Кластерный анализ

Наумов Д.А., доц. каф. КТ

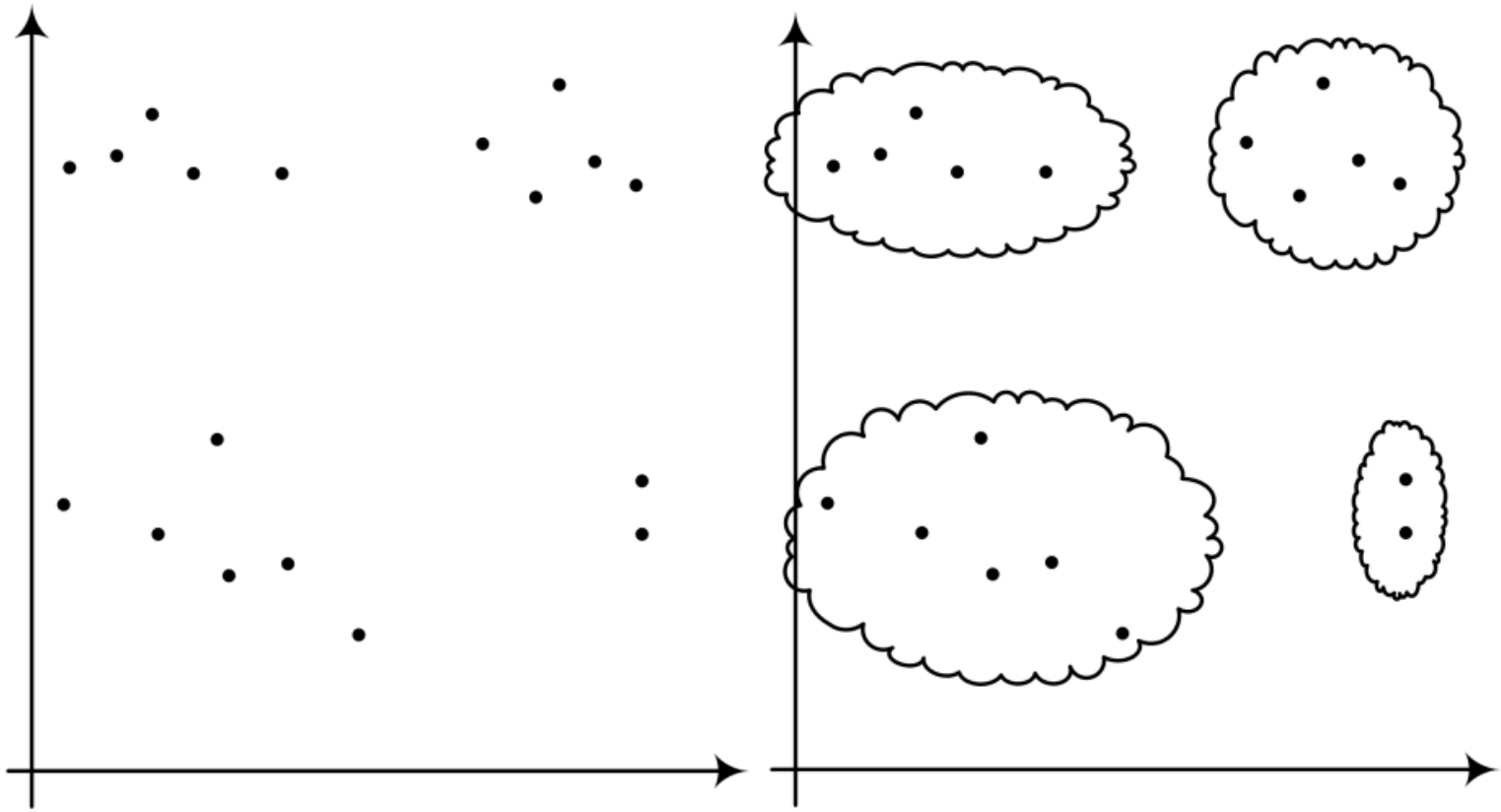
Экспертные системы и искусственный интеллект, 2019

Содержание лекции

Кластеризация

Задача кластеризации

частично сгруппированные точки на плоскости или в пространстве большей размерности разбиваются на близкорасположенные группы.



Кластеризация

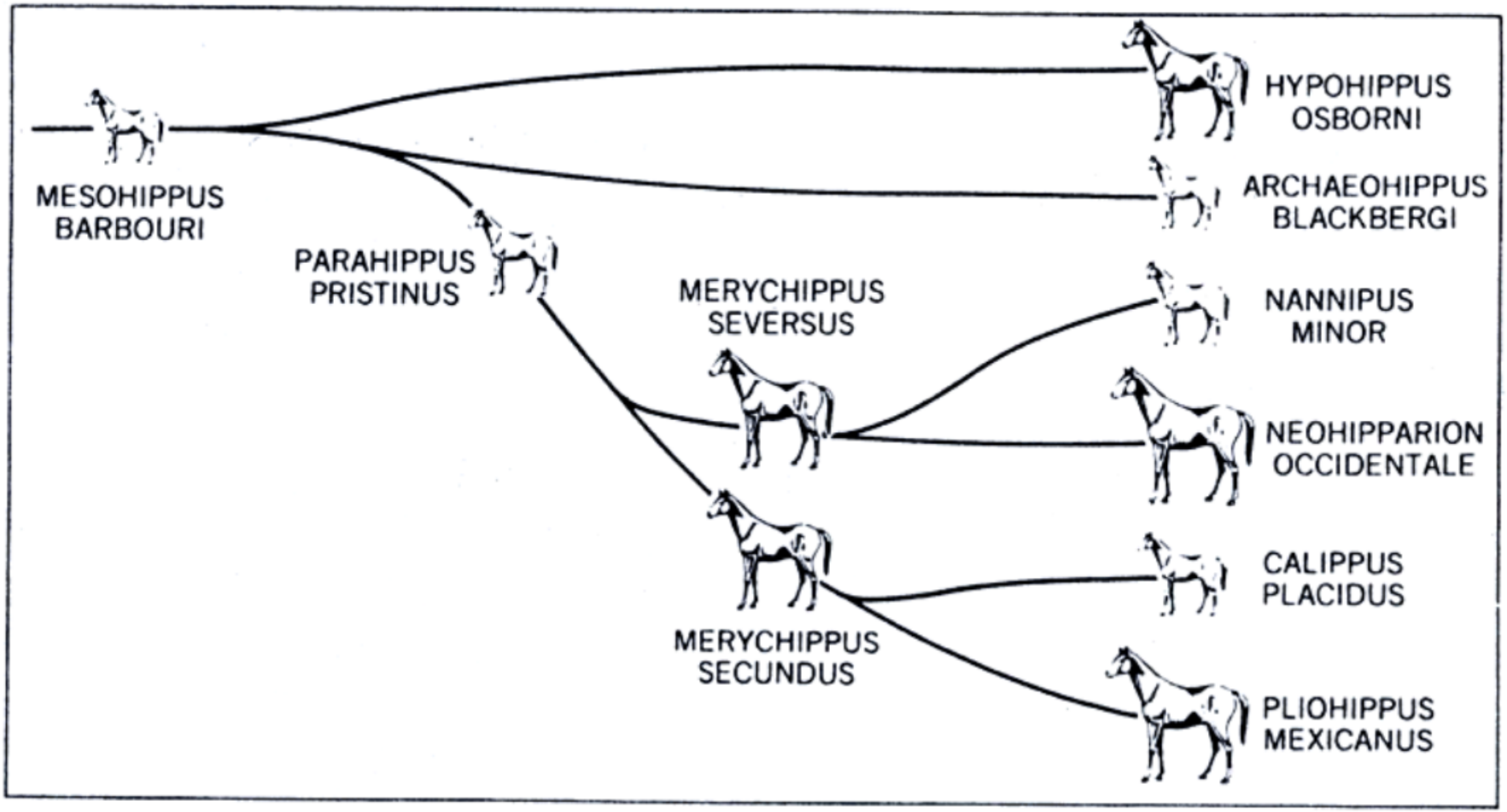


Figure 12 An example of a taxonomic tree (from Sokal, 1966).

Задачи кластеризации

- лежат в основе анализа данных (data mining);
  - анализ покупок в супермаркетах;
  - разделение документов на жанры и тематики;
  - контекстная рекалама и анализ предпочтений пользователя.
- распознование образов: получение границ областей и выделение общих принаков;
- биоинформатика: анализ длинных последовательностей атомов в белках
  - разбиение генов на кластеры;
  - выделение генов на группы по функциональной схожести;
  - выделение генов, отвечающие за биологическое свойство.
- маркетологические исследования для сегментации рынка;
- анализ социальныз сетей;

Кластеризация

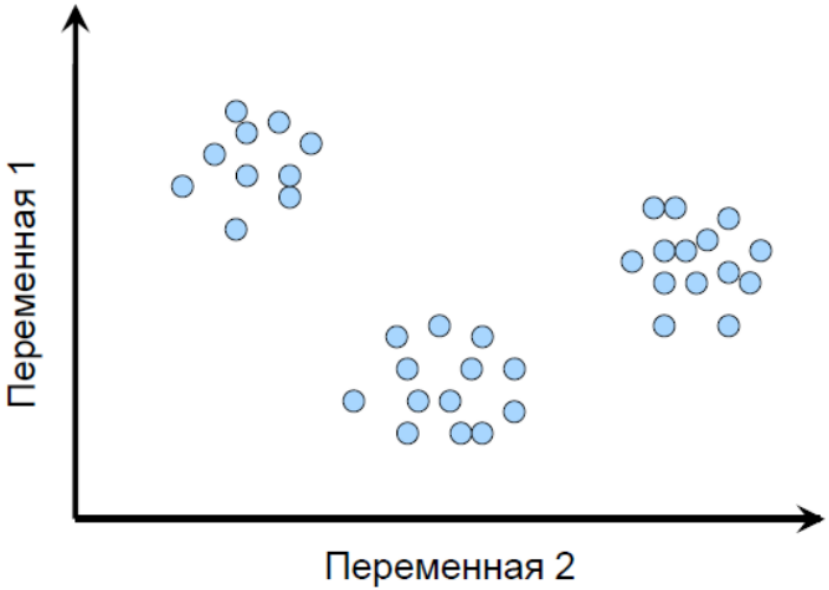
Кластеризация - типичная задача статистического анализа

задача классификации объектов одной природы в несколько групп так, чтобы объекта одной группы обладали одним и тем же свойством. Под свойством понимается близоть друг к другу относительно выбранной метрики.

Сложность задачи кластеризации:

- высокая размерность реальных задач (сотни и тысячи);
- необхождим компромис: число точек и размерность пространства.

Кластерный анализ в теории



Кластерный анализ на практике

Кластеризация

Пусть дан набор тестовых примеров

$$X = \{x_1, \ldots, x_n\}$$

и функция расстояния между ними

$$\rho : X \times X \to \mathbb{R}.$$

Требуется разбить X на непересекающиеся подмножества, которые, собственно, и называются кластерами, так, чтобы каждое подмножество состояло из *близких объектов*, а объекты разных подмножеств *существенно различались*.

- что такое "близкие объекты"?
- что такое "существенно различающиеся объекты"?

Меры близости между объектами

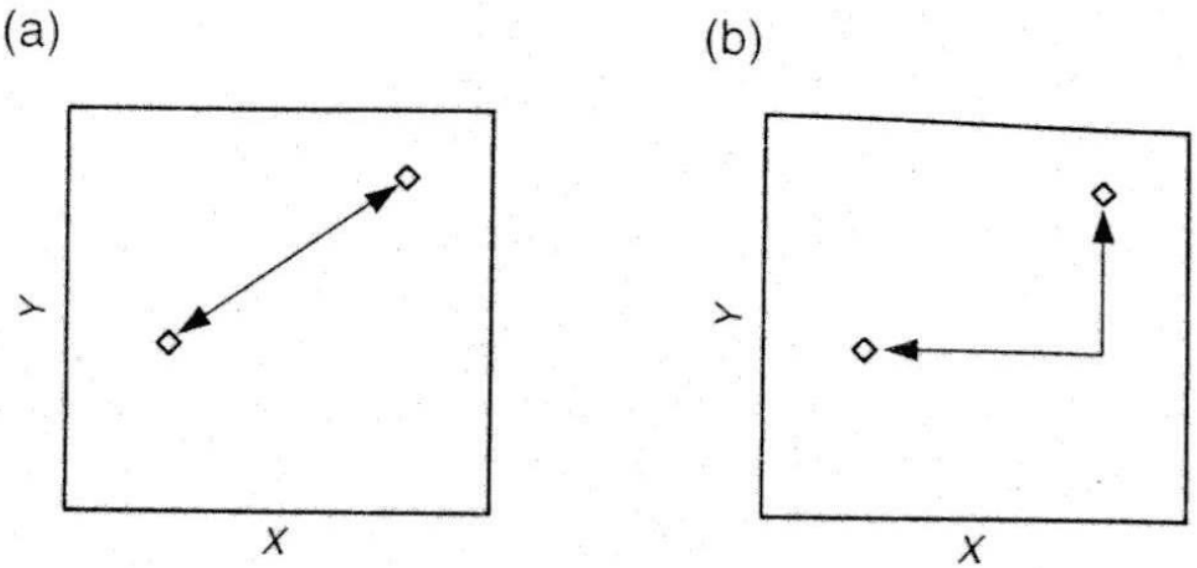


Figure 15.2 Different measures of distance between two points: (a) Euclidean distance, (b) Manhattan or city block distance

Меры близости между объектами

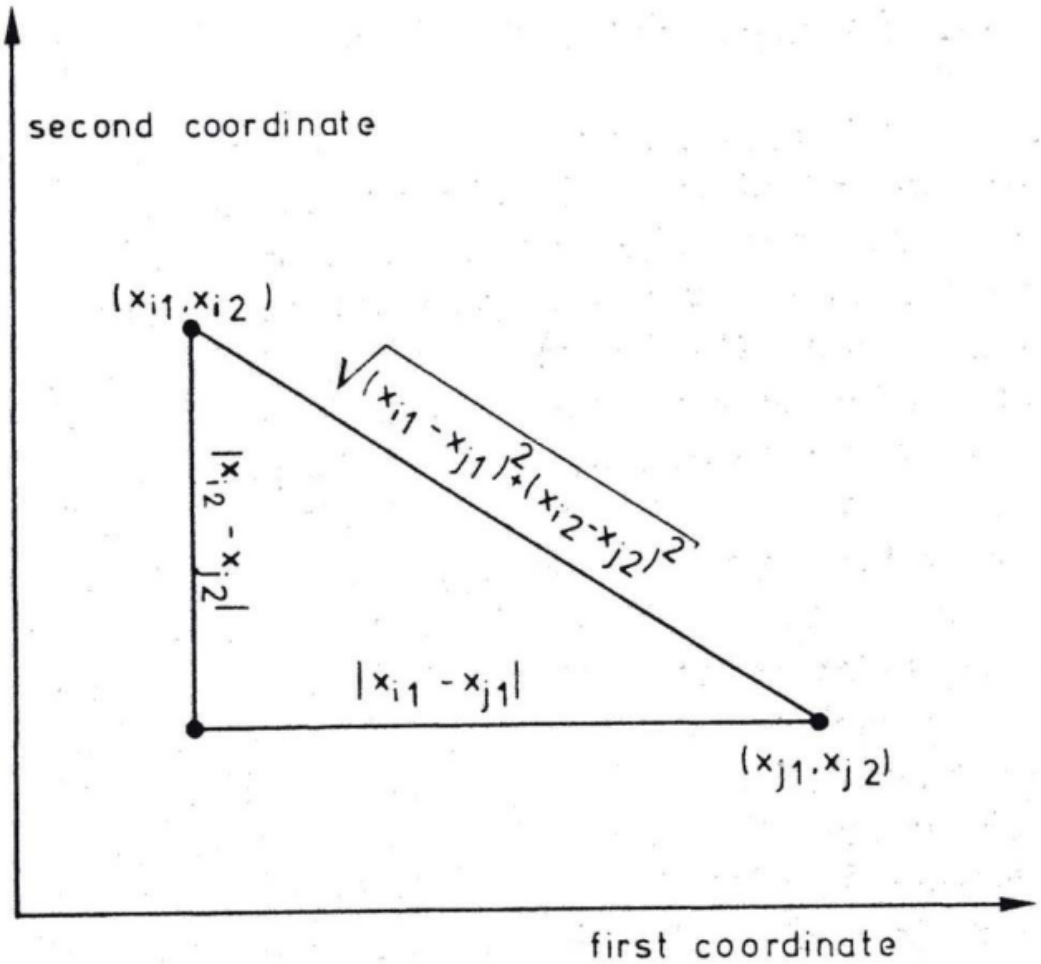


Figure 6 Illustration of the Euclidean distance formula.

Меры близости между объектами

Меры близости между объектами (меры подобия)

Показатели	*Формулы*
Для количественных шкал	
Линейное расстояние	$d_{1ij} = \sum_{l=1}^{m} \lvert x_i^l - x_j^l \rvert$
Евклидово расстояние	$d_{Eij} = \left(\sum_{l=1}^{m} (x_i^l - x_j^l)^2 \right)^{1/2}$
Квадрат евклидово расстояние	$d_{Eij}^2 = \sum_{l=1}^{m} (x_i^l - x_j^l)^2$
Обобщенное степенное расстояние Минковского	$d_{Pij} = \left(\sum_{l=1}^{m} (x_i^l - x_j^l)^p \right)^{1/p}$
Расстояние Чебышева	$d_{ij} = \max_{1 \le i,j \le l} \lvert x_i - x_j \rvert$
Расстояние городских кварталов (Манхэттенское расстояние)	$d_M(x_i, x_j) = \sum_{l=1}^{n} \lvert x_i^l - x_j^l \rvert$

Расстояние tf-idf

tf-idf

мера близости двух текстовых документов

Документ представляется в виде вектора из n термов с некоторыми весами. Разные подходы к анализу текстов различаются в том:

- что такое терм;
- как определять веса.

Термы

(обычно) слова, встречающиеся в документе.

Документ превращается при этом в неупорядоченный набор слов (bag of words).

Сложные структуры в качестве термов? нецелесообразно!

- нужно будет слишком много текстов,
- результат не будет существенно лучше bag-of-words подхода.

Расстояние tf-idf

Для определения весов обычно используют два основных подхода:

- либо бинарный атрибут со значениями 01 (есть слово или нет слова),
- либо весовую функцию, меру tdf (term frequency inverse document frequency).

Мера tf-idf

- была предложена в начале 1970-х годов и с тех пор активно используется в анализе текстовой информации и information retrieval;
- состоит из двух других: tf (частота терма, term frequency) и idf (обратная частота терма в документах, inverse document frequency).

Расстояние tf-idf

Частота терма

доля числа появлений этого терма по отношению к размеру всего документа.

$$\mathrm{tf}(t_k, d_j) = \frac{\#(t_k, d_j)}{\sum_k \#(t_k, d_j)},$$

где #(tk,dj) - число, показывающее, сколько раз терм tk встречается в документе dj.

Расстояние tf-idf

Обратная частота терма

показывает, насколько терм вообще важен, насколько он характерен для данного массива текстов.

Чем реже встречается терм в имеющемся массиве, тем он характернее.

$$\mathrm{idf}(t_k, d_j) = \log \frac{|D|}{\#_D t_k},$$

где где D - имеющийся набор данных, а #Dtk - количество документов из D, в которых хотя бы однажды встречается tk.

Расстояние tf-idf

Мера tdf для терма tk и документа dj в массиве D равна:

$$\mathrm{tfidf}(t_k,d_j) = \mathrm{tf}(t_k,d_j)\mathrm{idf}(t_k,d_j) = \frac{\#(t_k,d_j)}{\sum_k \#(t_k,d_j)} \log \frac{|D|}{\#_D t_k}$$

Вектор весом можно нормализовать:

$$w_{kj} = \frac{\mathrm{tfidf}(t_k,d_j)}{\sqrt{\sum_{s=1}^{r}(\mathrm{tfidf}(t_k,d_j))^2}}$$

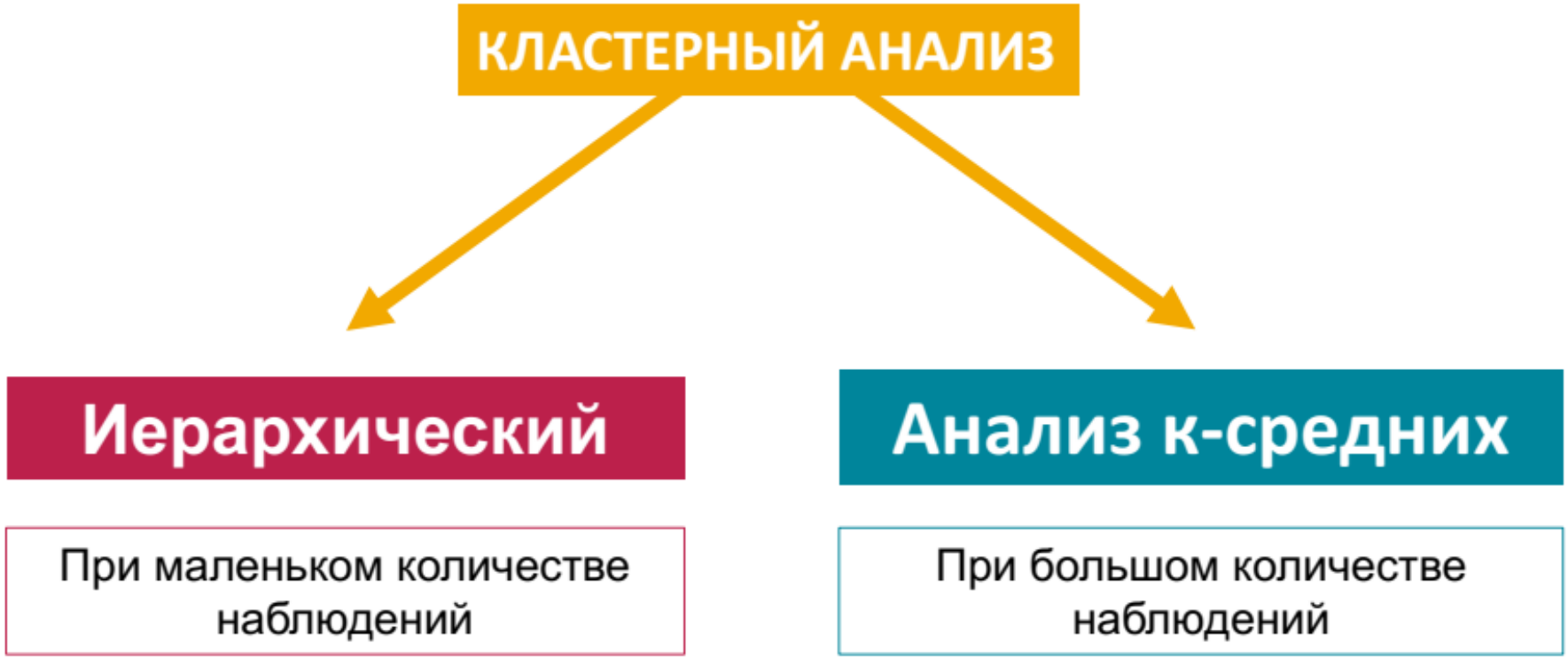
Теперь документ можно представить в виде вектора, размерность которого равна количеству термов (важно сохранять словарь не слишком большим за счет удачного выбора термов).

Расстояние между документами

Расстояние между документами - используется не простое декартово расстояние, а угол между векторами, косинусоидальная мера похожести (cosine similarity measure):

$$\theta(d_1, d_2) = \arccos \frac{d_1 \cdot d_2}{\|d_1\| \|d_2\|}$$

Кластерный анализ



- иерархическая кластеризация - алгоритм последовательно строит кластеры из уже найденных кластеров;
- неиерархическая кластеризация - алгоритм пытается распознать всю структуру сразу или строит кластеры один за другим, не разделяя и не соединяя уже выделенные кластеры.

Кластерный анализ

Иерархическая кластеризация:

- агломеративная: алгоритм начинает с индивидуальных элементов (кластеров из одного элемента), а затем последовательно объединяет их, получая требуемую структуру;
- разделительная, когда алгоритм начинает с одного кластера, содержащего все точки, а потом последовательно делит его на части.

Неиерархические методы, как правило, стремятся оптимизировать некую целевую функцию, которая описывает качество кластеризации.

- агоритмы, основанные на методах теории графов;
- алгоритм EM;
- алгоритм k-средних;
- нечеткие алгоритмы.

Иерархическая кластеризация. Пусть нужно кластеризовать точки x1,x2, . . . ,xn в некотором метрическом пространстве с метрикой.

1. на первом шаге мы считаем каждую точку отдельным кластером;
2. ближайшие точки объединяем и далее относимся к ним как к единому кластеру.
3. при итерации этого процесса получается дерево, в листьях которого отдельные точки, а в корне кластер, содержащий все точки вообще.

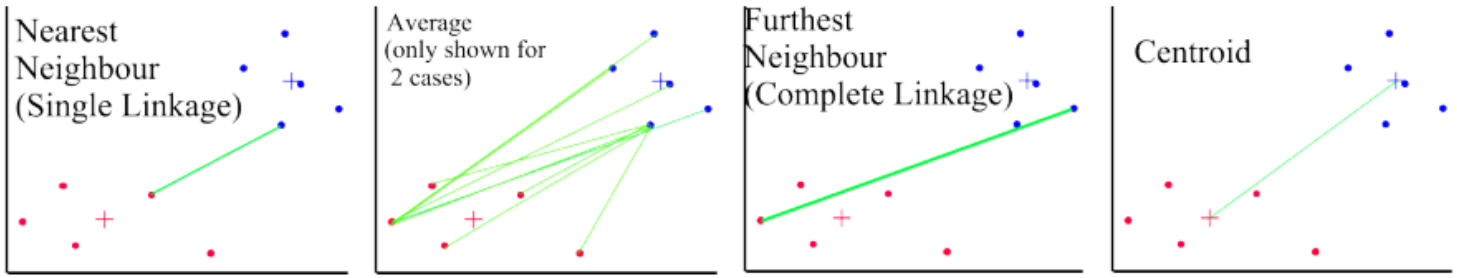
HierarchyCluster($X = \{x_1, \ldots, x_n\}$):

1. Инициализируем $C = X$, $G = X$.
2. Пока в C больше одного элемента:
   а) Выбираем два элемента C c_1 и c_2, расстояние между которыми минимально.
   б) Добавляем в G вершину c_1c_2, соединяем её с вершинами c_1 и c_2.
   в) $C := C \cup \{c_1c_2\} \setminus \{c_1, c_2\}$.
3. Выдаём G.

Расстояние между кластерами

Метод кластеризации – это способ вычисления расстояний между кластерами. Существуют следующие основные методы кластеризации:

- Межгрупповая связь (Between-groups linkage)
- Внутригрупповая связь (Within-groups linkage)
- Ближайший сосед (Nearest neighbor)
- Самый дальний сосед (Furthest neighbor)
- Центроидная кластеризация (Centroid clustering)
- Медианная кластеризация (Median clustering)
- Метод Варда (Уорда)(Ward's method)



Расстояние между кластерами

Стандартизация данных

$$Z_{ij} = \frac{X_{ij} - \bar{X}_j}{\sigma_{X_j}}$$

Z-стандартизация	Из значений вычитается среднее и затем они делятся на стандартное отклонение.
Разброс от -1 до 1	Линейным преобразованием переменных добиваются разброса значений от -1 до 1.
Разброс от 0 до 1	Линейным преобразованием переменных добиваются разброса значений от 0 до 1.
Максимум 1	Значения переменных делятся на их максимум.
Среднее 1	Значения переменных делятся на их среднее.
Стандартное отклонение 1	Значения переменных делятся на стандартное отклонение.

Расстояние между кластерами

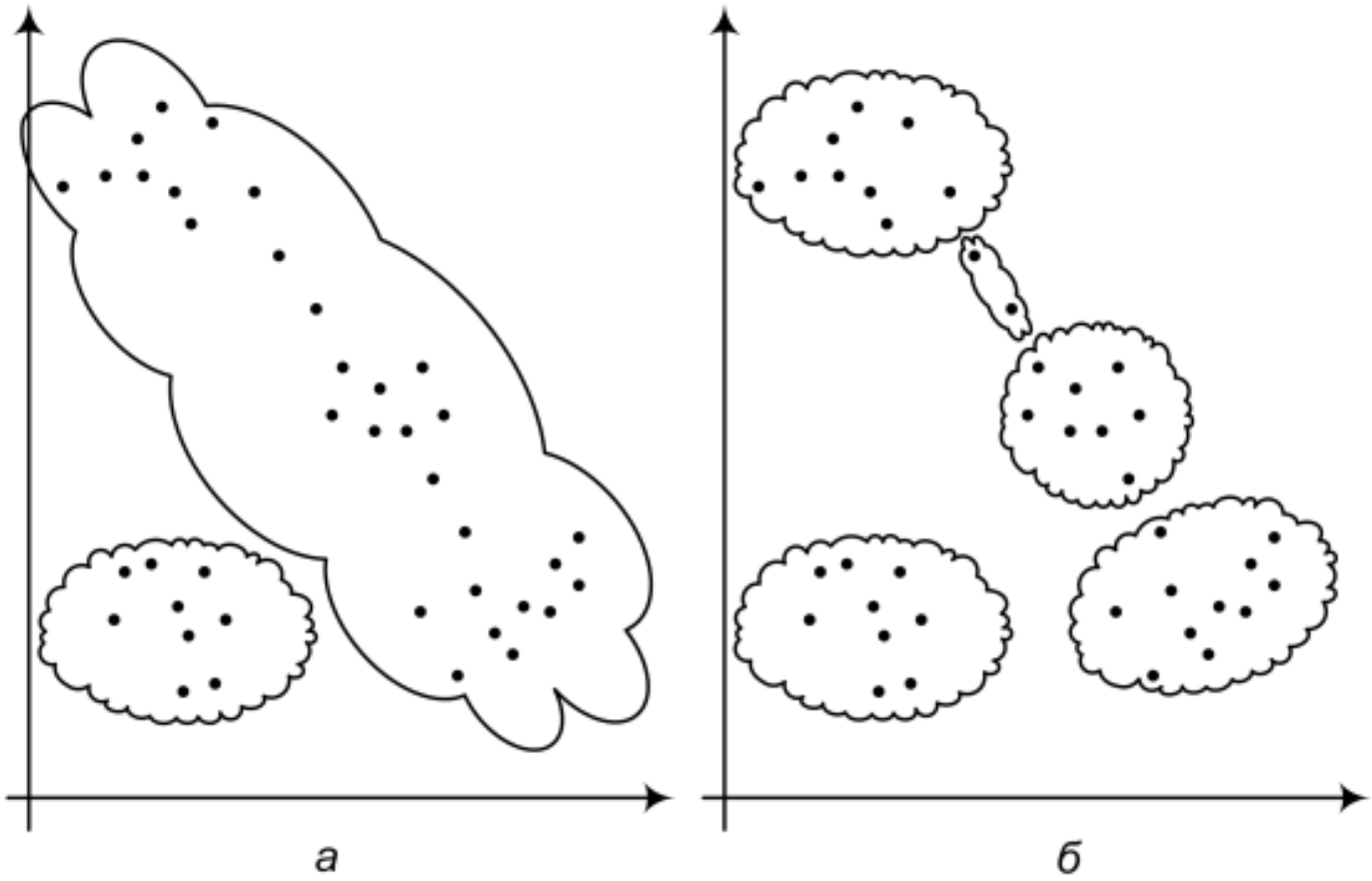


Рис. 6.3. Виды иерархической кластеризации:
а — single-link; *б* — complete-link

Алгоритм кластеризации на основе теории графов

Определение. *Остовное дерево* (spanning tree) графа $G = \langle V, E \rangle$ — это связный подграф G, содержащий все вершины G и являющийся деревом. *Минимальное остовное дерево* (minimal spanning tree) графа $G = \langle V, E \rangle$ с заданными на рёбрах весами $w : E \to \mathbb{R}$ — это такое остовное дерево T, что его суммарный вес не превосходит суммарного веса любого другого остовного дерева T':

$$\forall T' \quad \sum_{e \in T'} w(e) \geqslant \sum_{e \in T} w(e).$$

1. построить минимальное остовное дерево;
2. выкидывать из него ребра максимального веса до тех пор, пока не получится нужное число кластеров.

Сколько ребер выбросим, столько кластеров получим.

Алгоритм Краскала (Kruskal)

Kruskal$(G = \langle V, E \rangle, w : E \to \mathbb{R})$:

1. Отсортировать рёбра G по возрастанию веса, инициализировать подграф $S \subseteq G$, $S := \emptyset$.
2. Для каждого ребра e в порядке возрастания веса:
   а) если конечные точки e ещё не связаны в S, добавить e в S.

1. на каждом шаге выбираем ребро с минимальным весом,
2. если оно соединяет два дерева, добавляем его в остовное дерево, если нет, пропускаем.

Алгоритм Борувски (Boruvka)

Boruvka($G = \langle V, E \rangle, w : E \to \mathbb{R}$):

1. Инициализируем список из n деревьев L, в каждом дереве по одной вершине.
2. Пока в L больше одного дерева:
   а) для каждого $T \in L$ найти ребро минимального веса, соединяющее T с $G \setminus T$;
   б) добавить все эти рёбра к минимальному остовному дереву;
   в) объединить пары пересекающихся деревьев в L (размер L при этом уменьшается вдвое).

1. можно строить минимальное остовное дерево, начав с одной вершины и добавляя ребра минимального веса, пока не покроем весь граф (алгоритм Прима, Prim's algorithm);
2. будем делать то же самое, но во всех вершинах одновременно (распараллелим этот процесс).

Алгоритм k-средних (k-means)

1. инициализировать центры кластеров каким-нибудь начальным разбиением;
2. классифицировать точки по ближайшему к ним центру кластера;
3. перевычислить каждый из центров;
4. если ничего не изменилось, остановиться, если изменилось повторить.

kMeans(X,|C|):

1. Инициализировать центры |C| кластеров:

$$\mu_1, \ldots, \mu_{|C|}.$$

2. Пока принадлежность кластерам не перестанет изменяться:
   а) определить принадлежность x_i к кластерам:

$$\text{clust}_i := \text{argmin}_{c \in C} \rho(x_i, \mu_c);$$

   б) определить новое положение центров:

$$\mu_c := \frac{\sum_{\text{clust}_i = c} f_j(x_i)}{\sum_{\text{clust}_i = c} 1}.$$

Алгоритм k-средних (k-means)

Сначала определяется **центр кластера**, а затем группируют все объекты в пределах заданного от центра порогового значения.

Недостатки:

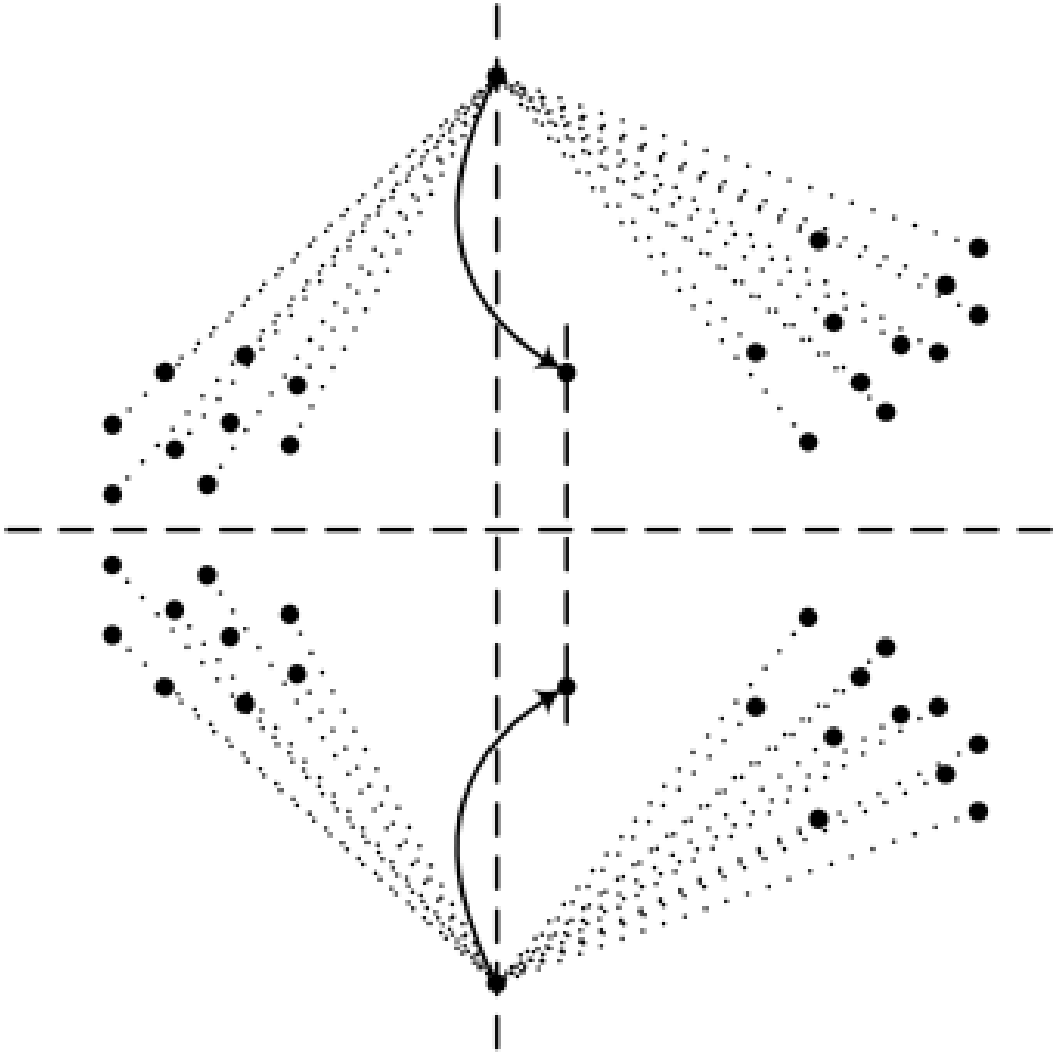
- Чувствительность к выбросам
- Необходимо заранее задавать количество кластеров, а не как в иерархическом анализе, получать это в качестве результата

Проблему с выбором числа кластеров можно преодолеть проведением иерархического анализа со случайно отобранной выборкой наблюдений и, таким образом, определить оптимальное количество кластеров.

Достоинства:

- Простота использования
- В качестве метрики используется Евклидово расстояние
- Возможность наглядной интерпретации кластеров с использованием графика «Средних значений в кластерах»

Пример некорретной кластеризации



Алгоритмы нечеткой кластеризации

Мера принадлежности кластеру - вещественное число из [0,1], и точки на краю кластера сменьше принадлежатї кластеру, чем в центре. Будем обозначать принадлежность кластеру $c \in C$ через $u_c(x)$. Меры принадлежности обычно выбирают так, чтобы

$$\forall x \; u_c(x) \geqslant 0 \qquad \sum_{c \in C} u_c(x) = 1$$

Нечеткие алгоритмы кластеризации (одним из которых является алгоритм с-средних) минимизируют ту или иную меру ошибки. Часто применяется мера

$$E(C) = \sum_{c \in C} \sum_{x \in X} u_c^m(x) \rho^2 \left(x, Center_c\right)$$

где m - некоторый вещественный параметр.

Алгоритмы нечеткой кластеризации

cMeans(X,|C|):

1. Случайно выбрать коэффициенты $u_c(x)$ для всех $x \in X$ и $c \in C$.
2. Пока алгоритм не сойдётся:
   а) Для всех $c \in C$

$$\text{Center}_c := \frac{\sum_{x \in X} u_c(x)^m x}{\sum_{x \in X} u_c(x)^m}.$$

   б) Для всех $c \in C$ и всех $x \in X$

$$u_c(x) := \frac{1}{\sum_{c' \in C} \left(\frac{\rho(\text{Center}_c, x)}{\rho(\text{Center}_{c'}, x)} \right)^{2/(m-1)}}.$$

- при $m = 2$, то перевзвешивание эквивалентно линейной нормализации коэффициентов так, чтобы их сумма была равна единице.
- при $m \to 1$ все больший и больший вес придается самому близкому кластеру, и алгоритм становится все более похож на алгоритм k-средних.

Этапы кластерного анализа

Принятие решений о числе кластеров

1. Необходимо руководствоваться практическими и теоретическими соображениями. Исходя из цели исследования, например, может быть необходимо три кластера.

2. В иерархической кластеризации в качестве критерия используются расстояния. Необходимо смотреть на **коэффициент в протоколе объединения** (расстояние между двумя кластерами, определенное на основании выбранной дистанционной меры с учётом предусмотренного преобразования значений).

   - Когда мера расстояния между двумя кластерами увеличивается скачкообразно, процесс объединения в новые кластеры необходимо остановить. Иначе будут объединены кластеры, находящиеся на большом расстоянии друг от друга.

   - Оптимальным считается число кластеров равное разности количества наблюдений и количества шагов, после которого коэффициент увеличивается скачкообразно.

3. Размеры кластеров должны быть значимыми.

Оценка качества кластеризации

- Необходимо выполнять кластерный анализ одних и тех же данных, но с использованием **различных способов измерения расстояния**.
- Сравнить результаты, полученные на основе различных способов расстояния, чтобы определить, насколько совпадают полученные результаты.
- Разбить данные на **две равные части** случайным образом. Выполнить кластерный анализ отдельно для каждой половины. Сравнить кластерные центроиды двух подвыборок.
- Случайным образом **удалить некоторые переменные**. Выполнить кластерный анализ по сокращенному набору переменных. Сравнить результаты с полученными на основе полного набора переменных.