Лекция 4: кластеризация

Что есть кластер?

- Кластер: группа «похожих» объектов
  - «похожих» между собой в группе (внутриклассовое расстояние)
  - «не похожих» на объекты других групп
  - Определение неформальное, формализация зависит от метода
- Кластерный анализ
  - Разбиение множество объектов на группы (кластеры)
- Тип моделей:
  - «описательный» (descriptive) Data mining => одна из задач - наглядное представление кластеров
  - «прогнозный» (predictive) Data mining => разбиение на кластеры, а затем «классификация» новых объектов
- Тип обучения:
  - всегда «без учителя» (unsupervised) => тренировочный набор не размечен
- Этапы кластерного анализа:
  - Подготовка данных
  - Применение алгоритма
  - Визуализация и интерпретация результатов

Применение методов кластеризации в задачах анализа данных

- Кластеризация ради кластеризации:
  - Выявление и описание групп (человек не способен «осознать» более 10 объектов в одной задаче, как обработать выборку с миллионами?)
  - «Сжатие» информации (особенно в обработке мультимедиа)
  - Построение различных поисковых индексов (сравниваем не со всеми, а начинаем с прототипов кластеров)
- Мощнейшее средство предобработки данных:
  - Дискретизация (от чисел к «понятиям»)
  - Уменьшение размерности (от больших объемов к «реальным»)
  - Обработка пропущенных значений (инициализируем и итерационно «улучшаем» пропуски)
  - Поиск исключений и артефактов (что не в кластере, то под «подозрением»)
- Предварительный этап перед классификацией:
  - Стратифицированные модели
  - Поиск «кандидатов» классов для классификации «с учителем»
  - Поиск состояний для марковских моделей прогнозирования
  - Поиск шаблонов правил (проекция кластеров)

Прикладное применение методов кластеризации

- Пожалуй, самая «востребованная» задача в Data Mining
  - Почему? Потому что получаем «что-то полезное» практически из «ничего»
- «Экономика»:
  - «Сегментация» рынков, клиентов, товаров, услуг
- «Наука» и медицина:
  - Таксономии и обработка результатов экспериментов или исследований
- Обработка мультимедиа:
  - «Сжатие» и «сегментация» изображений, видеоряда, звука
- Электронные текстовые документы:
  - Рубрикация и индексирование

Качество кластеризации

- Хороший метод кластеризации находит кластеры
  - с высоким «внутриклассовым» сходством объектов
  - и низким «межклассовым» сходством объектов
- Оценка качества кластеризации (нет понятия «точность»)
  - необходима, так как влияет на выбор параметров метода
  - определяется либо экспертом – субъективная величина
  - либо «перекрестной» проверкой целевой функции кластеризации
- Качество кластеризации зависит:
  - от метода кластеризации
  - от меры сходства (или расстояния)

Требования к методу кластеризации

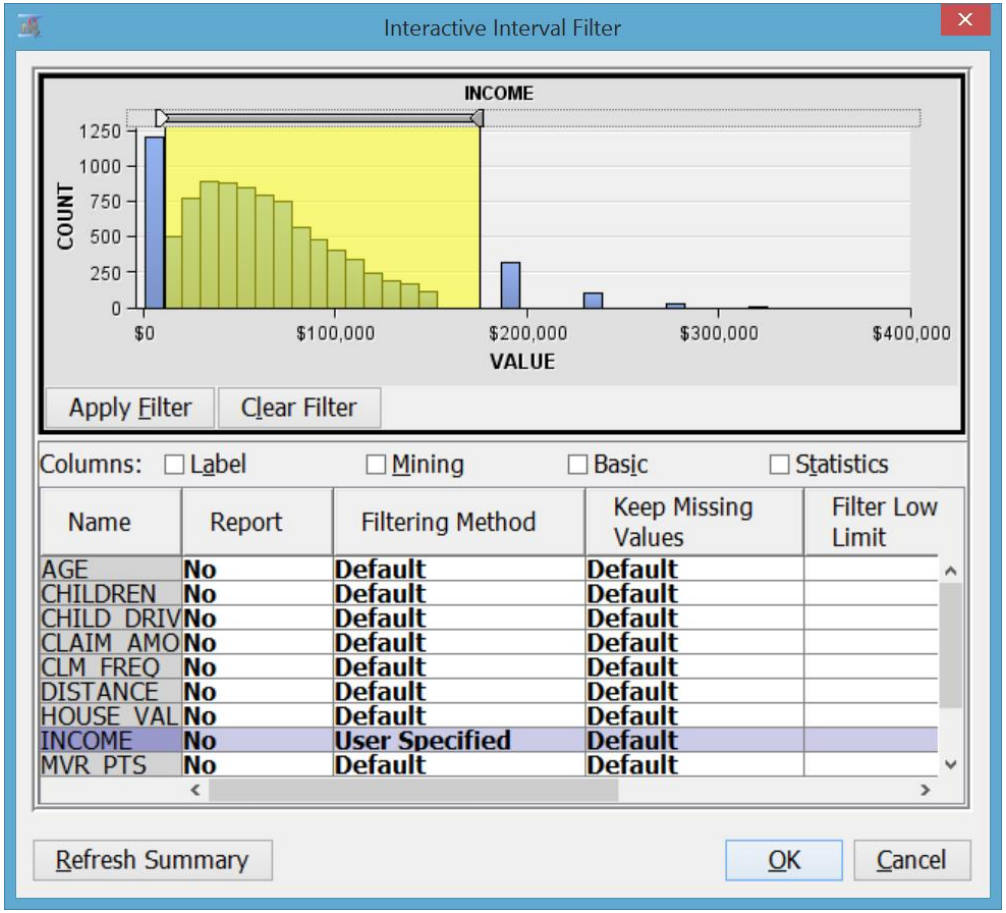
- Масштабируемость
- Поддержка различных типов атрибутов и структур данных
- Возможность находить кластеры сложной формы
- Отсутствие обязательных требований к наличию априорных знаний о выборке (например, о распределениях)
- Устойчивость к «шуму» и выбросам
- Возможность работы с высокой размерностью и с большой выборкой
- Возможность включать пользовательские ограничения и зависимости
- Интерпретируемость и наглядность (прототипы, границы, правила, функции принадлежности и т.п.)
- Интуитивность параметров кластеризации

Подготовка данных для кластеризации

- Отбор наблюдений
  - Что я разбиваю на кластеры?
  - Решаемые задачи: исключить выборсы (узел Filter), уменьшить выборку (узел Sample)
- Отбор и трансформация переменных
  - Какие характеристики объектов важны? Выбирает эксперт.
  - Переменные коррелируют? (узлы PCA и Variable Clustering)
  - Распределения переменных симметричны? (узел Transform)
- Стандартизация переменных
  - Сравнимы ли масштабы переменных?
  - Делается автоматически узлом кластеризации

Фильтрация данных

- Цель – удаление из выборки артефактов и выбросов
- Правила фильтрации задаются для отдельных переменных:
  - Ручные – задаются допустимые значения переменных (диапазоны для числовых, список для категориальных)
  - Редкие значения для категориальных
  - Нетипичные значения для числовых (задается допустимое отклонение от мат. ожидания или допустимое отклонение от медианы или экстремальные процентили и другое).

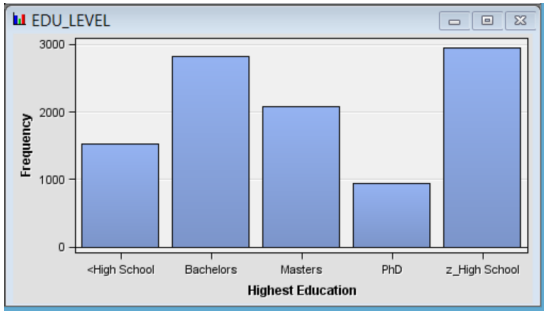


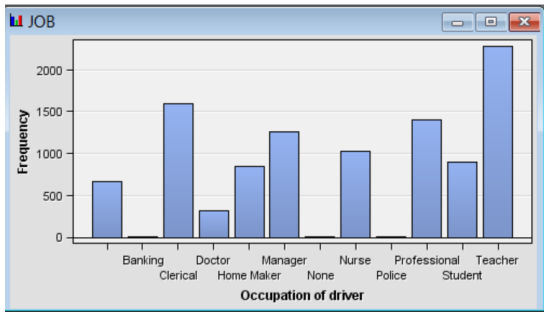
Сокращение обучающей выборки – случайная выборка (Sampling)

- Цель – выбрать «представительное» подмножество примеров:
  - В идеале с тем же распределением
  - Просто случайная выборка работает плохо – не удается сохранить характеристики всего набора
- Адаптивные методы случайной выборки:
  - В соответствии с «грубой» моделью, например кластерной
  - Случайная выборка в рамках экспертных «срезов» (условия на срезы формируются аналитиком)
  - Случайная выборка в рамках «срезов», построенных автоматически по какому-либо классу, высоко селективному атрибуту или их комбинации
  - Основная особенность – выборка в рамках среза или кластера пропорциональна размеру среза или кластера

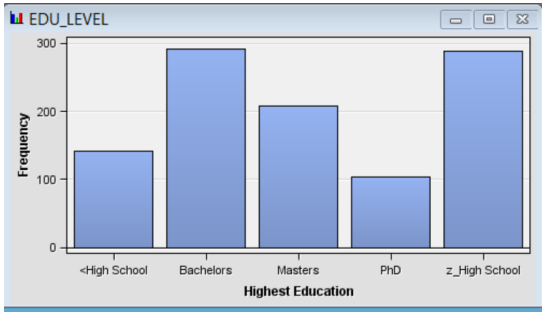
Сокращение обучающей выборки (SAMPLING) – метод стратификации

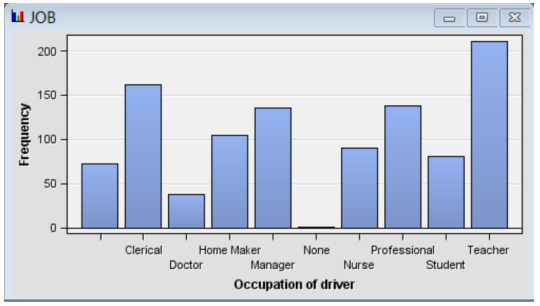
- Задается процент исходной выборки
- Для выбранной категориальной переменной (переменная стратификации) строится частотная диаграмма (для числовой необходима предварительная дискретизация)
- Наблюдения случайным образом выбрасываются так, чтобы сохранить распределение переменной стратификации





Name	Sample Role	Role	Level
AGE	**Default**	Input	Interval
AREA	**Default**	Input	Binary
CAR_USE	**Default**	Input	Binary
CHILDREN	**Default**	Input	Interval
CHILD_DRIV	**Default**	Input	Interval
CLAIM_AMO	**Default**	Rejected	Interval
CLAIM_IND	**Default**	Target	Binary
CLM_FREQ	**Default**	Input	Interval
DISTANCE	**Default**	Input	Interval
EDU_LEVEL	**Stratification**	Input	Nominal
GENDER	**Default**	Input	Binary
HOUSE_VAL	**Default**	Input	Interval
ID	**Default**	ID	Nominal
INCOME	**Default**	Input	Interval
JOB	**Default**	Input	Nominal
MVR_PTS	**Default**	Input	Interval
REVOKED	**Default**	Input	Binary
STATE_COD	**Default**	Rejected	Nominal
STATUS	**Default**	Input	Nominal
VEHICLE_AG	**Default**	Input	Interval
VEHICLE_TY	**Default**	Input	Nominal
VEHICLE_VA	**Default**	Input	Interval
YOJ	**Default**	Input	Interval

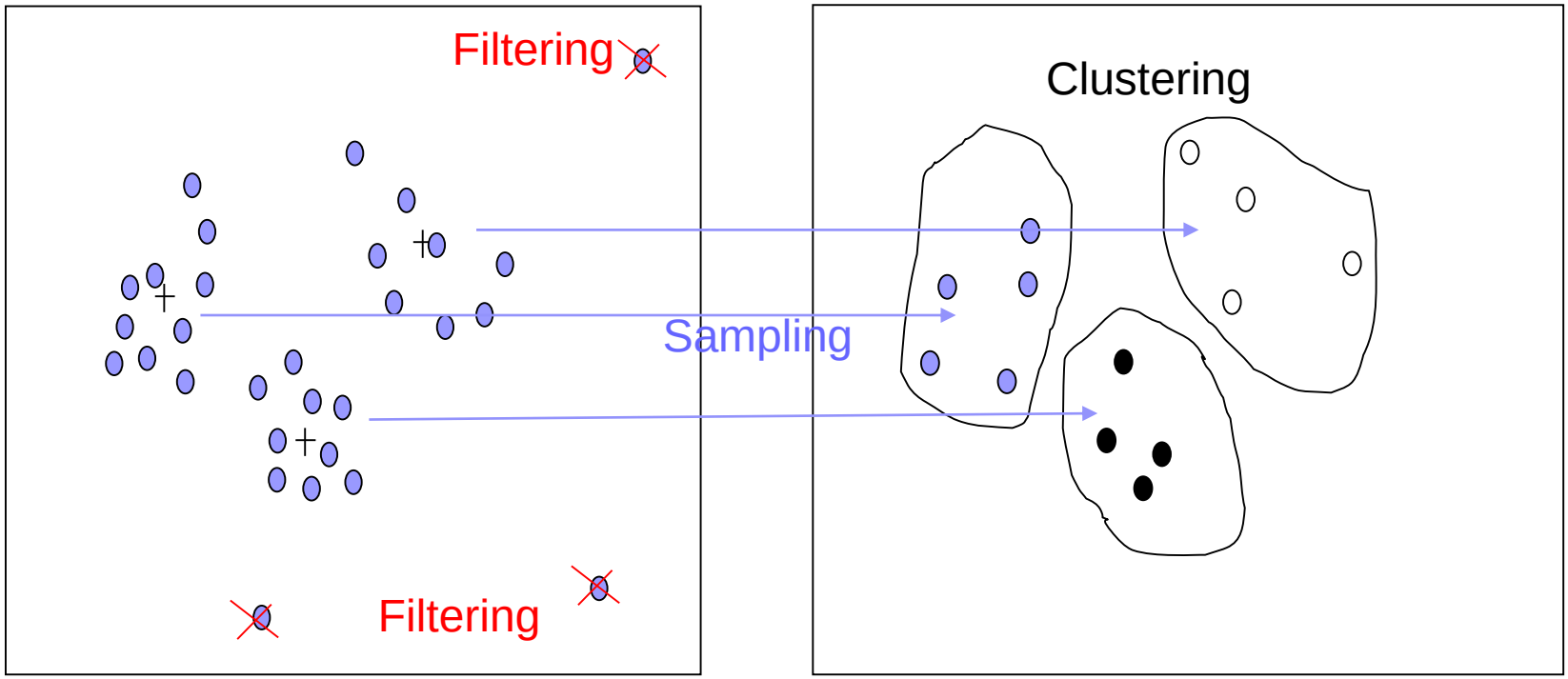




Подготовка данных для кластеризации

«Сырые» данные

Кластерная/стратифицированная случайная выборка



Исходные данные

- Матрица признаков:
  - Числовые
  - Бинарные
  - Номинальные (категориальные)
  - Упорядоченные шкалы
  - Нелинейные шкалы

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- Матрица различия (или сходства):
  - «Естественные» расстояния предметной области
  - Экспертные оценки (противоречивы, нетранзитивны, недостоверны)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Числовые значения – приведение к близким шкалам

- Нормализация на абсолютное отклонение более робастно (устойчиво к ошибкам), чем нормализация на стандартное отклонение:
  - Среднее абсолютное отклонение

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

где

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + ... + x_{nf}\right)$$

  - *z-score* $z_{if} = \frac{x_{if} - m_f}{s_f}$

- Обычная нормализация: $y_{if} = \frac{x_{if} - m_f}{std_f}$

$$std_f = \sqrt{\frac{1}{n-1}\left[(x_{1f} - m_f)^2 + (x_{2f} - m_f)^2 + ... + (x_{nf} - m_f)^2\right]}$$

Меры сходства и различия для исходных данных с числовыми атрибутами

- Обычно строится на основе расстояния:
  - $d(i,j) \geq 0$, $d(i,i) = 0$, $d(i,j) = d(j,i)$, $d(i,j) \leq d(i,k) + d(k,j)$
- Наиболее популярно расстояние Минковского:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)}$$

где $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ и $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ - два объекта с p числовыми атрибутами, q - положительное целое число

- $q = 2$ - Евклидово (не фамилия, но имя) расстояние:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

- $q = 1$, d – расстояние «Манхэтен»:

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

Бинарные атрибуты

- Расстояние Хэмминга = сумма единиц после XOR $(M_{10} + M_{01})$
- Таблица «сопряженных признаков»
  - В ячейках – число совпадающих и несовпадающих значений из p бинарных атрибутов для объектов *j* и *i*

		Object *j*		
		1	0	*sum*
Object *i*	1	M_{11}	M_{10}	$M_{11}+M_{10}$
	0	M_{01}	M_{00}	$M_{01}+M_{00}$
	sum	$M_{11}+M_{01}$	$M_{10}+M_{10}$	$M_{00}+M_{01}+M_{10}+M_{11}$

- На основе коэффициента совпадения $d(i,j) = \frac{M_{10} + M_{01}}{M_{11} + M_{01} + M_{10} + M_{00}}$
  - для симметричных атрибутов (значения равнозначны)
- На основе коэффициента Jaccard $d(i,j) = \frac{M_{10} + M_{01}}{M_{11} + M_{01} + M_{10}}$
  - для асимметричных атрибутов (единица важнее)

Пример

Имя	Пол	Жар	Кашель	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- пол - симметричный атрибут
- остальные ассиметричные
- пусть Y и P соответствует 1, а N соответствует 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

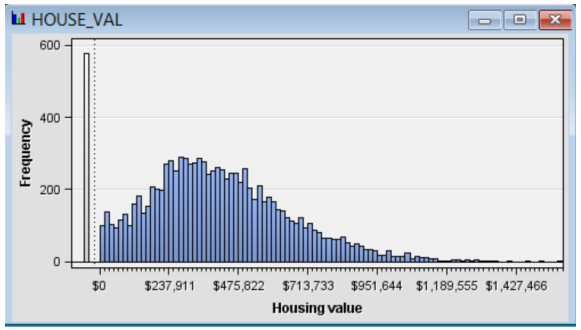
$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

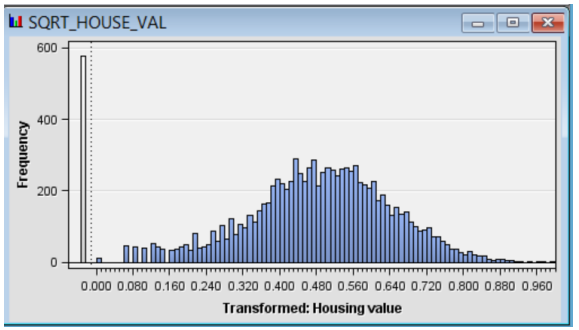
Категориальные атрибуты и шкалы

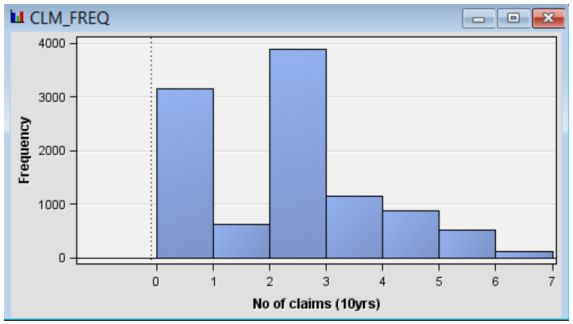
- Категориальные атрибуты:
  - много значений, например, цвета: red, yellow, blue, green
- Подход 1: простое совпадение
  - *M* - число совпадений, *p* - число переменных (аналог нормированного расстояния Хэмминга) $d(i, j) = \frac{p - m}{p}$
- Подход 2: кодирование бинарными векторами
  - Для каждого значения категориального атрибута создается отдельная бинарная переменная: один категориальный атрибут с M возможными значениями => бинарный вектор длины M
- Категориальные упорядоченные шкалы: $r_{if} \in \{1, \ldots, M_f\}$
  - Могут быть и дискретными и непрерывными
  - Порядок важен, «разница» - нет = ранги
  - сводятся к числовым: заменить x_{if} на его ранг, отобразить на [0, 1] с нормировкой: $z_{if} = \frac{r_{if} - 1}{M_f - 1}$
  - затем использовать стандартные расстояния

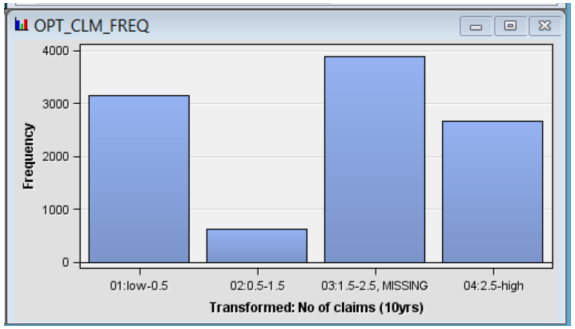
Преобразование переменных

- Простые преобразования:
  - Функции от исходной (log, exp, …), дискретизации (на бакеты и квантили), объединение редких категориальных значений и т.д.
- Адаптивные преобразования – перебор простых и выбор лучшего по некоторому криетрию:
  - Нормальность распределения результата, корреляция с откликом, Оптимальная дискретизация и т.д.





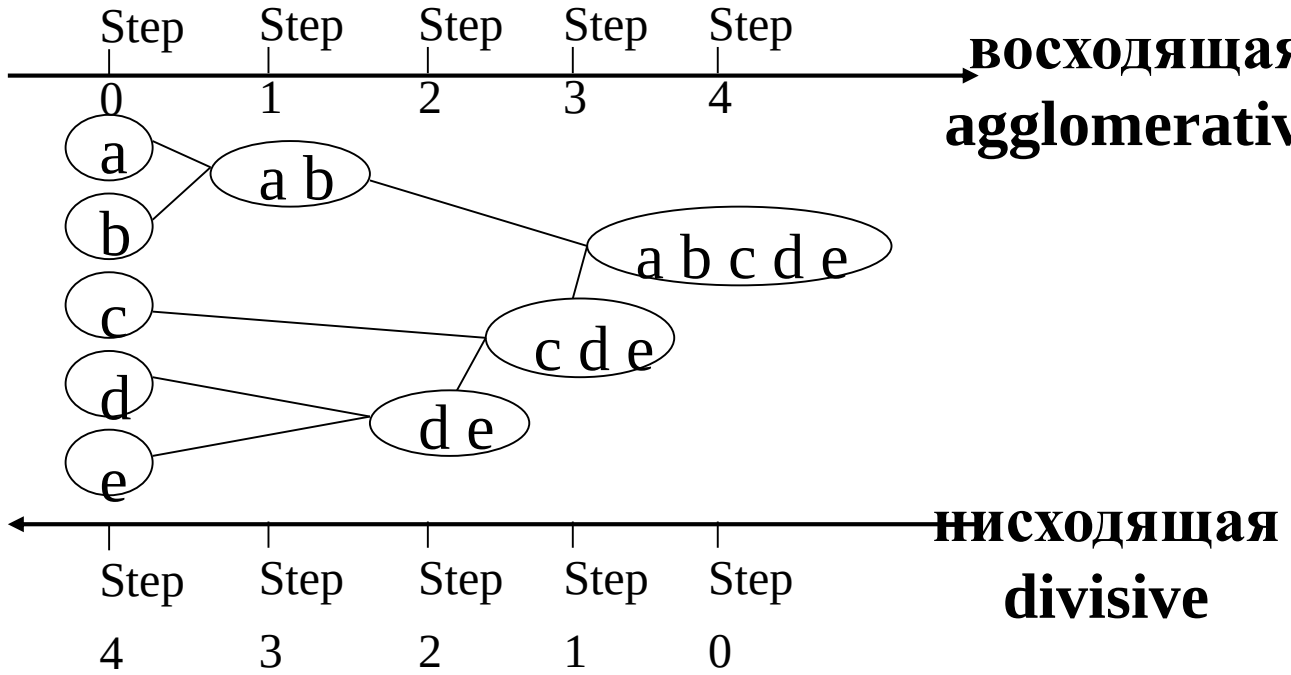




Основные типы алгоритмов кластеризации

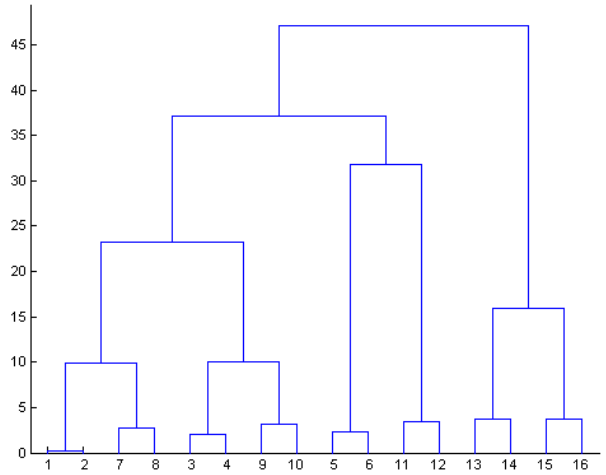
- Иерархические:
  - Создается иерархическая декомпозиция исходного множества объектов в соответствии с некоторой стратегией «объединения» (восходящая кластеризация) или «разбиения» (нисходящая)
- На основе группировки (partitioning):
  - Направленный перебор вариантов разбиения исходного множества объектов, выбор лучшего по некоторому критерию
  - k-means, k-medoids
- На основе связности:
  - Кластеры ищутся в виде связных областей с помощью локальной оценки числа ближайших соседей
- Модель-ориентированые (статистические):
  - Выбирается некоторая гипотеза (параметрическая модель) о структуре кластеров и находятся, параметры, наилучшим образом приближающие эту модель

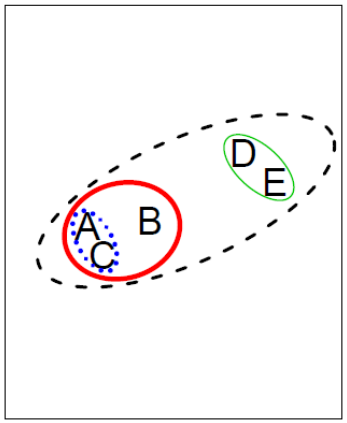
Иерархическая кластеризация

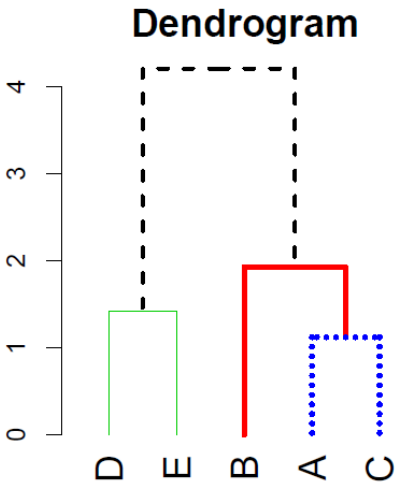


- Параметры:
  - Используется только матрица сходства (различия) и не требуется дополнительных параметров (например, числа кластеров)
- Процесс:
  - «Пошаговое» объединение ближайших кластеров (восходящая) или разбиение наиболее удаленных (нисходящая)

Представление иерархических кластеров - Дендрограмма

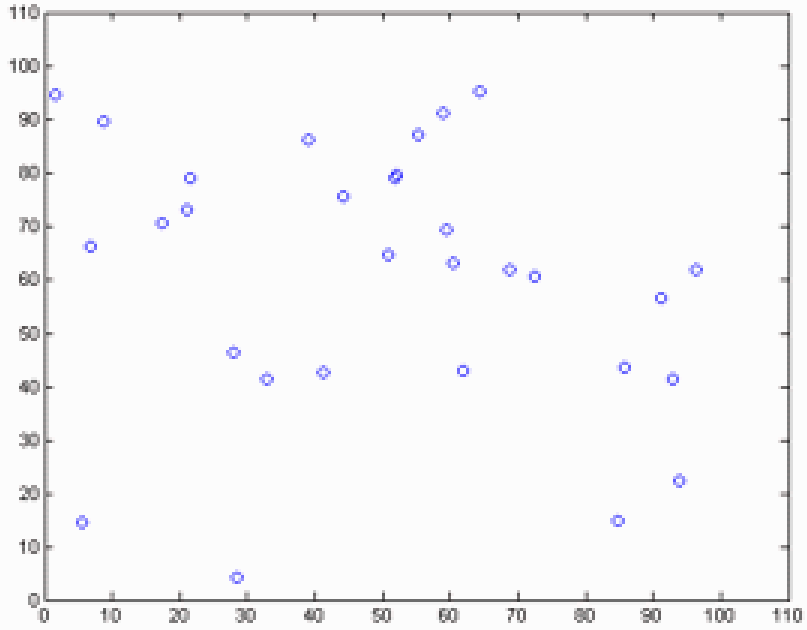


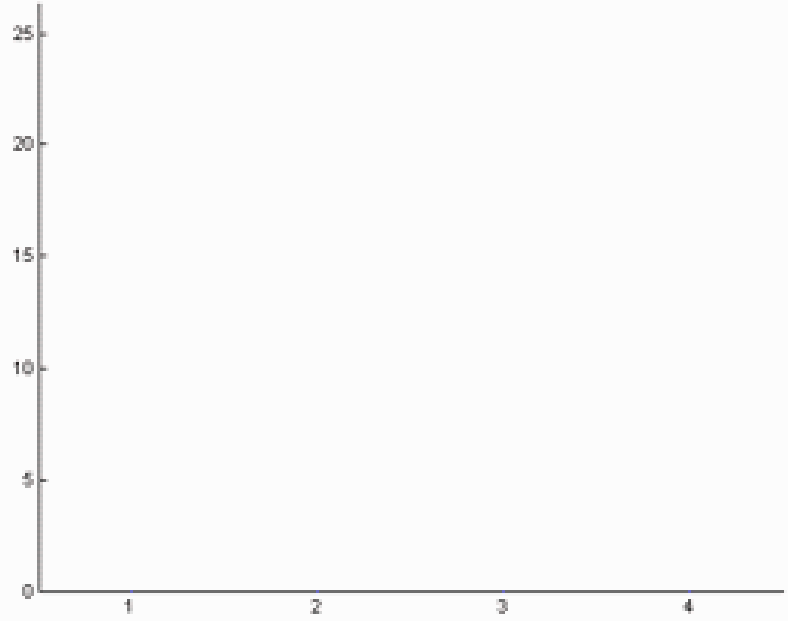




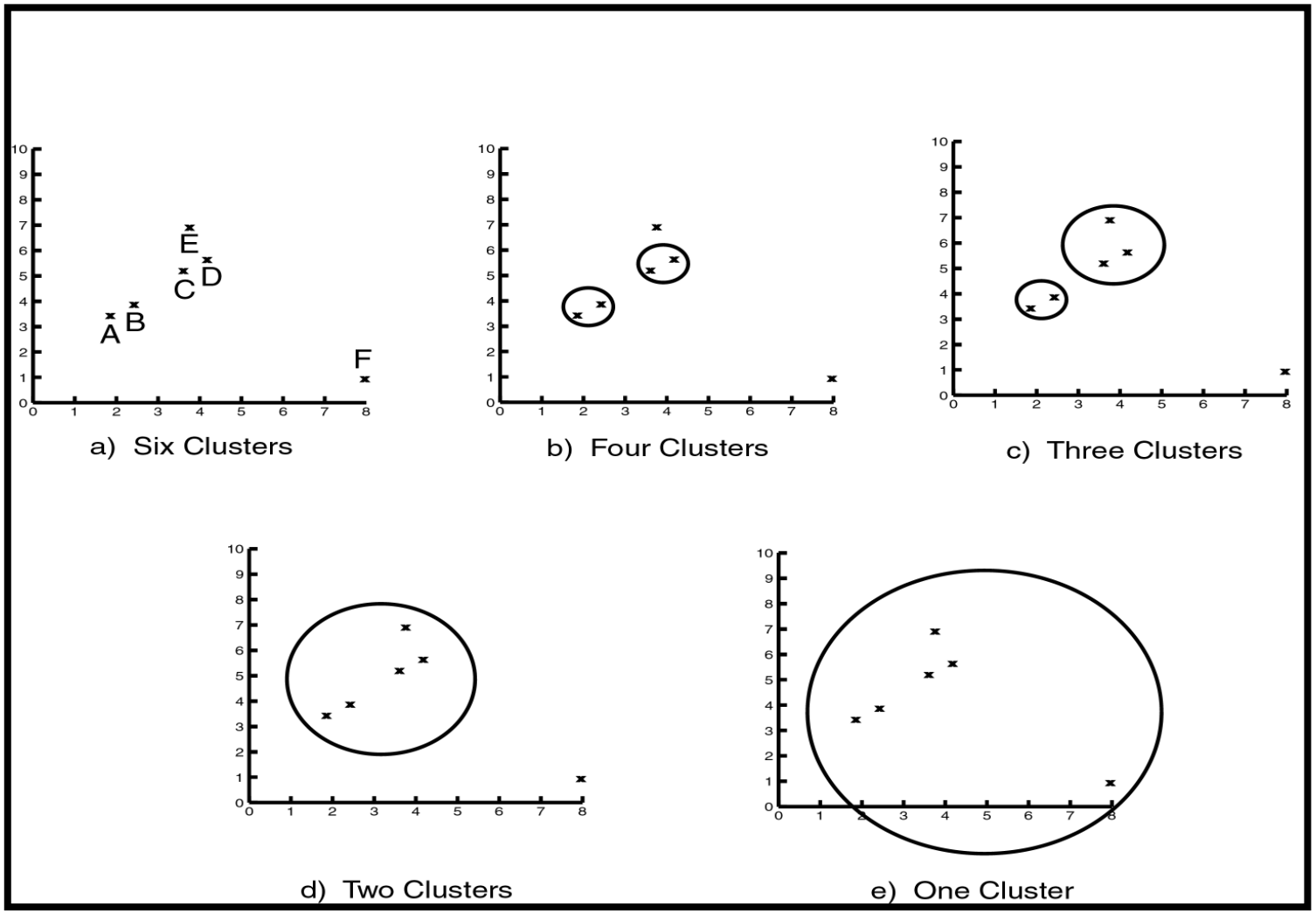
- бинарное дерево, описывающее все шаги разбиения
- Корень – общий кластер, листья - элементы
- «Высота» ветвей (до пересечения) – порог расстояния «склейки» («разделения»)
- Результат кластеризации – «срез» дендрограммы

Иерархическая кластеризация - Demo



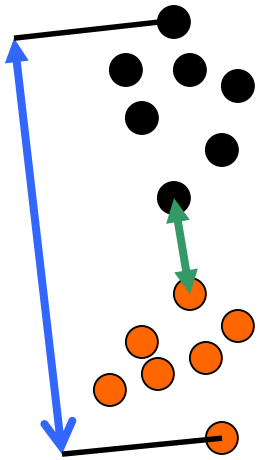


Уровни кластеризации



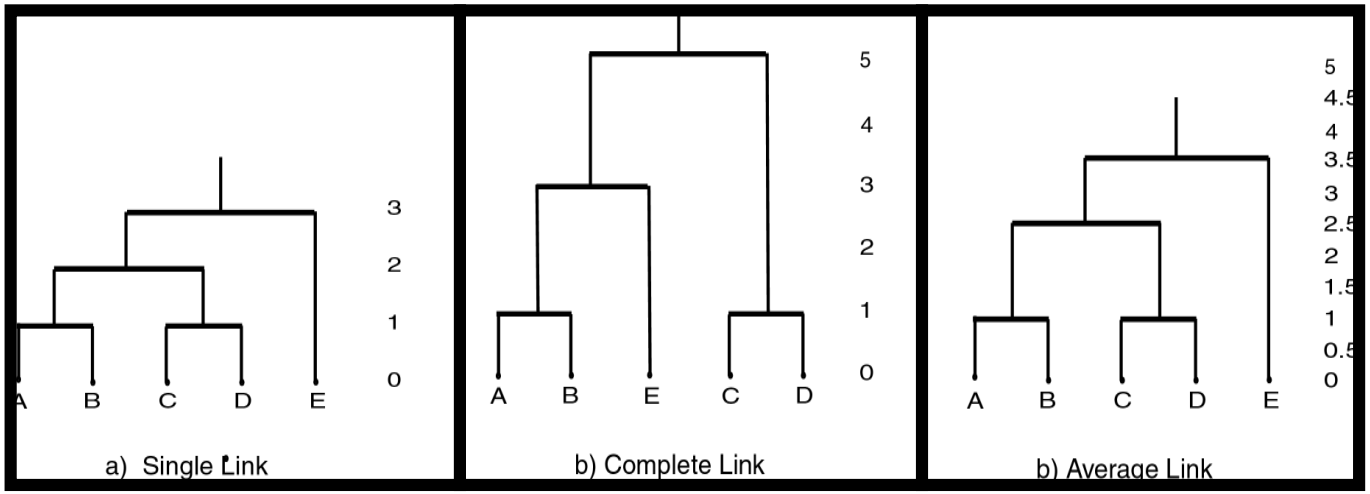
Оценка близости кластеров

- Расчет расстояния на основе попарных расстояний между элементами различных кластеров:
  - **Полное связывание**: наибольшее попарное расстояние. Дает компактные сферические кластеры.
  - **Среднее связывание**: усредненное попарное расстояние. Редко используется.
  - **Единственное связывание**: наименьшее попарное расстояние. Дает «растянутые» кластеры сложной формы.
  - **Центроидное связывание*:*** расстояние между центрами (мат. ожидание) кластеров.
  - Другие методы (например **метод Ward'a** – минимизирует внутрикластерные дисперсии или другую целевую функцию)



Многое зависит от расстояния

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



Обсуждение иерархической кластеризации

- Достоинства:
  - Очень просто и понятно, легко реализовать
  - Наглядные дендрограммы
  - Нет метапараметров
- Недостатки:
  - не масштабируется: временная сложность $O(n^2)$, где n - число кластеризуемых объектов
  - жадный алгоритм - локальная оптимальность с точки зрения минимизации внутриклассовых расстояний и максимизации межклассовых
  - относительно слабая интерпретируемость
  - поэтому нет компоненты в EM (но есть процедуры которые можно вызвать в коде)

Кластеризация на основе строгой группировки (partitioning):

- Основная задача:
  - Найти такое разбиение *C* исходного множества *X* из *N* объектов на *K* непересекающихся подмножеств C_k, покрывающих *X*, чтобы внутриклассовое расстояние было минимальным:

$$\min_{C_i \cap C_j = \varnothing, \bigcup C_i = X} \sum_{i=1}^{K} \sum_{x \in C_i} \sum_{x' \in C_i} d(x, x')$$

- Точное решение – перебор с отсечением
  - метод «ветвей и границ», но число комбинаций неприемлемо даже для 100 объектов:

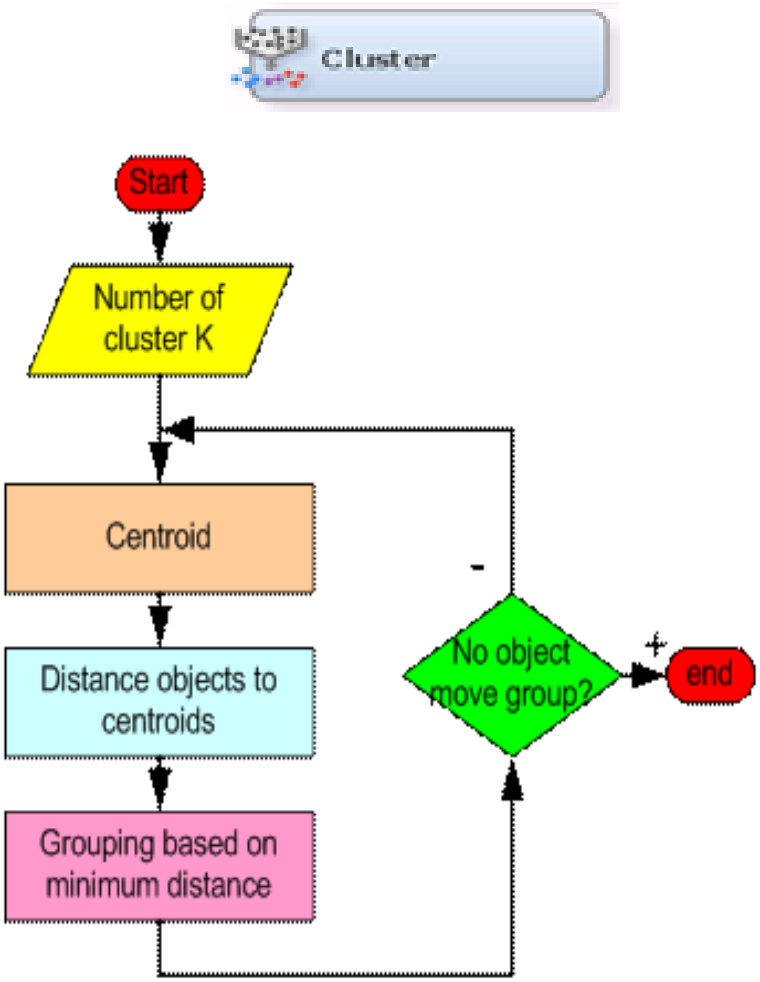
$$S(N, K) = \frac{1}{K!} \sum_{i=1}^{K} (-1)^{K-i} \binom{K}{i} K^N$$

- Эвристические методы:
  - K-means (прототип кластера – мат. ожидание *m*), K-medoids (прототип кластера – средний элемент)

$$\min_{C_i \cap C_j = \varnothing, \bigcup C_i = X} \sum_{i=1}^{K} \sum_{x \in C_i} d(m_i, x)$$

  - ищется локальный минимум

Метод K-Means в enterprise miner

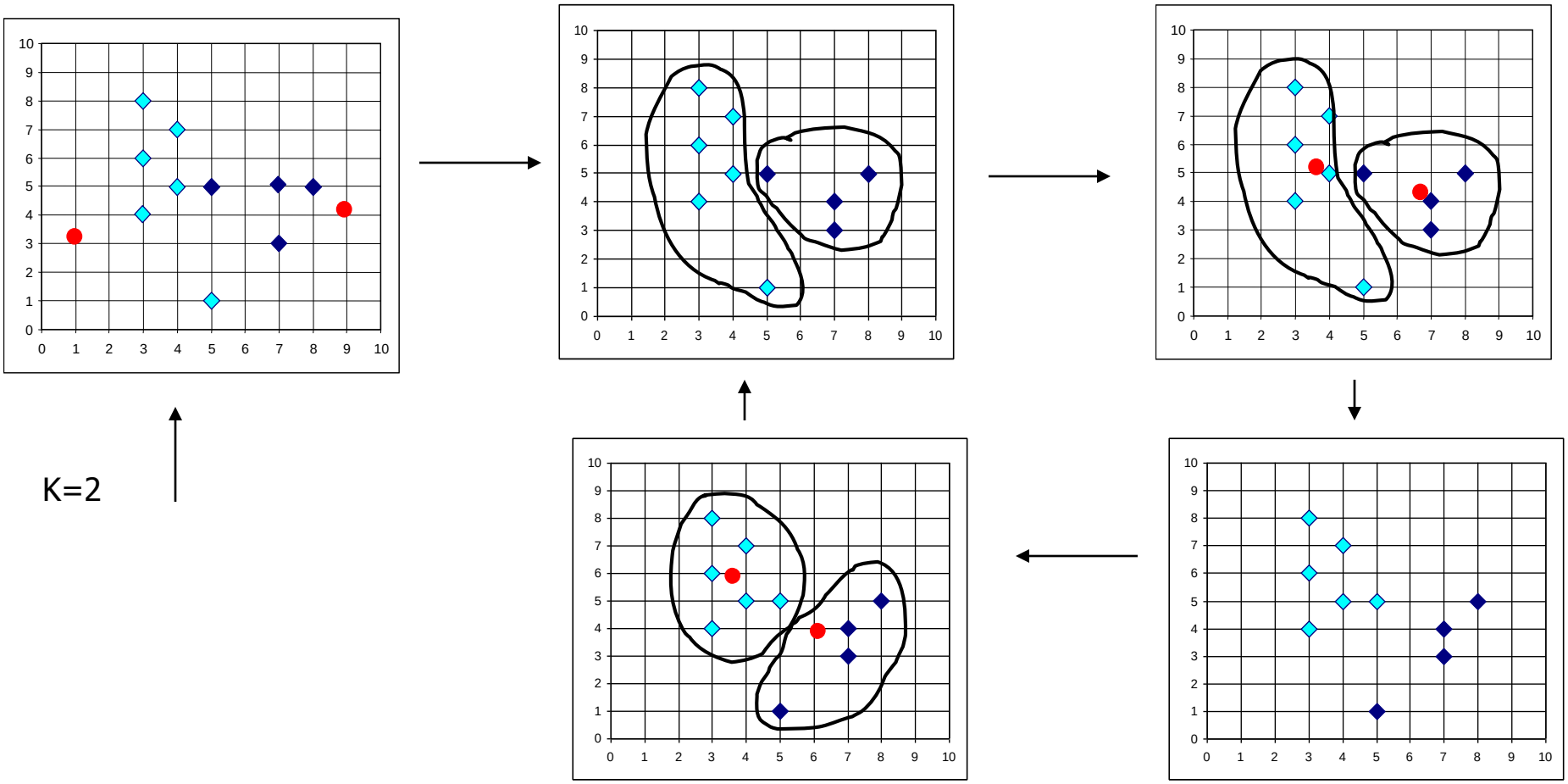


- Шаг 0. Инициализация:
  - произвольное разбиение на заданное число кластеров K (где значение K выбирается по CCC на основе иерархической кластеризации)
- Шаг 1. Поиск центров:
  - Для всех K кластеров $m_i = \sum_{x \in C_i} x \Big/ \|C_i\|$
- Шаг 2. Расчет расстояний до центров:
  - Для всех N объектов и K кластеров $d(m_i, x) = \sum_{x \in C_i} x \Big/ \|C_i\|$
- Шаг 3. Выбор ближайшего кластера:

$$x \in C_i \Leftrightarrow i = \min_j d(m_j, x)$$

- Если были перестановки, то Шаг 1.

Пример

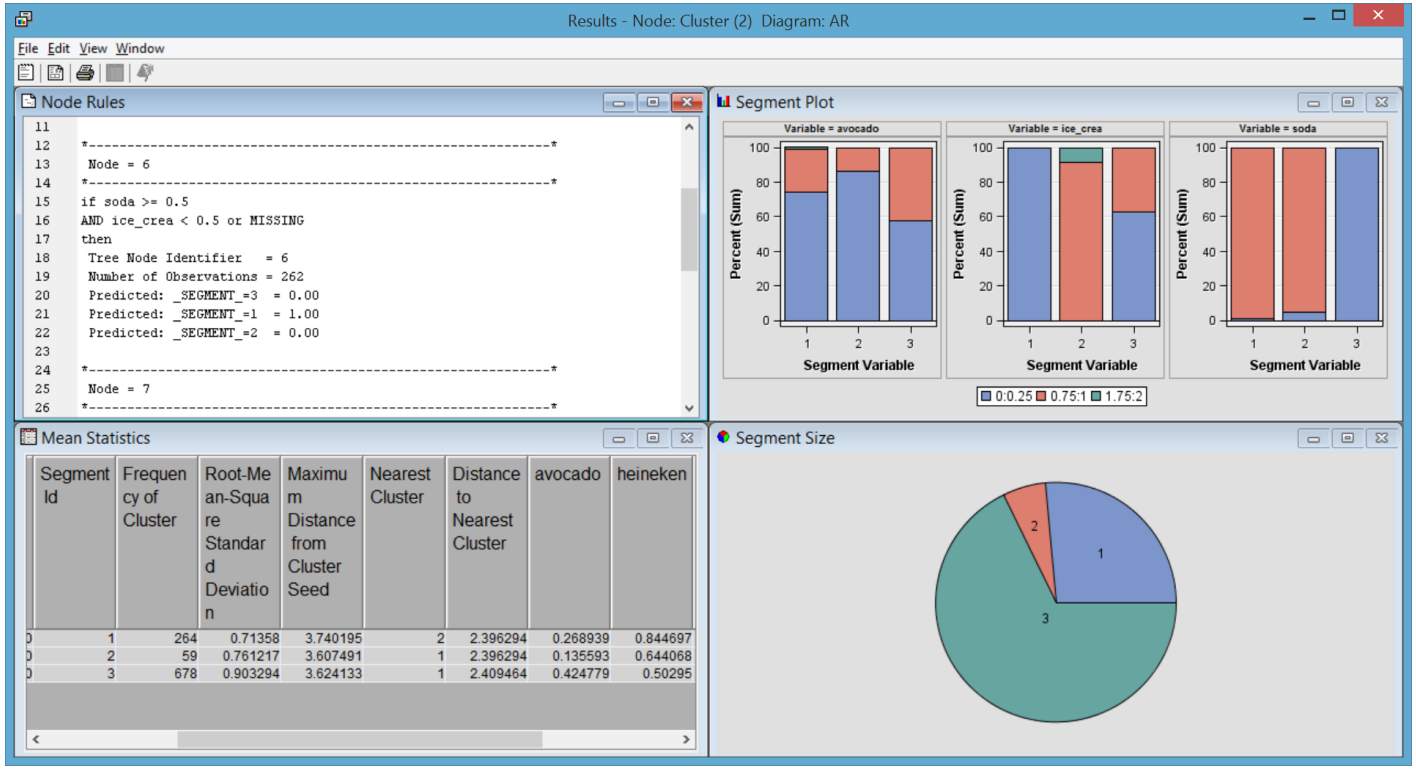


Особенности *K-Means*

- Достаточно быстрый
- Локальный экстремум:
  - Глобальный можно искать «разумным» перебором: имитация отжига, генетические алгоритмы и т.д.
  - В SAS на основе нескольких инициализаций
- Недостатки
  - Числовые данные (иначе как найти центр?) – моды, центр кластера – число (для числовых атрибутов)
  - Необходимо задавать K заранее (есть методы «отбора» K)
  - Чувствительность к шуму и выбросам, кластеры сферической формы

Использование в SAS EM

- Рассмотрим ту же задачу с рекомендательной системой:
  - Представили набор транзакций в виде матрицы частот использования продуктов
  - Применили Variable Clustering для удаления зависимых переменных
  - Построили кластеры

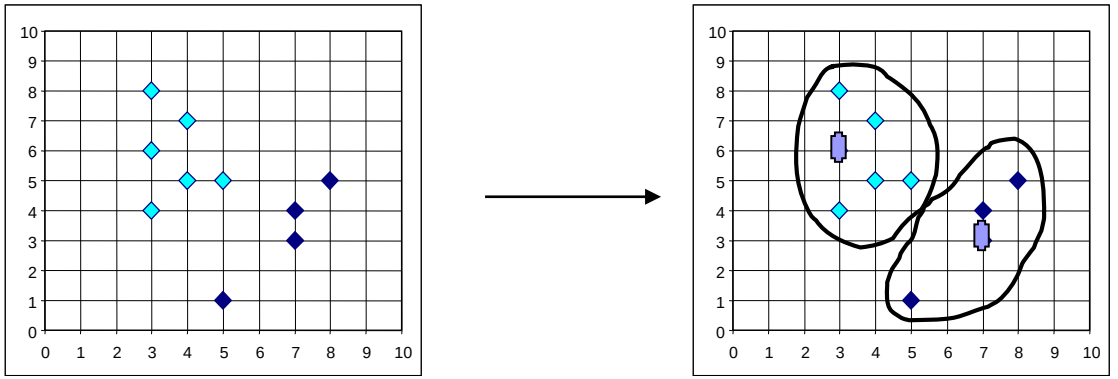


От K-Means к K-Medoids

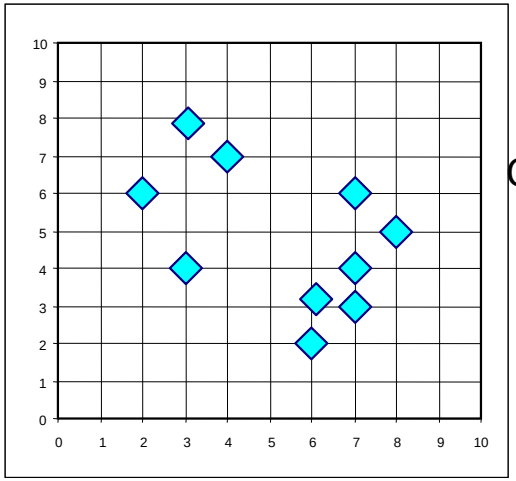
- Две главных беды k-means:
  - Чувствительность к выбросам и числовые данные
- K-Medoids:
  - Идея: вместо мат. ожидания кластера ищется представительный («наиболее центральный») объект – medoid
  - Процесс: случайная инициализация, переход к новому medoid, если это улучшает целевую функцию:

$$\min_{C_i \cap C_j = \varnothing, \bigcup C_i = X} \Sigma_{i=1}^{K} \Sigma_{x, m_i \in C_i} d(m_i, x)$$

- НО: не масштабируется и вычислительно неэффективный:
  - $O(K(N-K)^2)$ для каждой итерации

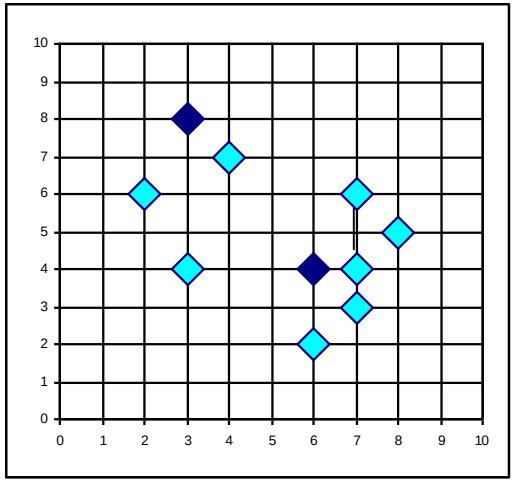


Алгоритм K-Medoids

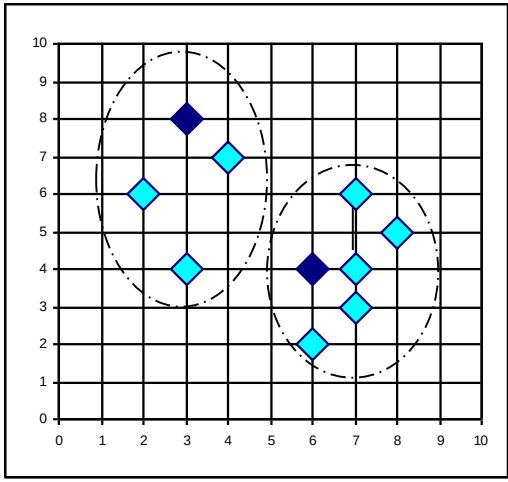


K=2

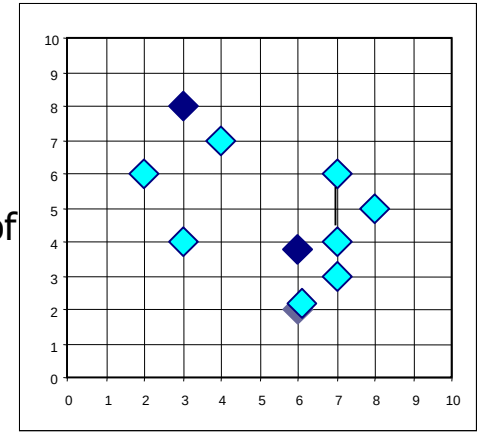
Случайн. K medoids



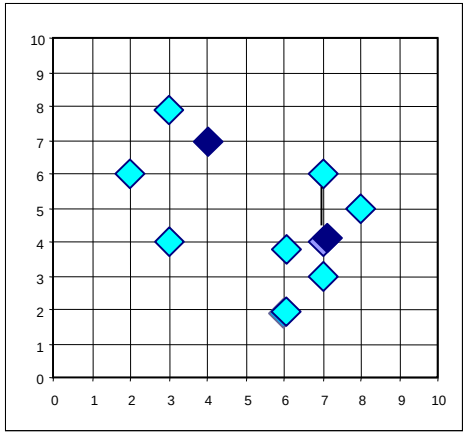
Каждый объект к ближ. medoid



Перебор всех кандидатов на замену medoid



Расчет стоимости перехода (total cost of swapping)



Если качество кластеризац. улучш., то переход к нов. medoid

Пока есть изменения

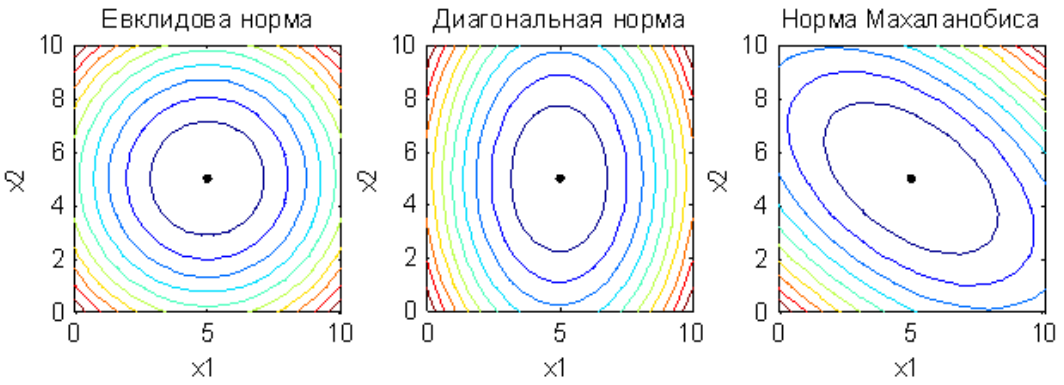
Fuzzy K-means

- Шаг 0. Инициализация:
  - случайная матрица *U* (но со всеми нормировками)
- Шаг 1. Поиск центров: для всех K кластеров

$$c_j = \sum_i (u_{ij})^m x_i \Big/ \sum_i (u_{ij})^m$$

- Шаг 2. Расчет расстояний до центров кластеров для всех объектов

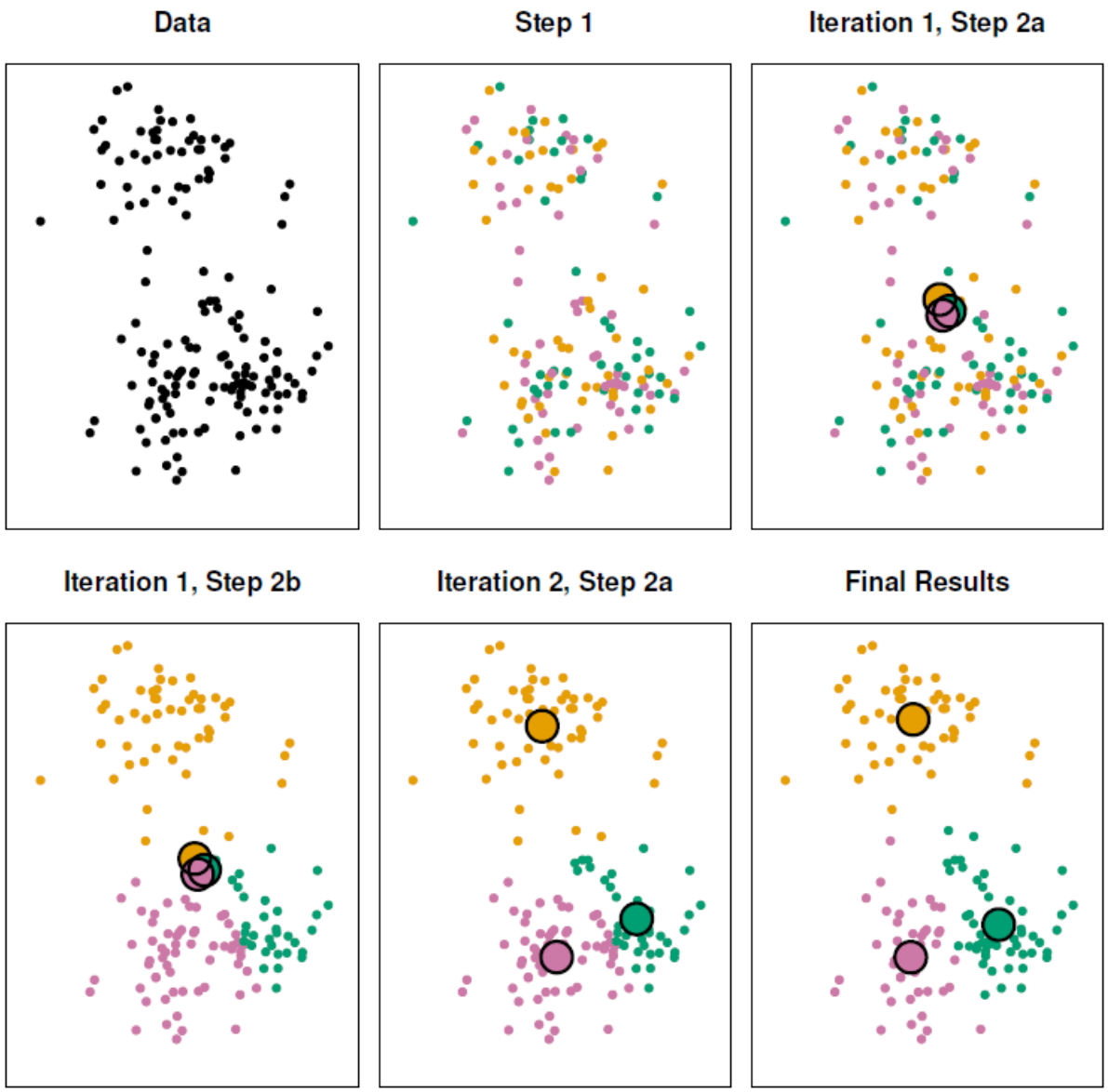
$$D_{ij} = \sqrt{\left\| x_i - c_j \right\|^2}$$



- Шаг 3. Перерасчет U*:

$$u_{ij}^{*} = \left(\frac{D_{ij}^{2}}{\sum_k D_{ik}^{2}} \right)^{-\frac{1}{m-1}}$$

- Если матрица сильно изменилась $\left\| U - U^{*} \right\| > \varepsilon$, то Шаг 1.


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

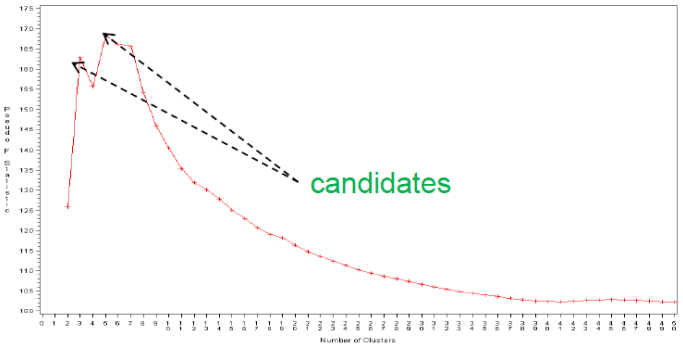
Определение числа кластеров

- По сути – перебор моделей с разным числом кластеров и выбор по некоторому критерию, например Pseudo-F (Calinski and Habarasz, 1974)

$$W_j = \sum_{i \in C_j} \| \mathbf{T}(v_i) - \overline{\mathbf{T}}_j \|^2 \qquad Q = \sum_{i=1}^{V} \| \mathbf{T}(v_i) - \overline{\mathbf{T}} \|^2$$

  - N- число объектов,
  - G – число кластеров,
  - W-сумма внутри-кластерных квадратов расстояний,
  - Q – сумма всех кв. расстояний до общего центра

$$pseudoF = \frac{(Q - \sum_{g=1}^{G} W_g)/(G-1)}{\sum_{g=1}^{G} W_g /(N-G)}$$

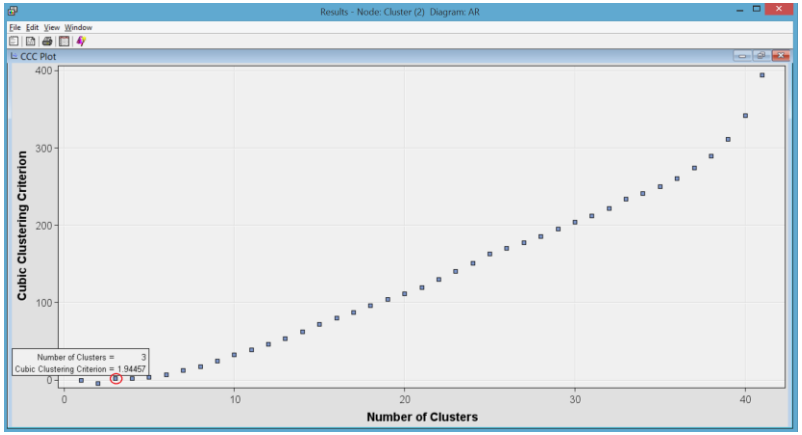


  - Выбирается вариант с максимальным *pseudoF*

Определение числа кластеров

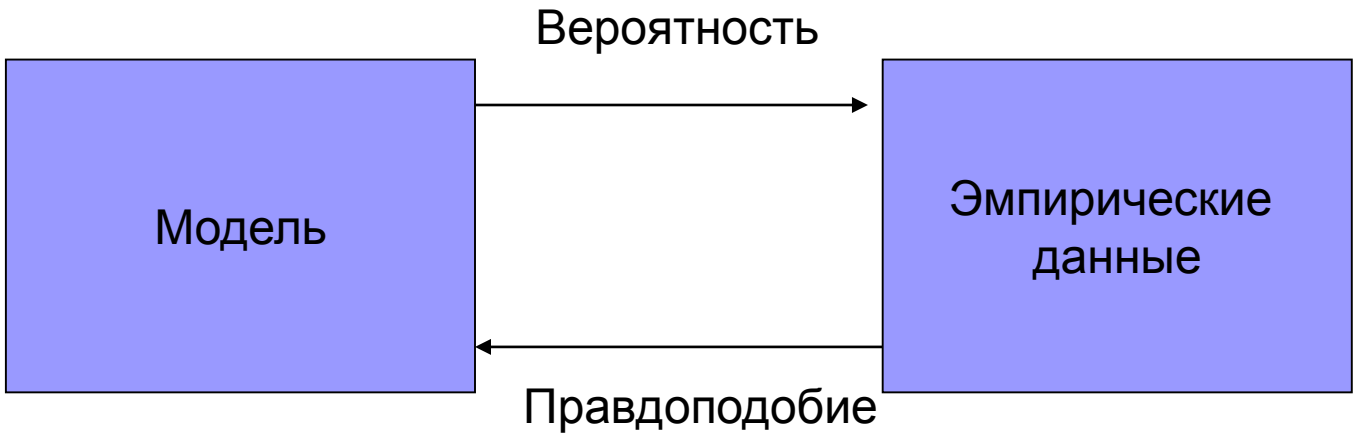
- SAS Cubic Clustering Criterion (CCC) (Sarle, 1983)
  - Основная идея: сравнение R^2 (для отображения матрицы данных с помощью индикаторной матрицы в прототипы кластеров) для заданной модели кластеризации с $E(R^2)$ для равномерно распределенного множества прототипов кластеров (как наихудший возможный вариант)

$$CCC = log\left[\frac{1-E(R^2)}{1-R^2}\right]\frac{\sqrt{np/2}}{(0.001+E(R^2))^{1.2}}$$



  - n = число наблюдений, p = число переменных, Y = nxp матрица данных (стандартизованных), M = qxp матрица центров кластеров, X = индикаторная матрица (x_{ik}=1 если *i* принадлежит *k*)
  - *T= Y'Y, B = ' X'X, W = T-B , R^2 = 1 – trace(W)/trace(T)*
  - Должно быть больше 2 и выбирается первый локальный максимум

Общая идея статистического подхода в кластеризации (и не только)



- Модель описывает процесс порождения эмпирических данных в терминах теории вероятности (например, плотность распределения для непрерывных данных)
- Наиболее общий подход оценки параметров модели через функцию правдоподобия

Функция правдоподобия

Формула Байеса

$$P(Model \mid Data) = \frac{P(Data \mid Model)P(Model)}{P(Data)}$$

Функция правдоподобия – совместное распределение выборки из параметрического распределения как функция параметра модели.

- Задача – найти «лучшую» модель, ту, которая наиболее точно описывает эмпирические данные.
- Функция правдоподобия – функция модели (параметров модели) при фиксированных данных.
- Спецификация модели – «искусство» (не процедура), фиксируется тип модели (например, распределение), а ищутся «лучшие» параметры.
- Для сложных задач модель = «смесь» (линейная комбинация) распределений.

Примеры

- Испытания Бернулли:
  - Пусть дана последовательность {0,1}: 0,1,1,0,0,1,1,0
  - Ищем модель в классе распределений B(p): $P(X = x) = p^x(1-p)^{1-x}$
  - Задача - найти наилучший параметр p: $\text{argmax}_p P(Data|B(p))$
- Смесь нормальных распределений:
  - Пусть дана выборка, например, рост людей:185,140,134,150,170 …
  - Предполагаем нормальное распределение, но в среднем женщины ниже мужчин, а значит – «смесь» распределений:

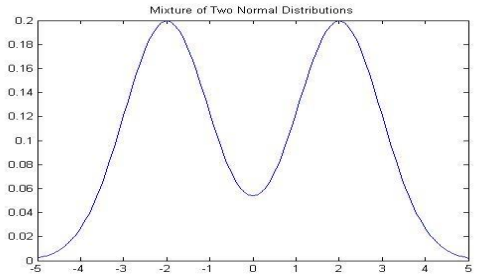
$$\pi_1 N(\mu_1, \sigma_1) + \pi_2 N(\mu_2, \sigma_2)$$

  - где:

$$N(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma}(x-\mu)^2}$$

  - Задача - найти параметры смеси:

$$\arg\max_{\pi_1,\pi_2,\mu_1,\mu_2,\sigma_1,\sigma_2} P(Data \mid \pi_1, \pi_2, \mu_1, \mu_2, \sigma_1, \sigma_2)$$



Оценка максимального правдоподобия

- Точечная оценка параметров, которая максимизирует функцию правдоподобия при фиксированной реализации выборки.
- Если $A_1,\ldots,A_n$ независимые одинаково распределенные сл. вел.:
$$P(A_1,\ldots,A_n) = \prod_{i=1}^{n} P(A_i)$$
- l(Data|Model) = log(P(Data|Model)) - логарифмическая функция правдоподобия:
  - log – монотонная функция, точки экстремумов совпадают
  - вместо произведения – сумма логарифмов
- Максимум функции правдоподобия в нуле производных:
  - Для всех параметров тета:
$$\frac{\partial l(X \mid \theta_1,\ldots,\theta_n)}{\partial \theta_i} = 0$$
  - Для нахождения максимума решаем полученную систему уравнений

Пример с распределением Бернулли

$$
\begin{aligned}
L(p) &= P(0,1,1,0,0,1,0,1|p) \\
&= P(0|p)P(1|p)\ldots P(1|p) \\
&= (1-p)p\ldots p \\
&= p^4(1-p)^4
\end{aligned}
$$

- Надо найти p, максимизирующий логарифмическое правдоподобие *l(p)= Log P(Data|B(p))*

$$
\begin{aligned}
\ell(p) &= logL(p) = 4log(p) + 4log(1-p) \\
\frac{d\ell(p)}{dp} &= \frac{4}{p} - \frac{4}{1-p} \equiv 0 \\
\rightarrow p &= \frac{1}{2}
\end{aligned}
$$

Скрытые (латентные) переменные и EM алгоритм

- Задача кластеризации как оценка скрытых (латентных) переменных – «меток» кластеров:
  - Пусть Data = {x(1),x(2),…x(n)} - набор сл. векторов «наблюдений»
  - Пусть H = {z(1),z(2),..z(n)} - множество значений соответствующих значений скрытой величины Z, где z(i) соответствует x(i)
  - Считаем, что Z – дискретна.
- Оценка скрытых (ненаблюдаемых) величин – цель EM
- Приложения EM:
  - кластеризация
  - заполнение пропущенных значений в выборке
  - поиск скрытых состояний марковской модели

EM Алгоритм

- Логарифмическое правдоподобие наблюдаемых данных:

$$l(\theta) = \log p(D \mid \theta) = \log \sum_H p(D, H \mid \theta)$$

- Нужно оценить не только параметры θ, но и H
- Пусть Q(H) есть распределение вероятности скрытых величин
- Неравенство Дженсона (для выпуклой функции):

$$\varphi\left(\frac{\sum a_i x_i}{\sum a_i}\right) \leq \frac{\sum a_i \varphi(x_i)}{\sum a_i};$$

$$\begin{aligned}
\ell(\theta) &= log \sum_H p(D, H|\theta) \\
&= log \sum_H Q(H) \frac{P(D, H|\theta)}{Q(H)} \\
&\geq \sum_H Q(H) log \frac{P(D, H|\theta)}{Q(H)} \\
&= \sum_H Q(H) log p(D, H|\theta) + \sum_H Q(H) log \frac{1}{Q(H)} \\
&= F(Q, \theta)
\end{aligned}$$

- F(Q,θ) - нижняя граница *l(θ)*

EM Алгоритм

- Процедура EM (в цикле):
  - максимизация F по Q (тета зафиксированы)
  - максимизация F по тета (Q фиксировано)

E-step $Q^{k+1} = argmax_Q F(Q^k, \theta^k)$

M-step $\theta^{k+1} = argmax_\theta F(Q^{k+1}, \theta^k)$

- E-step: $Q^{k+1} = P(H|D, \theta^k)$
- M-step: $\theta^{k+1} = argmax_\theta \sum_H p(H|D, \theta^k) logp(D, H|\theta^k)$

EM алгоритм для смеси нормальных распределений

Смесь нормальных распределений

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x, \mu_k, \sigma_k)$$

E Step

$$P(k|x) = \frac{\pi_k f_k(x, \mu_k, \sigma_k)}{f(x)}$$

M-Step

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} P(k|x(i))$$

$$\mu_k = \frac{1}{n\pi_k} \sum_{i=1}^{n} P(k|x(i))x(i)$$

$$\sigma_k = \frac{1}{n\pi_k} \sum_{i=1}^{n} P(k|x(i))(x(i) - \mu_k)^2$$

- Похоже по сути на k-means, но:
  - Оценка не только мат. ожидания, но и дисперсии (размер кластера)
  - «перекрывающиеся» кластеры, лучше с выбросами

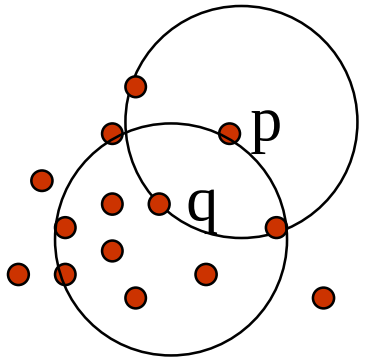
Кластеризация на основе связности

- Основные свойства:
  - Произвольная форма кластеров
  - Работа в условиях шума
  - Один проход базы
  - Недостаток: нужны параметры для «тонкой настройки»
- Популярные алгоритмы:
  - DBSCAN: Ester, et al. (KDD’96)
  - OPTICS: Ankerst, et al (SIGMOD’99).
  - DENCLUE: Hinneburg & D. Keim (KDD’98)
  - CLIQUE: Agrawal, et al. (SIGMOD’98)

Основные положения

- Важные параметры:
  - *Eps*: радиус области поиска ближайших соседей
  - *MinPts*: минимальное число ближайших соседей в Eps-области
- Множество ближайших соседей:
  - $N_{Eps}(p)$:{q | dist(p,q) <= Eps}
- Непосредственно достижимые точки:
  - Точка *p* непосредственно достижима из *q* с учетом. *Eps*, *MinPts,* если *p* принадлежит $N_{Eps}(q)$
- Ядровая точка:

$$|N_{Eps}(q)| >= MinPts$$

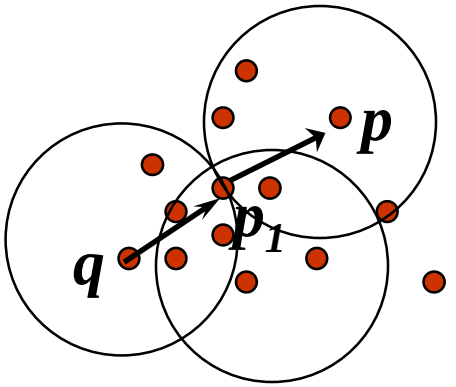


MinPts = 5

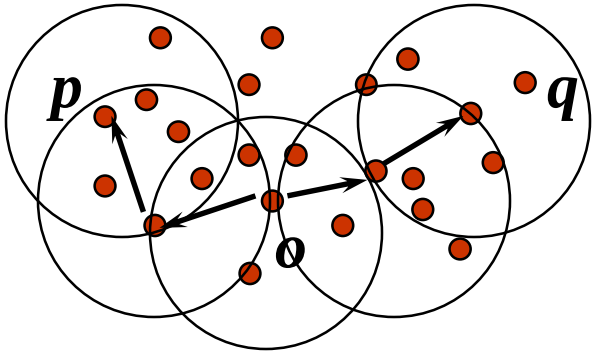
Eps = 1 cm

Основные положения

- Достижимость:
  - Точка *p* достижима из *q* с учетом *Eps*, *MinPts,* если существует путь p_1, …, p_n, p_1 = *q*, p_n = *p* такой, что p_{i+1} непосредственно достижима из p_i

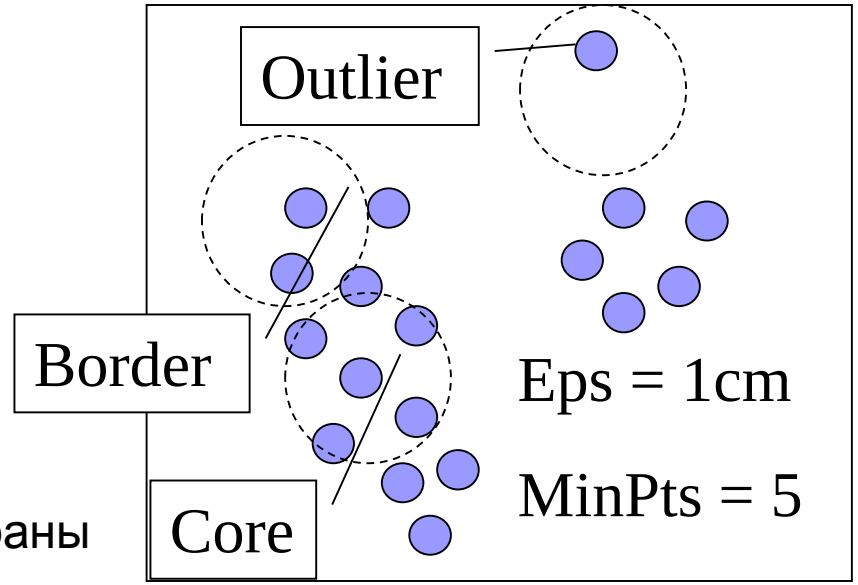


- Связность:
  - Точка *p* связана с *q* с учетом *Eps*, *MinPts,* если существует точка *o такая, что* обе точки *p* и *q* достижимы из *o* с учетом *Eps* и *MinPts*.



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Основан на понятии связного кластера:
  - Кластер определен как максимальное множество связных точек
- Позволяет находить кластеры произвольной формы в условиях шума
- Процедура:
  - Произвольный выбор точки *p*
  - Выбор всех достижимых из *p* точек с учетом *Eps* и *MinPts*.
  - Если *p* – ядровая, то кластер сформирован
  - Если *p* граничная или выброс, то обработка следующей точки
  - Продолжать пока не будут выбраны все точки



DENCLUE: использование функций плотности распределения

- DENsity-based CLUstEring by Hinneburg & Keim (KDD'98)
  - В SAS есть реализация, но в виде процедуры
- Основная идея:
  - Аппроксимация плотности распределения по методу Parzen window
  - Функция влияния (Influence function), а на самом деле - ядровая или потенциальная функция, определяет влияние точки на окружение:

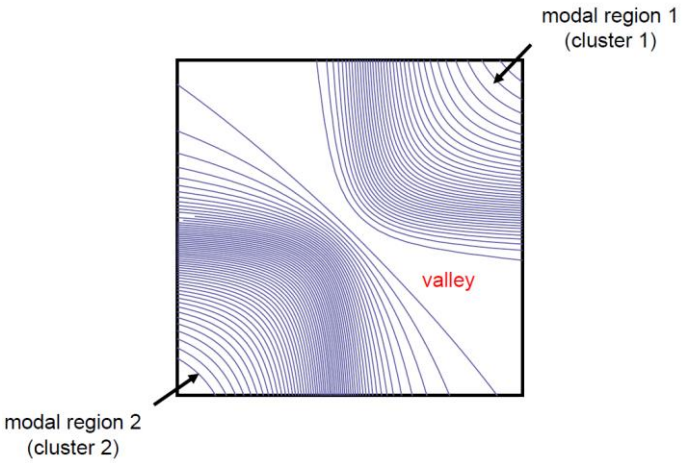
$$f_{Gaussian}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$$

  - Плотность в точке есть сумма всех функций окружения

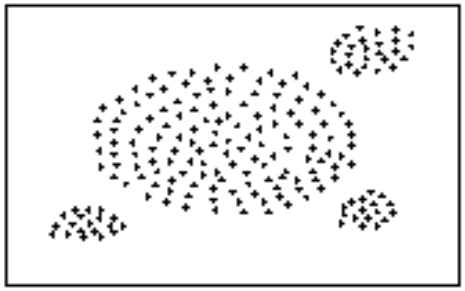
$$f^{D}_{Gaussian}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

  - Центры кластеров – локальные максимумы плотности (ищутся градиентным методом):

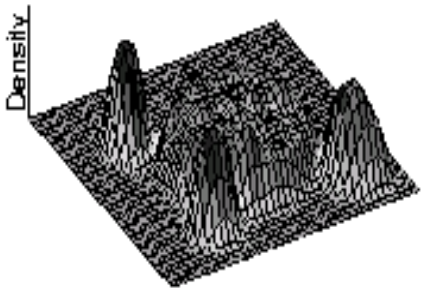
$$\nabla f^{D}_{Gaussian}(x,x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$



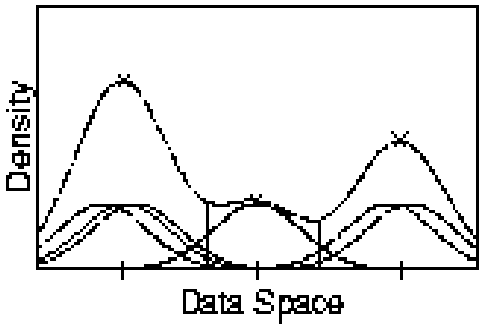
Кластеры DENCLUE



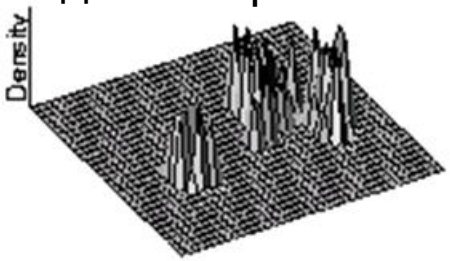
(a) Data Set



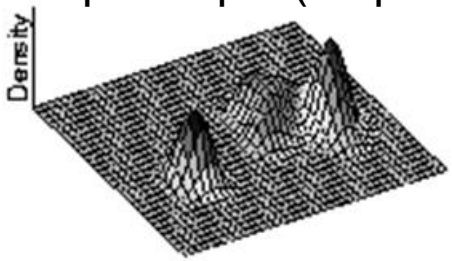
(c) Gaussian



- Основные достоинства:
  - Работает с шумом и большими объемами данных, достаточно быстрый метод
- НО:
  - нужно задавать критические параметры (ширину ядра)



(a) $\sigma = 0.2$



(b) $\sigma = 0.6$

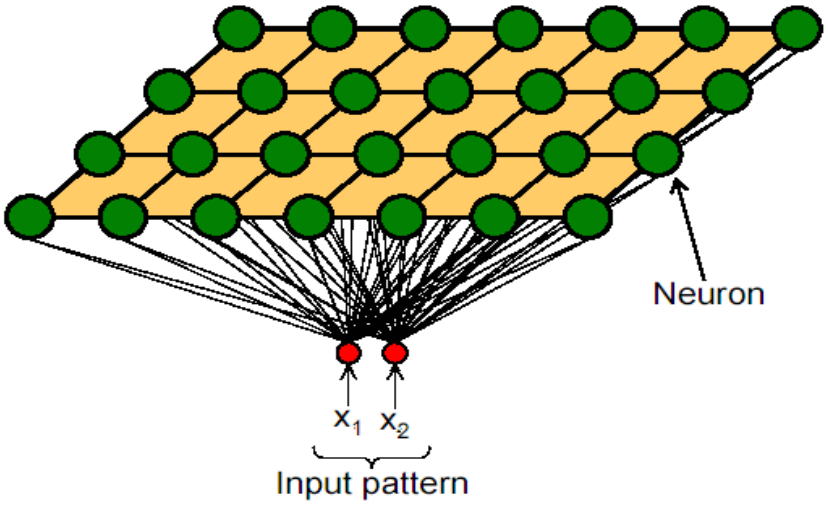
(d) $\sigma = 1.5$

  - только числовые данные, проблемы с интерпретируемостью

Self-Organizing Maps (SOM)

- Общая идея нейросетевого подхода (сети Кохонена):
  - Базируется на моделировании процесса обучения/запоминания в мозге
  - Каждый кластер (нейрон) определяется своим «прототипом» (число кластеров задается априори)
  - Прототипы (нейроны) объединены в виде 2D (реже 3D) решетки (сети) с квадратными (или шестигранными) ячейками
  - Структура решетки определяет понятие «окрестности» каждого прототипа (дискретное расстояние по решетке)
  - У прототипа кластера (нейрона) есть векторный «вес» – соответствует точке в исходном пространстве
  - Процесс активации – реакция на образ входного пространства, определяется мерой сходства между «весом» нейрона и входным образом (или расстоянием между прототипом кластера и объектом)
  - Конкурентное обучение: нейроны соревнуются за право активации (winner-takes-all, всегда один ближайший - победитель)

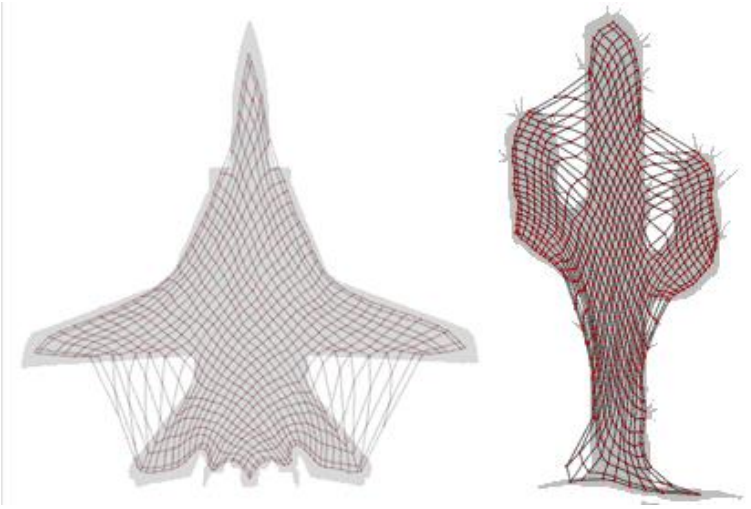
Основная задача SOM



- Задача:
  - формирование топографической карты входных образов, в которой пространственное расположение нейронов решетки (прототипов кластеров) в некотором смысле отражает статистические закономерности во входных параметрах.
- Или:
  - построение отображения многомерного исходного пространства на 2х (или 3х) мерную решетку с сохранением топологических зависимостей (близкие объекты исходного пространства будут рядом и на решетке).

Процедура работы SOM (неформально)

- Неформально:
  - Есть решетка нейронов *r x s,* с каждым узлом которой связан центр кластера исходного пространства $\mu_{1,1}, \ldots \mu_{r,s}$
  - Алгоритм SOM двигает центры кластеров в исходном многомерном пространстве, сохраняя топологию решетки
  - Точка исходного пространства относится к тому кластеру, чей вес ближе (расстояние до центра меньше)
  - При обработке новой точки центр кластера-победителя и всех его соседей по решетке сдвигается в сторону этой точки
- Упрощенный пример:
  - Добавляя точки из картинок, решетка «обтягивает» их контур
  - Небольшой обман, ибо тут размерность решетки и исходного пространства совпадают



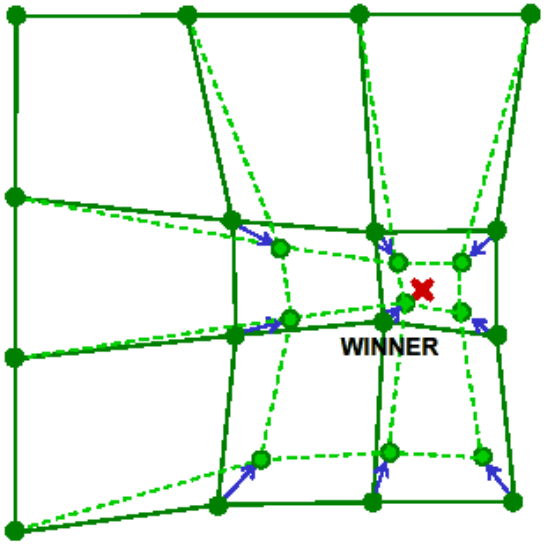
Процедура работы SOM

- Шаг 0. Инициализация:
  - структура решетки и число кластеров (нейронов)
  - инициализация «весов» прототипов wj(0) (полностью случайно или случайной выборкой из данных)
  - начальные параметры (скорость обучения и размер окрестности)
- Шаг 1. Выборка (итерация *t*):
  - Выбираем случайный *x(t)* из исходного пространства
- Шаг 2. Конкуренция:
  - Находим «лучший» нейрон для активации: $i(x) = \arg\min_j \left\| x(t) - w_j(t) \right\|$
- Шаг 3. Коррекция весов с учетом кооперации:
  - Для победителя и соседей по решетке пересчитываем их «вес» – двигаем их центры к точке *x* в исходном пространстве
  - Уменьшаем скорость обучения и размер окрестности
- Шаг 4. Проверка условий остановки и переход на Шаг 1.
  - Стабилизация структуры либо превышение числа выполненных итерации установленного значения

Коррекция весов с учетом кооперации

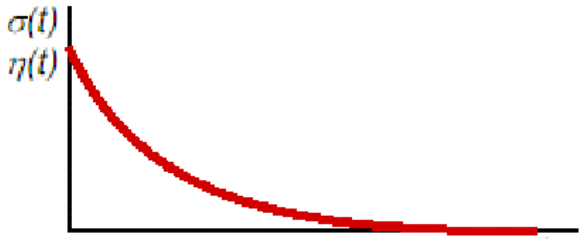
- Перерасчет весов победителя и соседей:
  - Стохастический градиентный спуск:

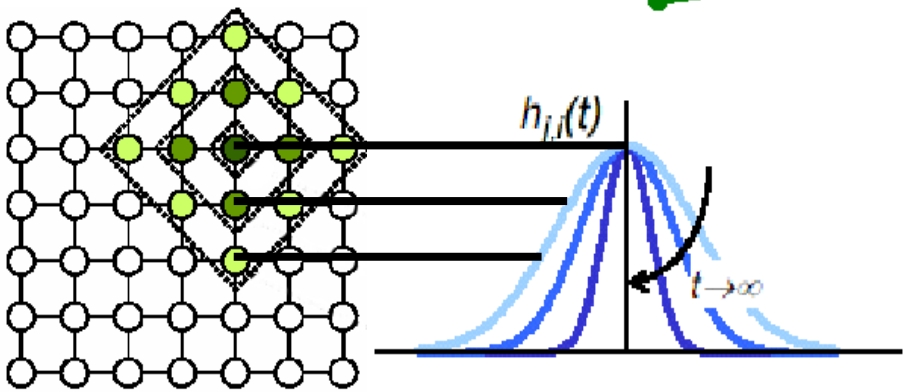
$$w_j(t+1) = w_j(t) + \eta(t) h_{ij(x)}(t)(x - w_j(t))$$



скорость обучения

размер топологической окрестности (на решетке!!!!)





$$\sigma(t+1) = \sigma_0 \exp\left(-\frac{t}{\tau_\sigma}\right)$$

$$\eta(t+1) = \eta_0 \exp\left(-\frac{t}{\tau_\eta}\right)$$

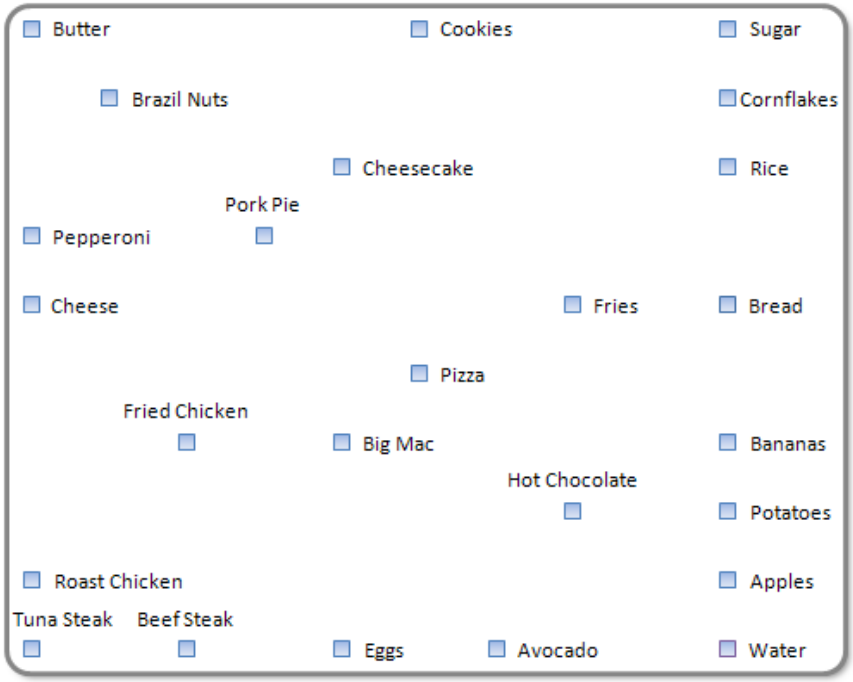
$$h_{ij(x)}(t) = \exp\left(-\frac{d^2_{grid}(i,j)}{2\sigma^2(t)}\right)$$

Пример

- Входные данные:

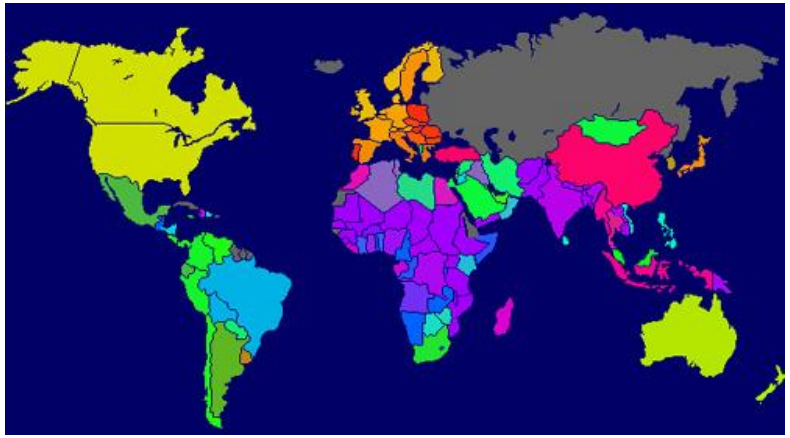
продукт	белки	углеводы	жиры
Apples	0.4	11.8	0.1
Avocado	1.9	1.9	19.5
Bananas	1.2	23.2	0.3
Beef Steak	20.9	0.0	7.9
Big Mac	13.0	19.0	11.0
Brazil Nuts	15.5	2.9	68.3
Bread	10.5	37.0	3.2
Butter	1.0	0.0	81.0
Cheese	25.0	0.1	34.4
Cheesecake	6.4	28.2	22.7
Cookies	5.7	58.7	29.3
Cornflakes	7.0	84.0	0.9
Eggs	12.5	0.0	10.8
Fried Chicken	17.0	7.0	20.0
Fries	3.0	36.0	13.0
Hot Chocolate	3.8	19.4	10.2
Pepperoni	20.9	5.1	38.3
Pizza	12.5	30.0	11.0
Pork Pie	10.1	27.3	24.2
Potatoes	1.7	16.1	0.3
Rice	6.9	74.0	2.8
Roast Chicken	26.1	0.3	5.8
Sugar	0.0	95.1	0.0
Tuna Steak	25.6	0.0	0.5

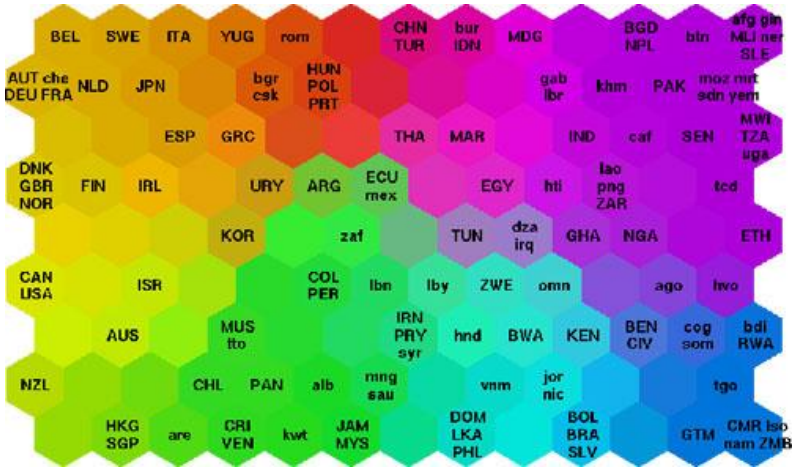
- SOM(10X10):



Визуализация и интерпретация SOM

- Когерентные области:
  - Близкие кластеры в исходном пространстве – рядом на решетке (свойство SOM) и одним (или спектрально близким) цветом
  - Группы кластеров – категории, области





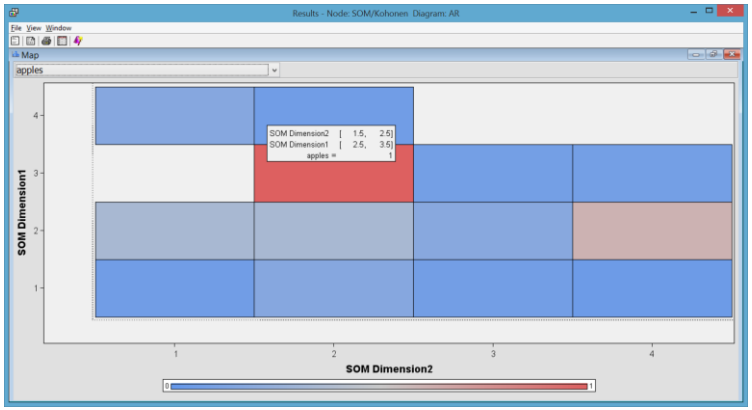
Свойства SOM

- Апроксимация входного пространства признаков
  - «Сжатие» информации, связь с методом LVQ (кластеризация на основе теории информации, задача - выбрать кодовые слова-кластеры так, чтобы минимизировать возможное искажение)
- Топологический порядок
  - Рядом в исходном пространстве => рядом на решетке и наоборот
- Соответствие плотности
  - Области исходного пространства с высокой плотностью отображаются в большие области на решетке и наоборот
- Выбор признаков:
  - осуществляет нелинейную дискретную аппроксимацию главных компонент (точнее главных кривых и плоскостей)
- Недостатки:
  - Алгоритм простой, но мат. анализу поддается плохо, в общем случае не доказана ни сходимость, ни даже устойчивость
  - Много неочевидных, но важных параметров, задаваемых априори, включая структуру решетки

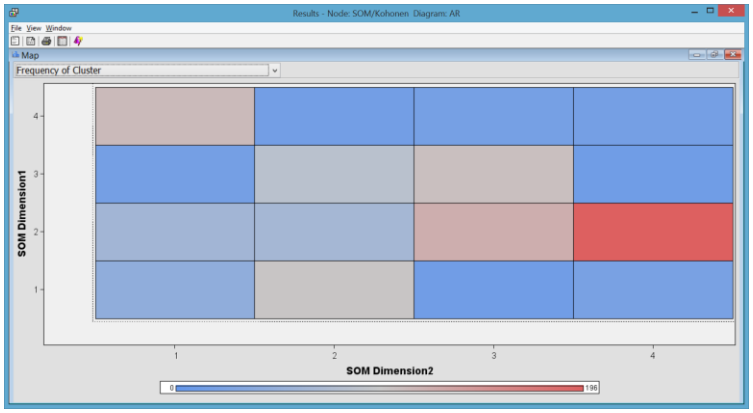
Использование в SAS EM

- Рассмотрим ту же задачу с рекомендательными системами, но без отбора переменных (для SOM это не так важно):

Распределение переменной (карта интенсивности)



Размеры кластеров (карта интенсивности)



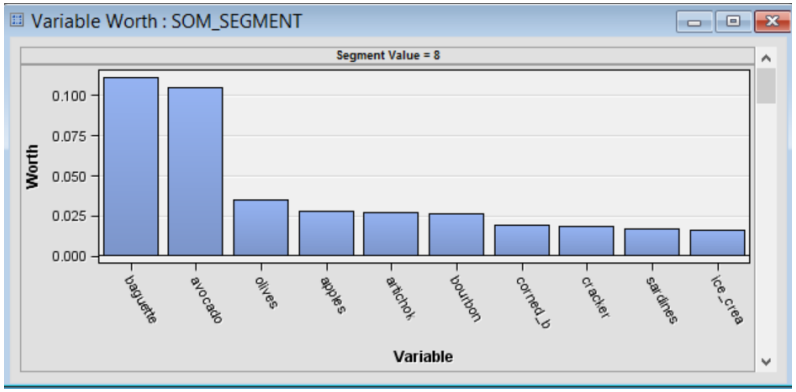
- В результирующие данные добавляется и номер кластера (ячейка решетки) и координаты на решетке (нелинейные главные компоненты):

EMWS5.SOM_TRAIN

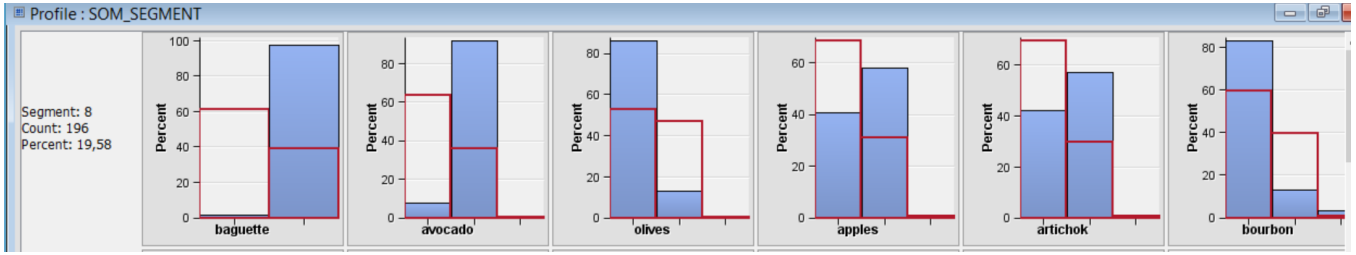
	ok	avocado	sardines	coke	peppers	apples	steak	chicken	SOM Segment ID	Distance	SOM Dimension1	SOM Dimension2	SOM ID
1		0.0	0.0	0.0	0.0	0.0	0.0	0.0	13.0	3.015675845416068	4.0	1.0	4:1
2		0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0	5.078695914955276	2.0	3.0	2:3
3		1.0	1.0	0.0	0.0	0.0	0.0	0.0	11.0	4.348182428722983	3.0	3.0	3:3
4		0.0	0.0	1.0	1.0	0.0	0.0	0.0	2.0	4.215271813354532	1.0	2.0	1:2

Визуализация

- Результаты кластеризации можно оценить с точки зрения:
  - Выделения важных переменных для каждого кластера



  - Различия распределения переменных внутри каждого кластера и по всей выборке



  - Реализовано в узле Segment Profile (раздел Assess) для любых алгоритмов кластеризации и группировки