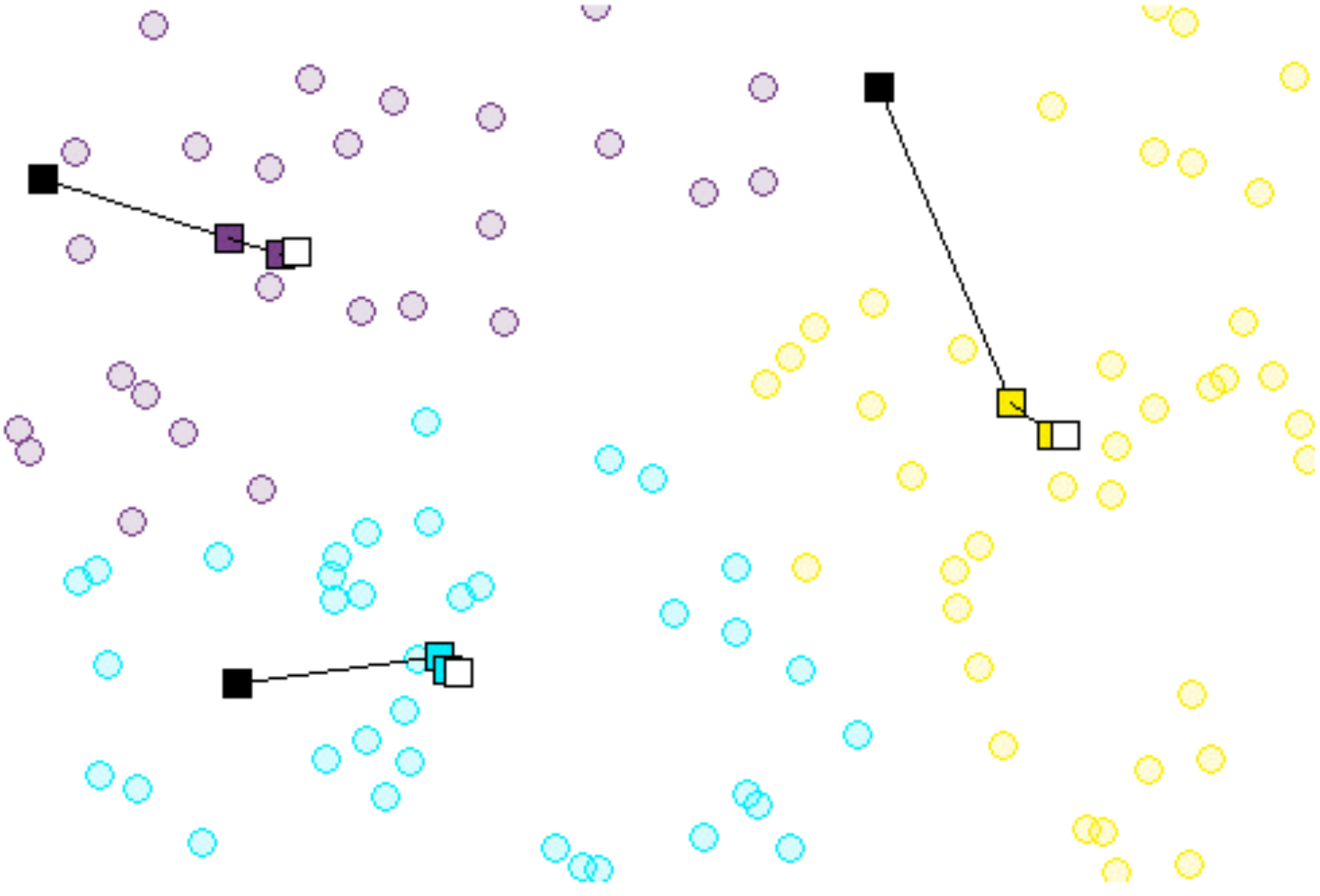
K-средних. Пример



K-средних. Пример



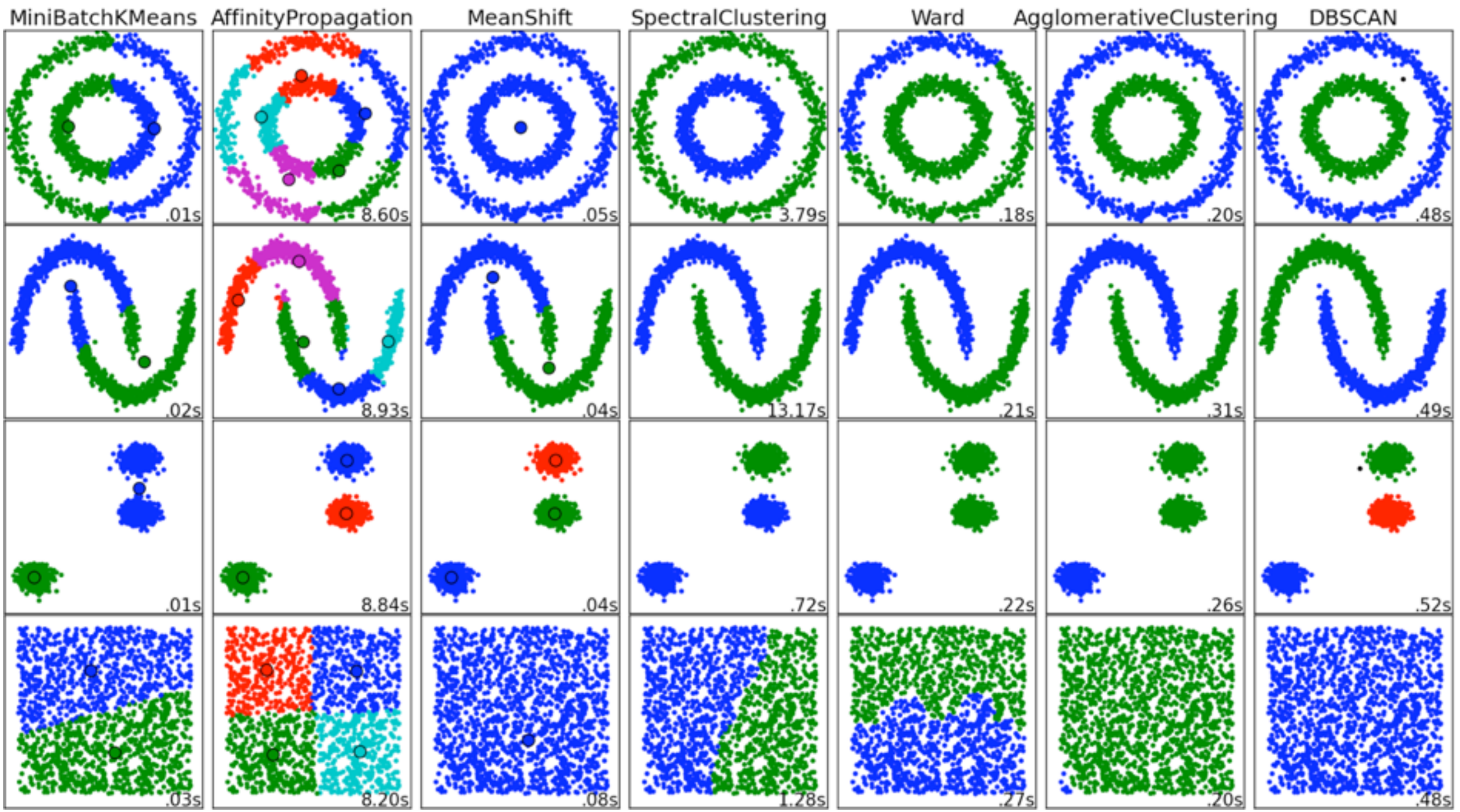
K-средних. Пример



Проблемы

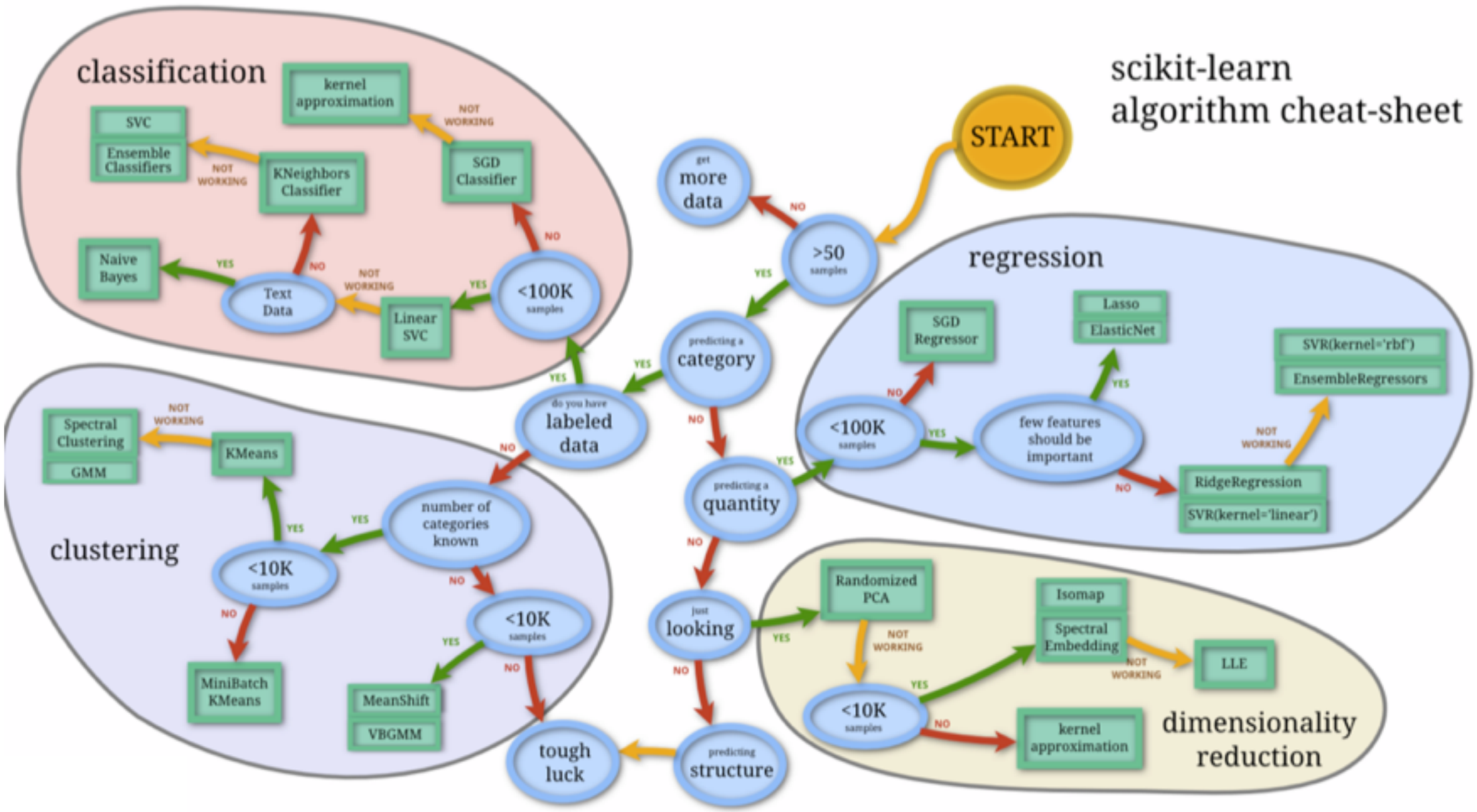
- Алгоритм чувствителен к начальному выбору центроидов
  - –запуск с различной начальной инициализацией и выбор варианта с наиболее плотными кластерами
- Чувствителен к выбросам
  - –можно фильтровать выбросы
- Не подходит для нахождения кластеров, не являющихся элипсоидами
  - –преобразование пространства

Какой алгоритм выбрать



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

Что делать



*http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Следующая лекция

- Методы классификации текстов на основе искусственных нейронных сетей