Big Data'13

Лекция X: алгоритмы кластеризации

Дмитрий Барашев
bigdata@barashev.net

Computer Science Center

25 апреля 2013

Сегодня в программе

- Задача кластеризации
- Методы кластеризации
- Иерархическая кластеризация
- Алгоритм k-средних
- Зонтичная кластеризация

Сегодня в программе

- Задача кластеризации
- Методы кластеризации
- Иерархическая кластеризация
- Алгоритм k-средних
- Зонтичная кластеризация

Что хотим

- Дано
  - некоторый набор точек
  - функция расстояния между точками
- Что хотим
  - сгруппировать точки в некоторое количество *кластеров* по каким-то правилам

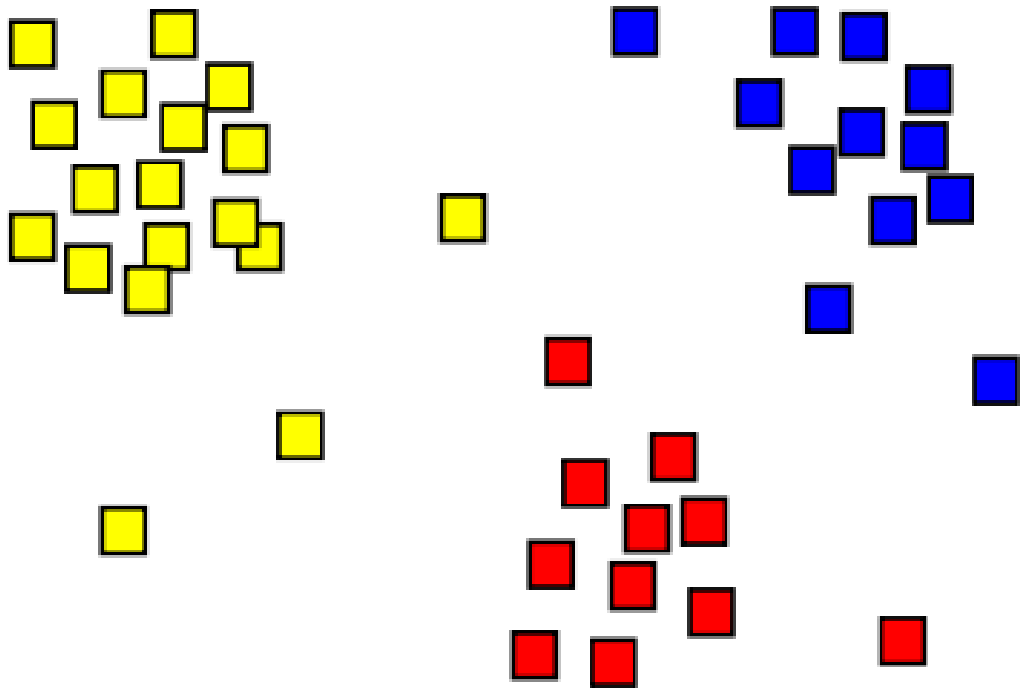
Что хотим

- Дано
  - некоторый набор точек
  - функция расстояния между точками
- Что хотим
  - сгруппировать точки в некоторое количество *кластеров* по каким-то правилам
- Сколько кластеров?
- Какие правила?
- Какая функция расстояния?

Точки и расстояние

- Обычно точки в евклидовом многомерном пространстве
- А может и в неевклидовом
  - в евклидовом пространстве обычно можно найти некую ”среднюю” точку, возможно отсутствующую в исходном множестве
- Функция расстояния – Евклидово, косинусное, Жаккардово, Левенштейна

Четкие кластеры



Приложения

- Кластеризация результатов поиска

Приложения

- Кластеризация результатов поиска
- Последовательная автоматическая рекластеризация документов
  - **”Россия”**, ”политика”, ”экономика”, **”спорт”** $\rightarrow$ ”Россия”, **”футбол”**, ”хоккей”, **”лига чемпионов”**, ”Гагарин” $\rightarrow$ ”Барселона”, ”Германия”, ”полуфиналы”, ~~”какая боль”~~

Приложения

- Кластеризация результатов поиска
- Последовательная автоматическая рекластеризация документов
  - **”Россия”**, ”политика”, ”экономика”, **”спорт”** $\rightarrow$ ”Россия”, **”футбол”**, ”хоккей”, **”лига чемпионов”**, ”Гагарин” $\rightarrow$ ”Барселона”, ”Германия”, ”полуфиналы”, ~~”какая боль”~~
- Поиск тесно связанных между собой пользователей контактика

Приложения

- Кластеризация результатов поиска
- Последовательная автоматическая рекластеризация документов
  - **”Россия”**, ”политика”, ”экономика”, **”спорт”** $\rightarrow$ ”Россия”, **”футбол”**, ”хоккей”, **”лига чемпионов”**, ”Гагарин” $\rightarrow$ ”Барселона”, ”Германия”, ”полуфиналы”, ~~”какая боль”~~
- Поиск тесно связанных между собой пользователей контактика
- Поиск научных статей, похожих по темам на вашу

Приложения

- Кластеризация результатов поиска
- Последовательная автоматическая рекластеризация документов
  - **”Россия”**, ”политика”, ”экономика”, **”спорт”** $\rightarrow$ ”Россия”, **”футбол”**, ”хоккей”, **”лига чемпионов”**, ”Гагарин” $\rightarrow$ ”Барселона”, ”Германия”, ”полуфиналы”, ~~”какая боль”~~
- Поиск тесно связанных между собой пользователей контактика
- Поиск научных статей, похожих по темам на вашу
- Определение оптимального местоположения базовых станций сотовой сети

Евклидово расстояние

$$x = [x_1, \ldots, x_n], y = [y_1, \ldots, y_n]$$

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Косинусное расстояние

$$x = [x_1, \ldots, x_n], y = [y_1, \ldots, y_n]$$

$$d(x, y) = \frac{\sum_{i=1}^{n}(x_i \times y_i)}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Жаккардово расстояние

- Коэффициент схожести $J(A,B) = \frac{A \cap B}{A \cup B}$
- Расстояние $d(A,B) = 1 - J(A,B)$

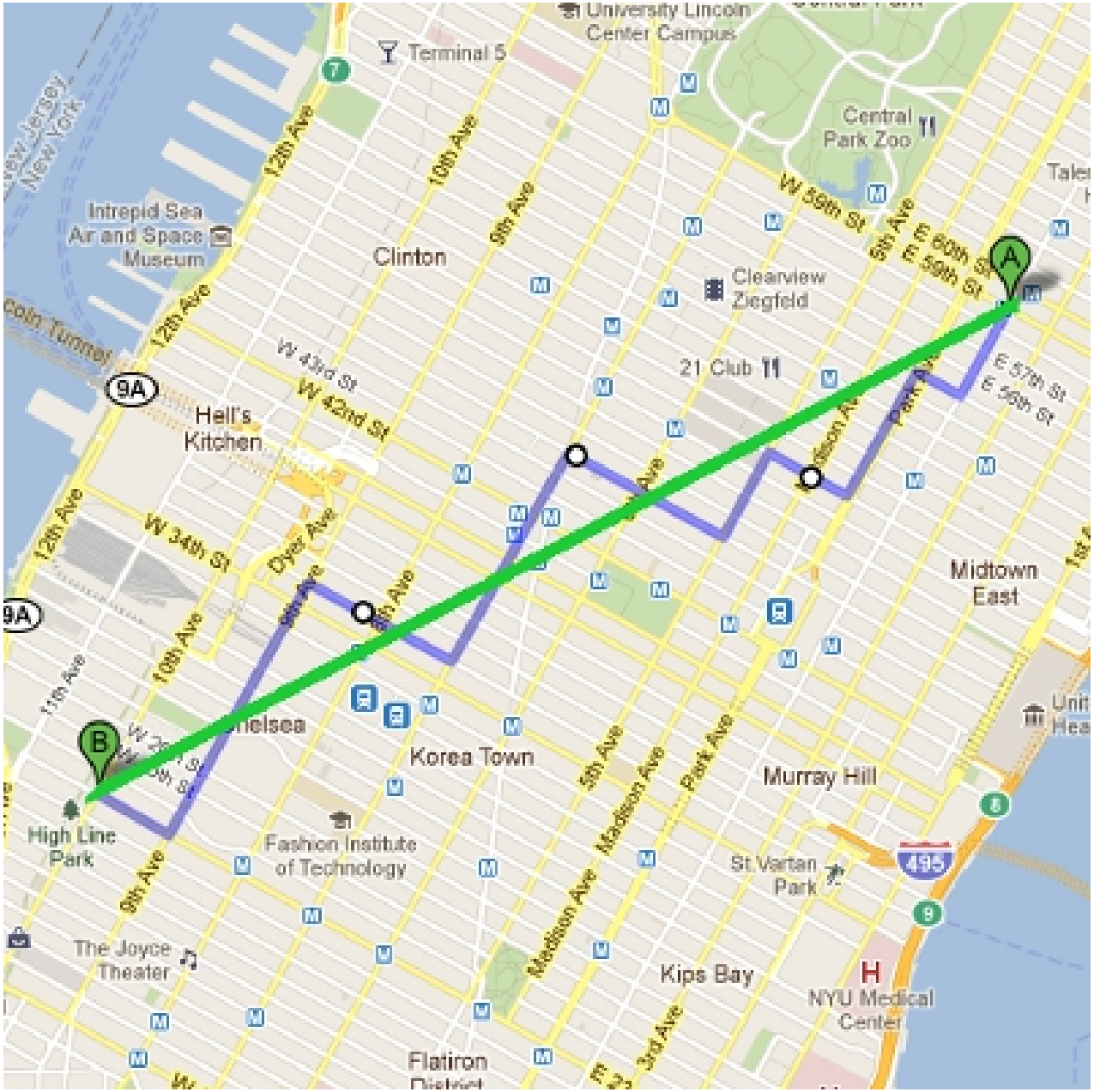
Манхеттенское расстояние

$$x = [x_1, \ldots, x_n], y = [y_1, \ldots, y_n]$$

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- Сколько кварталов нужно пройти по Манхеттену чтоб попасть с одного перекрестка на другой

Манхеттенское расстояние



Сегодня в программе

Задача кластеризации

Методы кластеризации

Иерархическая кластеризация

Алгоритм k-средних

Зонтичная кластеризация

Две разные стратегии

- Иерархическая (восходящая и нисходящая)
- Плоская

Иерархическая стратегия

- Сначала каждая точка – один кластер
- На итерации «ближайшие» кластеры объединяются
- Итерации останавливаются при достижении какого-то критерия
- В нисходящей стратегии все наоборот, начинается с одного большого кластера, включающего все точки

Плоская стратегия

- Определяются первоначальные кластеры
- Точки рассматриваются по очереди и приписываются какому-то кластеру
- Если результат достаточно хорош, итерации останавливаются

Возможные характеристики кластеров

- **Диаметр**: максимальное расстояние между любыми двумя точками в кластере
- **Радиус**: максимальное расстояние от некого «центра» до любой из точек кластера
- **Плотность**: количество точек в кластере поделенное на «объём»: радиус в какой-то степени
- **Межкластерное расстояние**: расстояние между центрами, между ближайшими точками, среднее расстояние между всеми парами

Критерии прекращения кластеризации

- Построено нужное число кластеров
- Характеристики кластеров (диаметр, плотность) достигли граничных значений

Центры кластеров

- В евклидовом пространстве есть *центроид* – среднем арифметическом точек кластера
- В неевклидовом пространстве (например в пространстве строк) центроида нет. Центром (*кластроидом*) выбирается одна из точек кластера, минимизирующая
  - максимальное расстояние до остальных точек
  - или сумму растояний
  - или сумму квадратов расстояний

Сегодня в программе

Задача кластеризации

Методы кластеризации

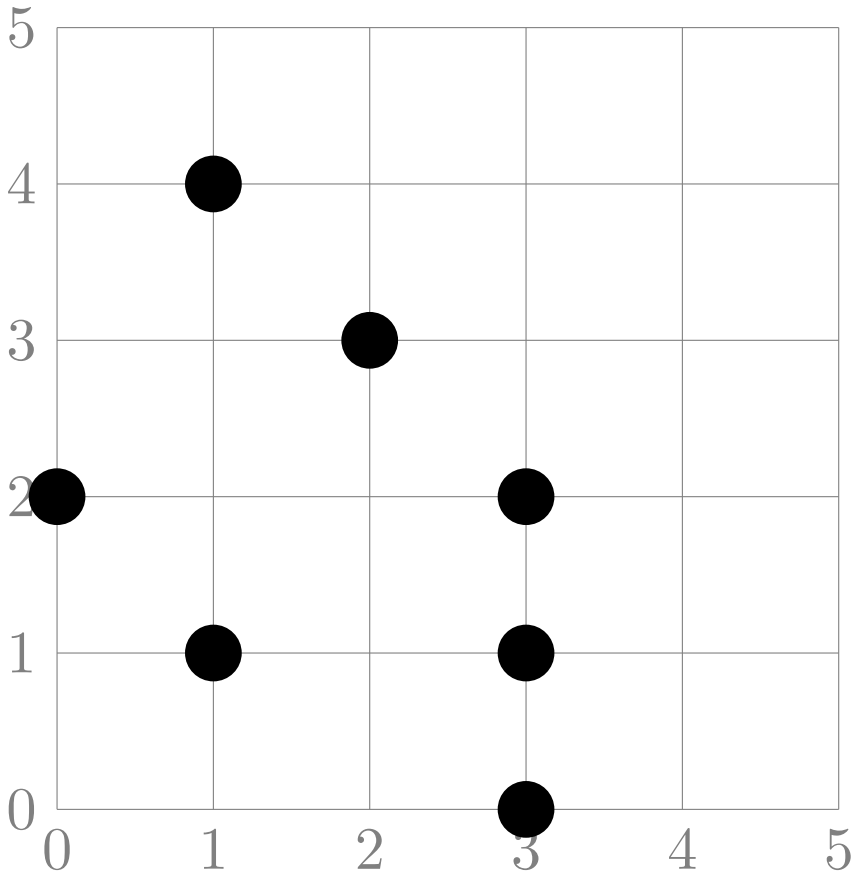
Иерархическая кластеризация

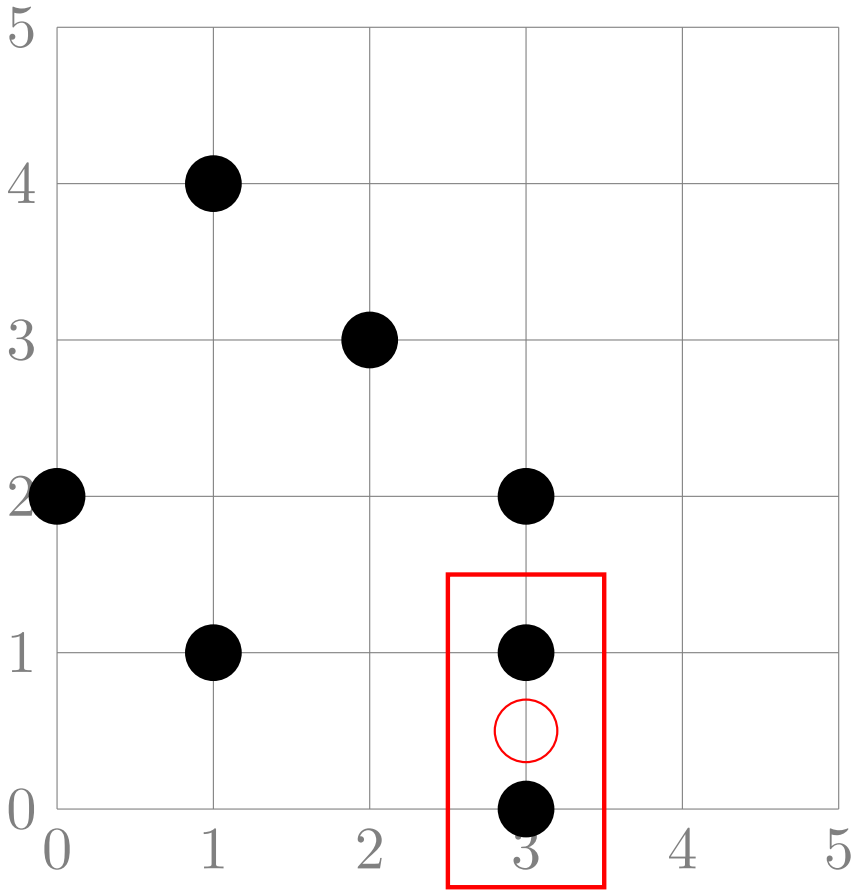
Алгоритм k-средних

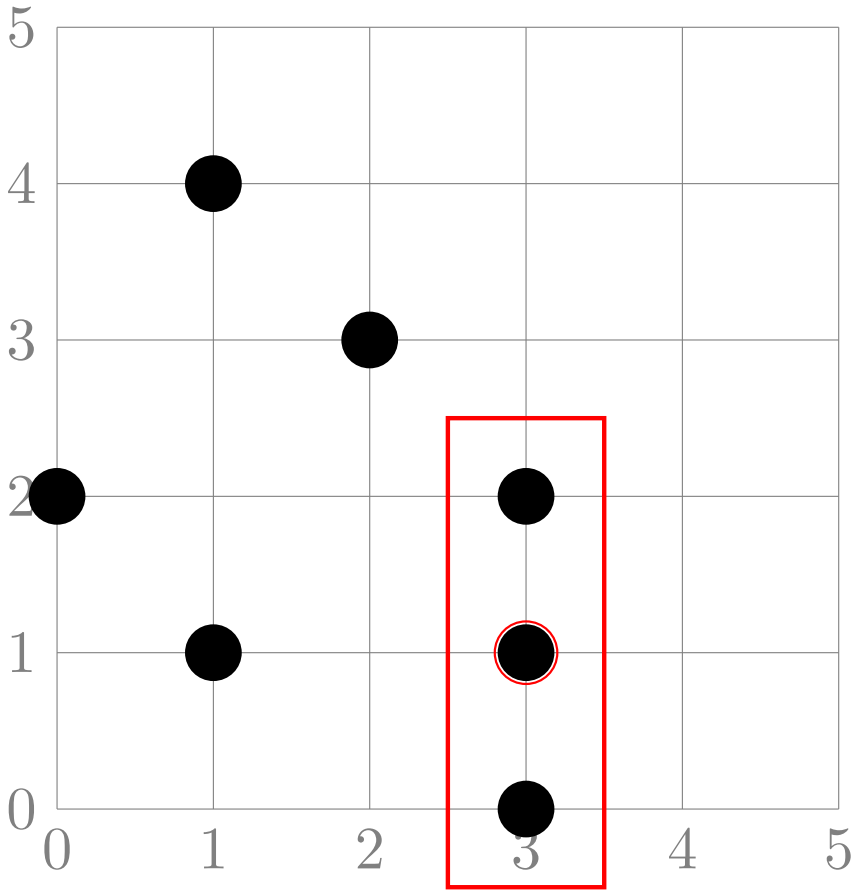
Зонтичная кластеризация

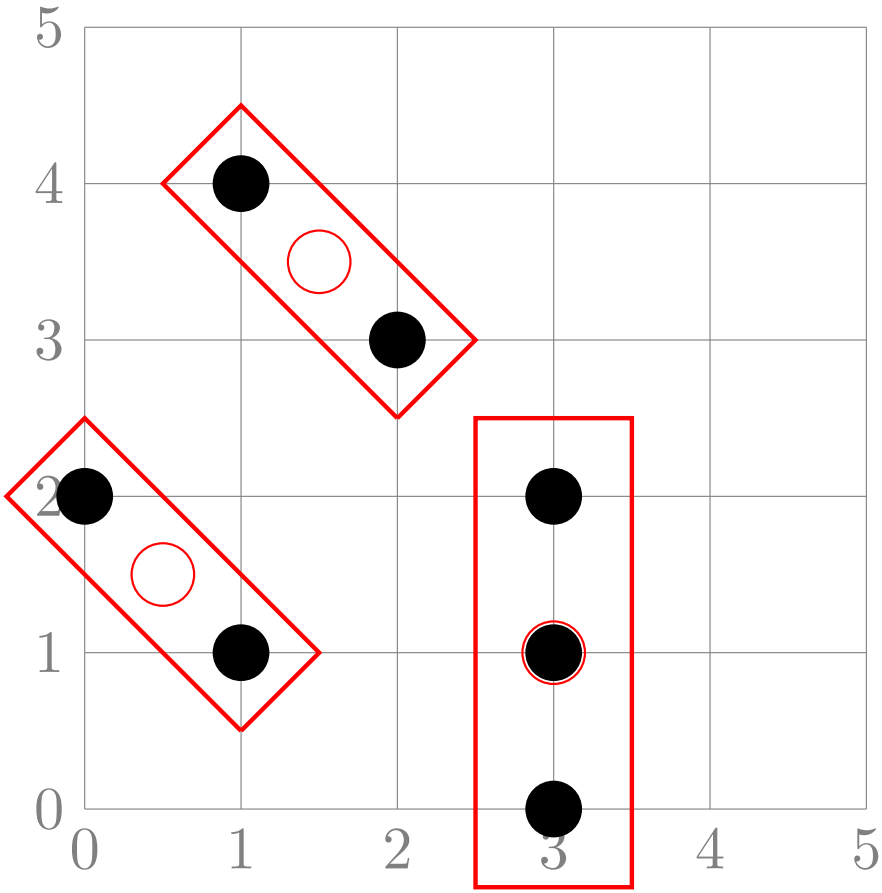
Восходящая иерархическая кластеризация

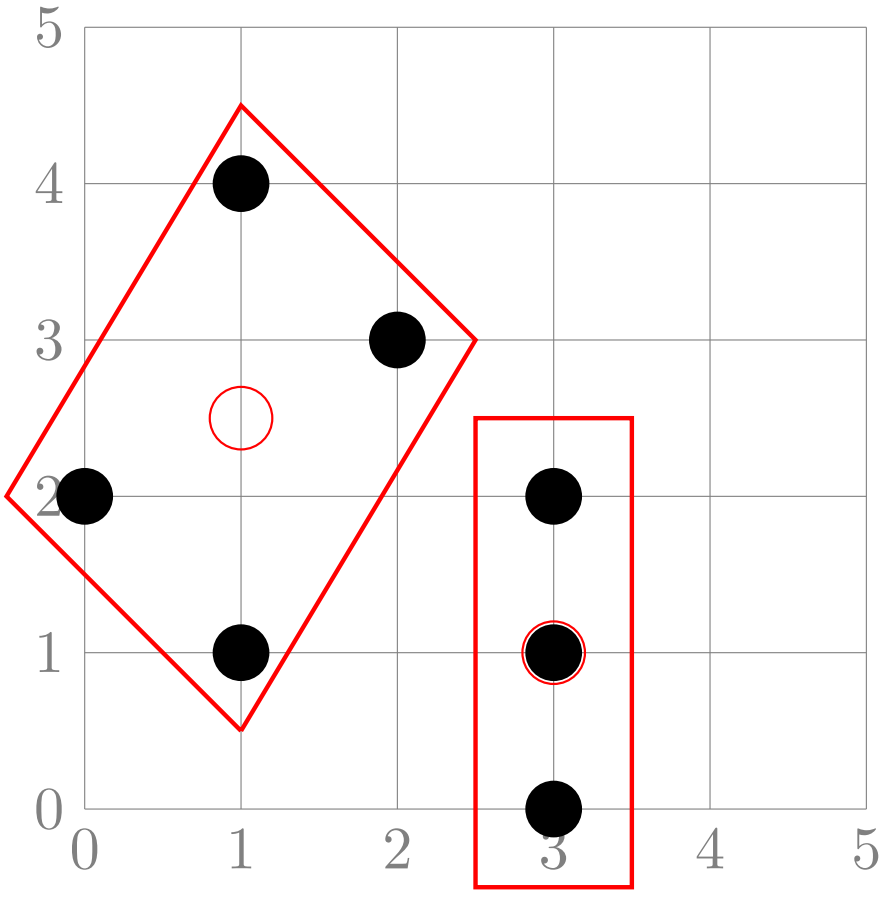
- Евклидово пространство
- Исходное положение: каждая точка - один кластер
- Итерация: объединяются два самых близких кластера


5
4
3
2
1
0
0
1
2
3
4
5


5
4
3
2
1
0
0
1
2
3
4
5


5
4
3
2
1
0
0
1
2
3
4
5


5
4
3
2
1
0
0
1
2
3
4
5


5
4
3
2
1
0
0
1
2
3
4
5

Сегодня в программе

Задача кластеризации

Методы кластеризации

Иерархическая кластеризация

Алгоритм k-средних

Зонтичная кластеризация

Схема

- Выбрать k точек, находящиеся вероятно в разных кластерах и объявить их центроидами
- Итерация: для каждой точки найти ближайший центроид и отнести её к соотв. кластеру
- Пересчитать центроиды и если требуется, сделать следующую итерацию
  - например если ни одна точка не переехала в другой кластер то уже наверное хватит

Map-Reduce реализация

- Подготовка: заготовить список центроидов $[c_i]_{i=1}^{k}$. Он скорее всего поместится в RAM
- Map: для каждой точки p прочитать центроиды, найти ближайший c_j, выплюнуть пару (c_j, p)
- Reduce: для полученного кластера, представленного центроидом c_i (ключ свертки) и списка точек найти новый центроид и посчитать характеристики кластера. Результат записать.

Снова о прекращении итераций

- Среднеквадратическая ошибка
- Для одного кластера

$$D_i = \sum_{x \in \omega_i} |x - c_i|^2$$

- Для всего множества

$$D = \sum_{i=1}^{k} D_i$$

Снова о прекращении итераций

- Среднеквадратическая ошибка
- Для одного кластера

$$D_i = \sum_{x \in \omega_i} |x - c_i|^2$$

- Для всего множества

$$D = \sum_{i=1}^{k} D_i$$

- Алгоритм k-means находит (локальный) минимум D
  - можно останавливаться, если D стал ниже ~~плинтуса~~ порога
  - или если изменение D стало маленьким

Выбор k

- Если число кластеров неизвестно априорно, то можно перебрать разные k
- Если поставить цель минимизировать среднеквадратичную ошибку то лучшее k

Выбор k

- Если число кластеров неизвестно априорно, то можно перебрать разные k
- Если поставить цель минимизировать среднеквадратичную ошибку то лучшее k
  - $k = N$

Выбор k

- Если число кластеров неизвестно априорно, то можно перебрать разные k
- Если поставить цель минимизировать среднеквадратичную ошибку то лучшее k
  - $k = N$
- Можно рассмотреть кривую уменьшения D в зависимости от увеличения k и брать k соответствующий точкам изгиба

Выбор начальных центроидов

- Случайный, как можно дальше друг от друга
  - outlier может все испортить
- Случайный с выкидыванием отщепенцев
- Предварительная иерархическая кластеризация и центроиды получившихся кластеров в качестве первоначальных для k-means
  - можно проводить над небольшой случайной выборкой из исходных точек
- Несколько разных наборов начальных центроидов и выбор показавшего лучший результат

Сегодня в программе

Задача кластеризации

Методы кластеризации

Иерархическая кластеризация

Алгоритм k-средних

Зонтичная кластеризация

Сокращение вычислений

- Измерять расстояние от каждой точки до каждого центроида может быть дорого
- С некоторыми центроидами даже связываться не хочется
- Давайте заранее определим «зонтики» – (перекрывающиеся) области, где имеет смысл вычислять расстояние
- Расстояние между точками из разных зонтиков будет равно ∞

Построение зонтиков

- У зонтика есть центр, внешний радиус T_1 и внутренний радиус T_2
- Шаги построения:
  1. случайно выбрать центр c
  2. для каждой точки p исходного множества если $d(p, c) <= T_2$ то забыть ее; если $T_2 < d(p, c) <= T_1$ то записать точку в зонтик
  3. делать так пока каждая точка не окажется хотя бы в одном зонтике

Применение к k-means

- В k-means расстояние между точками из разных зонтиков равно ∞
- Все остальное точно так же

Построение зонтиков с Map-Reduce

- Шаг первый:
  - входные данные поделить на фрагменты
  - map (много задач): найти зонтики в одном фрагменте и выплюнуть их центры
  - reduce (1 задача): сделать то же самое с полученными центрами. Получить список центров.
- Шаг второй:
  - map: взять фрагмент, список центров и определить в какие зонтики попала каждая точка
  - reduce: записать либо пары (точка, центр) либо (центр, список точек) либо и то и другое

Инструменты

- Apache Mahout: алгоритмы кластеризации поверх Hadoop
- Weka: набор data mining библиотек и алгоритмов для одной машины

Занавес

- Кластеризация решает практические задачи
- Много разных алгоритмов
- Простой, широко используемый и масштабируемый алгоритм k-средних
- И его можно существенно ускорить

Эта презентация сверстана в

PAPEERIA

LaTeX в вашем браузере
papeeria.com

Литература I