# Лекция 4

Кластеризация

## Кластеризация

Кластер (англ. *cluster*) переводится как «группа», «скопление». Этот термин означает множество объектов функционально схожих между собой и собранных в одну связку.

Применительно к интеллектуальному анализу данных под кластерным анализом (data clustering) понимается задача разбиения выборки объектов на непересекающиеся подмножества, называемые кластерами, так чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались.

## Кластеризация

Кластеризация является задачей Data Mining, относящейся к стратегии «обучение без учителя», т.е. не требует наличия значения целевых переменных в обучающей выборке.

Синонимами термина «кластеризация» являются «*автоматическая классификация*», «*обучение без учителя*» и «*таксономия*» (taxonomy).

## Кластеризация

Кластеризация предназначена для разбиения совокупности объектов на однородные группы (кластеры или классы).

Обычно данные представляют собой выборки точек в пространстве признаков.

Наибольшее применение кластеризация первоначально получила в таких науках как биология, антропология, психология. Кластерный анализ полезен, когда нужно классифицировать большое количество информации.

Алгоритмы кластеризации являются в большой степени *эвристическими*.

Эвристический алгоритм – это алгоритм решения задачи, правильность которого для всех возможных случаев не доказана, но про который известно, что он даёт достаточно хорошее решение в большинстве случаев.

В действительности может быть даже известно, что эвристический алгоритм формально неверен.

Его всё равно можно применять, если при этом он даёт неверный результат только в отдельных, достаточно редких и хорошо выделяемых случаях или же даёт неточный, но всё же приемлемый результат.

Проще говоря, эвристика – это не полностью математически обоснованный (или даже «не совсем корректный»), но при этом практически полезный алгоритм.

Важно понимать, что эвристика, в отличие от корректного алгоритма решения задачи, обладает следующими особенностями:

- Она не гарантирует нахождение лучшего решения.
- Она не гарантирует нахождение решения, даже если оно заведомо существует (возможен «пропуск цели»).
- Она может дать неверное решение в некоторых случаях.

## Формальная постановка задачи кластеризации

Пусть $\mathbf{X} = \{x^1, x^2, \dots, x^N\}$ – множество $N$ объектов, заданных в *n*-мерном векторном пространстве признаков:

$$x^k = \left(x_1^k, x_2^k, \dots, x_n^k\right)^T, k = \overline{1, N}.$$

Пусть $\mathbf{Y} = \{1,2,3, \dots\}$ – множество номеров (имён, меток) кластеров.

## Формальная постановка задачи кластеризации

Считаем, что между объектами задана функция расстояния $\boldsymbol{\rho}(x, x')$.

Чаще всего используется евклидова метрика (расстояние):

$$\boldsymbol{\rho}_2(x, x') = \sqrt{\sum_{j=1}^{n} \left(x_j - x'_j\right)^2}$$

## Формальная постановка задачи кластеризации

Используются также:

- манхэттенское расстояние

$$\boldsymbol{\rho}_1(x, x') = \sum_{j=1}^{n} \left| x_j - x'_j \right|$$

- расстояние Чебышёва

$$\boldsymbol{\rho}_\infty(x, x') = \max_{j \in [1;n]} \left| x_j - x'_j \right|$$

## Формальная постановка задачи кластеризации

Все приведенные выше метрики являются частными случаями метрики Минковского:

$$\boldsymbol{\rho}_p(x, y) = \left( \sum_{j=1}^{n} |x_j - y_j|^p \right)^{1/p} .$$

Стоит также упомянуть расстояние Махаланобиса:

## Формальная постановка задачи кластеризации

Требуется разбить конечную выборку объектов $\mathbf{X}$ на $K$ непересекающихся подмножеств

$$\mathbf{S}_k, k = \overline{1, K};\ \mathbf{X} = \bigcup_{k=1}^{K} \mathbf{S}_k ,$$

называемых кластерами, так чтобы каждый кластер состоял из объектов, близких по метрике $\boldsymbol{\rho}$, а объекты разных кластеров существенно отличались.

## Кластеризация

При этом каждому объекту $x^i \in \mathbf{X}$ приписывается номер кластера $y_i \in \mathbf{Y}$.

Алгоритм кластеризации – это функция $\mathbf{X} \to \mathbf{Y}$, которая любому объекту $x \in \mathbf{X}$ ставит в соответствие номер кластера $\mathbf{y} \in \mathbf{Y}$.

## Кластеризация

Множество $\mathbf{Y}$ в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров с точки зрения того или иного критерия качества кластеризации.

Кластеризация (*обучение без учителя*) отличается от классификации (*обучения с учителем*) тем, что метки исходных объектов $y_i$ изначально не заданы и даже может быть неизвестно само множество $\mathbf{Y}$, т.е. число кластеров.

Центром кластера $\mathbf{S}_k$ (центроидом) называется геометрический центр точек $k$-го кластера в евклидовом пространстве:

$$X_k = \frac{1}{|\mathbf{S}_k|} \sum_{x^i \in \mathbf{S}_k} x^i,$$

где $|\mathbf{S}_k| = \mathbf{card}\, \mathbf{S}_k$ – число точек в $k$-ом кластере, $k = \overline{1, K}$; $K$ – число кластеров.

Дисперсия кластера $\mathbf{S}_k$ – это мера рассеяния точек в пространстве относительно центра кластера:

$$D_k = \frac{1}{|\mathbf{S}_k|} \sum_{x^i \in \mathbf{S}_k} \boldsymbol{\rho}^2\left(x^i, X_k\right).$$

Радиус кластера $\mathbf{S}_k$ – также мера рассеяния точек относительно центра кластера – максимальное расстояние до центра кластера:

$$R_k = \max_{x^i \in \mathbf{S}_k} \boldsymbol{\rho}\left(x^i, X_k\right).$$

# Кластеризация

Цели кластеризации могут быть различными в зависимости от особенностей конкретной прикладной задачи:

- Понять структуру множества объектов $\mathbf{X}$, разбив его на группы схожих объектов. Упростить дальнейшую обработку данных и принятия решений, работая с каждым кластером по отдельности (стратегия «разделяй и властвуй»).
- Сократить объём хранимых данных в случае сверхбольшой выборки $\mathbf{X}$, оставив по одному наиболее типичному представителю от каждого кластера.
- Выделить нетипичные объекты, которые не подходят ни к одному из кластеров. Эту задачу называют одноклассовой классификацией, обнаружением нетипичности или новизны (novelty detection).

## Кластеризация

В первом случае число кластеров стараются сделать поменьше.

Во втором случае важнее обеспечить высокую степень сходства объектов внутри каждого кластера, а кластеров может быть сколько угодно.

В третьем случае наибольший интерес представляют отдельные объекты, не вписывающиеся ни в один из кластеров.

Во всех этих случаях может применяться иерархическая кластеризация, когда крупные кластеры дробятся на более мелкие, те, в свою очередь, дробятся ещё мельче и т. д.

Такие задачи называются задачами таксономии (*taxonomy*).

Результатом таксономии является не простое разбиение множества объектов на кластеры, а древообразная иерархическая структура.

## Кластеризация

В этом случае вместо номера кластера объект характеризуется перечислением всех кластеров, которым он принадлежит: от крупного к мелкому.

Таксономии строятся во многих областях знания, чтобы упорядочить информацию о большом количестве объектов.

## Кластеризация

Решение задачи кластеризации принципиально неоднозначно по следующим причинам:

1. Не существует однозначно наилучшего критерия качества кластеризации. Все они могут давать разные результаты. Следовательно, для определения качества кластеризации требуется эксперт предметной области, который бы мог оценить осмысленность выделения кластеров.

2. Число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием.

3. Результат кластеризации существенно зависит от метрики, выбор которой, как правило, также субъективен и определяется экспертом.

Но стоит отметить, что есть ряд рекомендаций к выбору мер близости для различных задач.

## Кластеризация

К настоящему времени разработано более сотни различных алгоритмов кластеризации, в результате применения которых получаются неодинаковые результаты, что объясняется особенностью работы того или иного алгоритма, ориентированного на решение конкретной задачи.

## Метод *k*-средних (*k*-means)

Наиболее популярным является метод ***k-средних***.

Алгоритм относится к классу эвристических EM-алгоритмов (англ. *Expectation - maximization*), используемых в математической статистике для нахождения оценок максимального правдоподобия параметров вероятностных моделей.

## Метод *k*-средних (*k*-means)

Каждая итерация алгоритма состоит из двух шагов.

На E-шаге (expectation) вычисляется ожидаемое значение функции правдоподобия. На M-шаге (maximization) вычисляется оценка максимального правдоподобия, увеличивая ожидаемое правдоподобие, вычисляемое на E-шаге. Затем это значение используется для E-шага на следующей итерации.

Алгоритм выполняется до сходимости.

## Метод *k*-средних (*k*-means)

Основная идея алгоритма заключается в минимизации суммарного квадратичного отклонения точек кластеров от центров этих кластеров, то есть

$$J = \sum_{k=1}^{K} \sum_{x^i \in \mathbf{S}_k} \boldsymbol{\rho}^2\left(x^i, X_k\right) \rightarrow \min_{\mathbf{S}}$$

где $K$ – известное число кластеров.

## Метод *k*-средних (*k*-means)

Алгоритм заключается в том, что на каждой итерации заново вычисляется центр масс $X'_k$ для каждого кластера, полученного на предыдущем шаге; затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике:

$$x^i \in \mathbf{S}'_k, \text{ если } \boldsymbol{\rho}\left(x^i, X_k\right) = \min_{k=\overline{1,K}}.$$

Алгоритм завершается, когда на какой-то итерации не происходит изменения центра масс кластеров.

# Алгоритм кластеризации *k*-средних

**Начало**

Задать начальное приближение центров всех $K$ кластеров $X_k^{(0)}, k \in \{1,2,\dots K\}$.

**Repeat**

Отнести каждый объект к ближайшему центру, разбивая пространство признаков и формируя новые кластеры $\mathbf{S}_k^{(j)}$:

$$\forall x^i : k = \arg\min_{k \in 1,K} \boldsymbol{\rho}\left(x^i, X_k^{(j-1)}\right), \qquad \mathbf{S}_k^{(j)} = \mathbf{S}_k^{(j)} \bigcup x^i .$$

Вычислить новое положение центров:

$$X_k^{(j)} = \frac{1}{\left|\mathbf{S}_k^{(j)}\right|} \sum_{x^i \in \mathbf{S}_k^{(j)}} x^i .$$

Перейти к следующей итерации: $j = j + 1$.

**Until**

Пока состав кластеров $\mathbf{S}_k^{(j)}$ не перестанет изменяться.

## Недостатки алгоритма k-средних:

- Не гарантируется достижение глобального минимума суммарного квадратичного отклонения $J$, а только одного из локальных минимумов.
- Результат зависит от начального выбора центров кластеров $\left\{X_k^{(0)}\right\}$, их оптимальный выбор неизвестен.
- Число кластеров $K$ надо знать заранее.

## Методы инициализации:

- Метод Forgy. Из имеющегося набора данных случайным образом выбираются $K$ наблюдений.
- Случайное разбиение. Каждому из наблюдений на начальном этапе случайным образом присваивается номер кластера.
- Алгоритм k-means++. Из имеющегося набора данных случайным образом выбирается одна точка (первый центроид). Затем следующая точка выбирается из оставшихся с вероятностью, пропорционально зависящей от квадрата расстояния от точки до ближайшего центроида. Итерации повторяются до тех пор, пока не будут выбраны $K$ центроидов.

Вариации алгоритма *k*-средних:

- Метод k-медиан (k-medians). Центроиды выбираются вычислением медиан, а не средних значений.
- Метод k-медоидов (k-medoids). Центроиды всегда выбираются из объектов выборки. Минимизируется не квадрат евклидовой меры, а сумма различий между объектами (обычно, расстояние Минковского).
- Метод k-means++. Изменен алгоритм инициализации.
- … и другие.

## Алгоритм DBSCAN

DBSCAN - Density-based spatial clustering of applications with noise - Основанная на плотности пространственная кластеризация для приложений с шумами.

DBSCAN является одним из наиболее часто используемых алгоритмов кластеризации, и наиболее часто упоминается в научной литературе.

## Алгоритм DBSCAN

Если дан набор точек в некотором пространстве, алгоритм

- группирует вместе точки, которые тесно расположены (точки со многими близкими соседями),
- помечая как выбросы точки, которые находятся одиноко в областях с малой плотностью (ближайшие соседи которых лежат далеко).

## Алгоритм DBSCAN

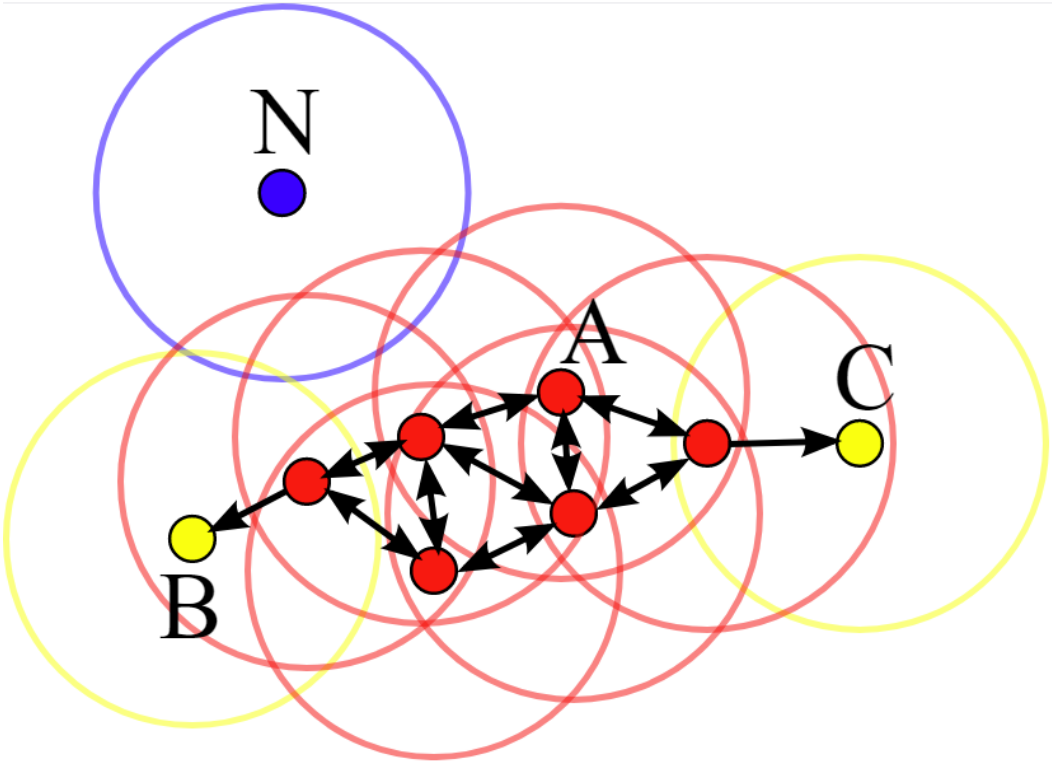
Рассмотрим набор точек, требующий кластеризации. Для выполнения DBSCAN точки делятся на *основные точки*, *достижимые по плотности точки* и *выпадающие* следующим образом:

- Точка $p$ является основной точкой, если по меньшей мере $minPts$ точек (включая саму точку $p$) находятся на расстоянии, не превосходящем $\varepsilon$. Говорят, что эти точки достижимы прямо из $p$.

## Алгоритм DBSCAN

- Точка $q$ прямо достижима из $p$, если она находится на расстоянии, не большем $\varepsilon$, от точки $p$, и $p$ должна быть основной точкой.
- Точка $q$ достижима из $p$, если имеется путь $p_1(p), \ldots, p_{n-1}, p_n(q)$, где каждая точка $p_{i+1}$ достижима прямо из $p_i$ (все точки на пути должны быть основными, за исключением $q$).
- Все точки, не достижимые из основных точек, считаются выбросами.

## Алгоритм DBSCAN



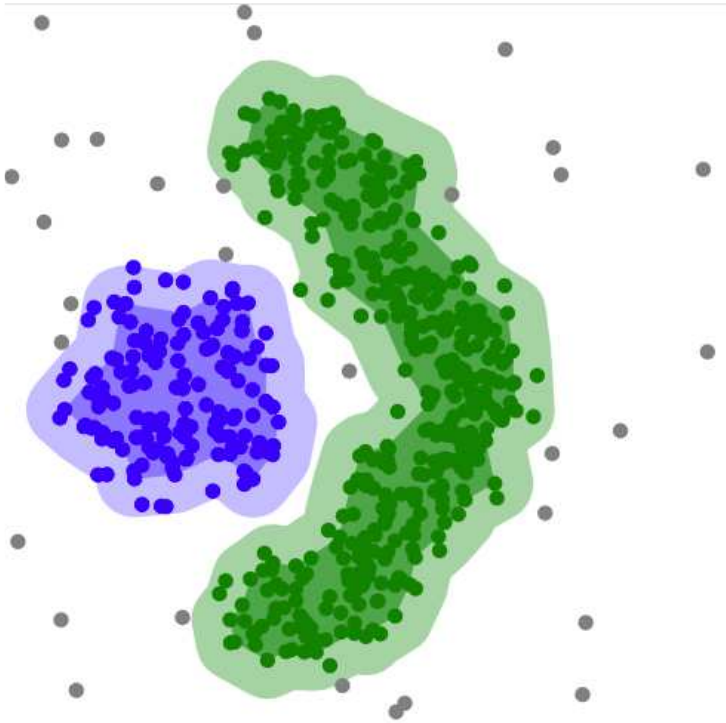
## Алгоритм DBSCAN

Абстрактно алгоритм можно представить как последовательность следующих этапов:

- Найти точки в $\varepsilon$ -окрестности каждой точки и определить основные точки с более чем $minPts$ соседями.
- Найти связные компоненты основных точек на графе соседей, игнорируя все неосновные точки.
- Назначить каждую неосновную точку ближайшему кластеру, если кластер является $\varepsilon$-соседним, в противном случае считаем точку шумом.

## Преимущества DBSCAN

- DBSCAN не требует спецификации числа кластеров в данных априори в отличие от метода k-средних.
- DBSCAN может найти кластеры произвольной формы. Он может найти даже кластеры полностью окружённые (но не связанные с) другими кластерами.



©Chire

## Преимущества DBSCAN

- DBSCAN имеет понятие шума и устойчив к выбросам.
- DBSCAN требует лишь двух параметров ($minPts$ и $\varepsilon$) и большей частью нечувствителен к порядку точек в базе данных.

Однако, точки, находящиеся на границе двух различных кластеров могут оказаться в другом кластере, если изменить порядок точек, а назначение кластеров единственно с точностью до изоморфизма.

## Недостатки DBSCAN

- DBSCAN не полностью однозначен — краевые точки, которые могут быть достигнуты из более чем одного кластера, могут принадлежать любому из этих кластеров, что зависит от порядка просмотра точек.

DBSCAN* является вариантом алгоритма, который трактует краевые точки как шум и тем самым достигается полностью однозначный результат.

## Недостатки DBSCAN

- Качество DBSCAN зависит от функции измерения расстояния. Наиболее часто используемой метрикой расстояний является евклидова метрика. В случае кластеризации данных высокой размерности эта метрика может оказаться почти бесполезной, что делает трудным делом нахождение подходящего значения $\varepsilon$.

Этот эффект, однако, присутствует в любом другом алгоритме, основанном на евклидовом расстоянии.

## Недостатки DBSCAN

- DBSCAN не может хорошо разделить на кластеры наборы данных с большой разницей в плотности, поскольку не удается выбрать приемлемую для всех кластеров комбинацию $minPts$ и $\varepsilon$.

## Алгоритм «Распространение близости»

Алгоритм распространения близости (*Affinity Propagation, AP*) – это алгоритм кластеризации, в основе которого лежит идея «передачи сообщений» между точками.

Также, как и DBSCAN, *AP* не требует априорного задания числа кластеров.

Подобно алгоритму $k$-medoids, *AP* находит «экземпляры» - элементы входного набора данных, являющиеся представителями («лидерами») кластеров.

## Алгоритм «Распространение близости»

Пусть $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ - множество объектов в пространстве признаков, и пусть $s$ – функция, вычисляющая сходство между двумя точками, так что $s(i,j) > s(i,k)$ тогда и только тогда, когда $\mathbf{x}_i$ более сходно с $\mathbf{x}_j$, чем с $\mathbf{x}_k$.

Примем $s(i,j) = -\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2$ - отрицательный квадрат эвклидова расстояния.

## Алгоритм «Распространение близости»

Значения функции $s(i, j)$, вычисленные для всех $N$ точек, можно представить в виде матрицы $\mathbf{S}$ размера $N \times N$.

При этом диагональные элементы матрицы $\mathbf{S}$, т.е. $s(i, i)$, будут представлять «предпочтения» элементов $\mathbf{x}_i$, т.е. указывать на то, сколь вероятно, что объект $\mathbf{x}_i$ станет «экземпляром» (лидером кластера).

## Алгоритм «Распространение близости»

Если на этапе инициализации установить одинаковые значения $s(i,i) = p$ для всех $i$, то эта величина будет определять, сколько кластеров получится в итоге.

Меньшее значение $p$ даст в итоге меньшее количество кластеров.

*Обычно* начальное значение $p$ определяют как медиану значений $s(i,j), i \neq j$, вычисленных для всех точек.

## Алгоритм «Распространение близости»

Таким образом, можно записать

$$\mathbf{S} = s + p \times \mathbf{I}_N,$$

где $s \equiv s(i,j) = -\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2$;
$p$ – величина «предпочтения»;
$\mathbf{I}_N$ - единичная матрица порядка $N$.

## Алгоритм «Распространение близости»

В ходе работы алгоритма поочередно выполняются два шага, на которых обновляются матрицы:

- матрица «ответственности» $\mathbf{R}$ со значениями $r(i,k)$, показывающими, насколько хорошо объект $\mathbf{x}_k$ подходит для того, чтобы быть экземпляром для $\mathbf{x}_i$, относительно других кандидатов в экземпляры;
- матрица «доступности» $\mathbf{A}$ со значениями $a(i,k)$, показывающими, насколько «уместно» для $\mathbf{x}_i$ выбрать $\mathbf{x}_k$ в качестве своего экземпляра.

## Алгоритм «Распространение близости»

На этапе инициализации обе матрицы содержат нули. На каждой итерации выполняются следующие действия:

$$r(i,k) \leftarrow s(i,k) - \max_{k' \neq k}\{a(i,k') + s(i,k')\}$$

$$\begin{cases} a(i,k) \leftarrow \min\left\{0, \left[r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\right]\right\}; i \neq k; \\ a(k,k) \leftarrow \sum_{i' \neq k} \max\{0, r(i',k)\}. \end{cases}$$

## Алгоритм «Распространение близости»

Итерации выполняются до тех пор пока

- либо состав кластеров не меняется на протяжении нескольких итераций,
- либо достигнуто некоторое заранее определенное число итераций.

В итоге экземплярами становятся те объекты, для которых

$$r(i,i) + a(i,i) > 0.$$

## Алгоритм «Распространение близости»

Обозначим через $c_i$ номер кластера, к которому в итоге отнесен объект $\mathbf{x}_i$; тогда

$$c_i = \arg\max_k \big(r(i,k) + a(i,k)\big).$$

Т.е. номер кластера для объекта $\mathbf{x}_i$ определяется номером столбца максимального элемента в $i$-ой строке матрицы $\mathbf{R} + \mathbf{A}$.

## Алгоритм «Распространение близости»

Авторы алгоритма указали на то, что использование указанных выше формул может в некоторых случаях привести к численным осцилляциям, т.е. состав кластеров будет меняться с определенным периодом.

Для решения этой проблемы авторы предложили, *во-первых*, на этапе инициализации добавить к элементам матрицы сходства $\mathbf{S}$, небольшую величину случайного шума: $\mathbf{S} = s + p \times \mathbf{I}_N + \boldsymbol{\varepsilon}_{N \times N}$.

## Алгоритм «Распространение близости»

*Во-вторых*, при вычислении матриц $\mathbf{R}$ и $\mathbf{A}$ в теле алгоритма используют экспоненциальное сглаживание с демпфирующим фактором $\lambda$:

$$r^{(j+1)}(i,k) \leftarrow \lambda r^{(j)}(i,k) + (1-\lambda)\left[s(i,k) - \max_{k' \neq k}\{a(i,k') + s(i,k')\}\right]$$

$$\begin{cases} a^{(j+1)}(i,k) \leftarrow \lambda a^{(j)}(i,k) + (1-\lambda)\left[\min\left\{0, \left[r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\right]\right\};\right] \\ a^{(j+1)}(k,k) \leftarrow \lambda a^{(j)}(k,k) + (1-\lambda)\left[\sum_{i' \neq k} \max\{0, r(i',k)\}\right]. \end{cases}$$

## Алгоритм «Распространение близости»

Значение демпфирующего фактора $\lambda$ выбирается из интервала $[0; 1)$.

При этом, авторы рекомендуют первоначально использовать значение $\lambda = 0{,}5$, а в случае, если алгоритм не будет сходиться, увеличить значение $\lambda$ до $0{,}9 - 0{,}95$.

Нужно понимать, что увеличение значения $\lambda$ приведет к росту числа итераций!

## Алгоритм «Распространение близости»

В большом количестве научных работ, посвященных алгоритму *AP*, заявляется о его преимуществах по сравнению с другими алгоритмами кластеризации, в том числе и о меньшей (в целом) ошибке кластеризации.

Однако, признаваемым всеми недостатком алгоритма *AP* является бóльший (в сравнении с другими алгоритмами) объем памяти, необходимый для работы, особенно при кластеризации больших наборов данных ($> 10^5$).

## Алгоритм «Распространение близости»

В настоящее время существует большое число вариаций алгоритма AP, позволяющих повысить скорость работы.

## Валидация кластеров

Под валидацией кластеров понимают проверку их обоснованности.

Различают два типа валидации:

- внутреннюю – по тому, насколько кластеры соответствуют данным,
- и внешнюю – по тому, насколько кластеры соответствуют информации, не учитывавшийся при их построении, но известной специалистам.

## Внешние критерии валидации

Внешние критерии валидации обычно используют информацию об истинном разбиении данных на кластеры и чаще используются для оценки качества результатов различных алгоритмов кластеризации.

Среди таких критериев можно упомянуть:

- отрегулированный индекс Рэнда (Adjusted Rand Index, ARI);
- отрегулированная взаимная информация (Adjusted Mutual Information, AMI).

## Внутренние критерии валидации

Внутренние критерии валидации используют информацию о составе получившихся кластеров и их взаимном расположении.

Наиболее популярными критериями являются:

- индекс Дэвиса – Болдина (Davis-Bouldin Index, DBI);
- индекс Данна (Dunn Index, DI);
- силуэт (Silhouette).

## Отрегулированный индекс Рэнда, ARI

$ARI$ измеряет сходство между двумя различными разбиениями набора данных. Предполагаем, что одно из разбиений – истинное.

Пусть имеется набор $\mathbf{S} = \{s_1, s_2, \dots, s_N\}$ из $N$ элементов и два разбиения $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_r\}$ и $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_p\}$, делящих $\mathbf{S}$ на $r$ и $p$ подмножеств (кластеров), соответственно.

## Отрегулированный индекс Рэнда, ARI

Составим таблицу сопряженности:

| | $\mathbf{Y}_1$ | $\mathbf{Y}_2$ | … | $\mathbf{Y}_p$ | $\Sigma$ |
|---|---|---|---|---|---|
| $\mathbf{X}_1$ | $N_{11}$ | $N_{12}$ | … | $N_{1p}$ | $N_{1*}$ |
| $\mathbf{X_2}$ | $N_{21}$ | $N_{22}$ | … | $N_{2p}$ | $N_{2*}$ |
| … | … | … | … | … | … |
| $\mathbf{X}_r$ | $N_{r1}$ | $N_{r2}$ | … | $N_{rp}$ | $N_{r*}$ |
| $\Sigma$ | $N_{*1}$ | $N_{*2}$ | … | $N_{*p}$ | $N$ |

# Кластеризация

| | $\mathbf{Y}_1$ | $\mathbf{Y}_2$ | … | $\mathbf{Y}_p$ | $\Sigma$ |
|---|---|---|---|---|---|
| $\mathbf{X}_1$ | $N_{11}$ | $N_{12}$ | … | $N_{1p}$ | $N_{1*}$ |
| $\mathbf{X}_2$ | $N_{21}$ | $N_{22}$ | … | $N_{2p}$ | $N_{2*}$ |
| … | … | … | … | … | … |
| $\mathbf{X}_r$ | $N_{r1}$ | $N_{r2}$ | … | $N_{rp}$ | $N_{r*}$ |
| $\Sigma$ | $N_{*1}$ | $N_{*2}$ | … | $N_{*p}$ | $N$ |

Здесь $N_{ij} = \left|\mathbf{X}_i \cap \mathbf{Y}_j\right|$ - число элементов, содержащихся одновременно в кластерах $\mathbf{X}_i$ и $\mathbf{Y}_j$;

$N_{i*}$ - число элементов в кластере $\mathbf{X}_i$;

$N_{*j}$ - число элементов в кластере $\mathbf{Y}_j$.

Вычислим

$$a = \sum_{i=1}^{r}\sum_{j=1}^{p}\binom{N_{ij}}{2};\ \ b = \sum_{i=1}^{r}\binom{N_{i*}}{2};$$

$$c = \sum_{j=1}^{p}\binom{N_{*j}}{2};\ \ d = \frac{2bc}{N(N-1)}.$$

где $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ - биномиальный коэффициент.

Тогда отрегулированный индекс Рэнда равен:

$$ARI(\mathbf{X}, \mathbf{Y}) = \frac{a - d}{0{,}5(b + c) - d}.$$

$$ARI \in [-1; 1].$$

Значение $ARI$ равное 1 соответствует идентичному разбиению, близкие к 0 – случайному разбиению. Отрицательные значения соответствуют "независимым" разбиениям на кластеры.

## Отрегулированная взаимная информация, AMI

$AMI$ очень похожа на $ARI$. Эта мера определяется с использованием функции *энтропии*, интерпретируя разбиения выборки, как дискретные распределения (вероятность отнесения к кластеру равна доле объектов в нём).

Индекс $MI$ определяется как взаимная информация для двух распределений, соответствующих разбиениям выборки на кластеры.

## Отрегулированная взаимная информация, AMI

Для вычисления $AMI$ нам потребуется такая же таблица сопряженности, что и для $ARI$.

Предположим, из набора $\mathbf{S} = \{s_1, s_2, \dots, s_N\}$ наугад выбирается один объект.

## Отрегулированная взаимная информация, AMI

Вероятность того, что он принадлежит кластеру $\mathbf{X}_i$ , определяется как

$$P(i) = \frac{|\mathbf{X}_i|}{N}.$$

Аналогично, вероятность того, что выбранный объект принадлежит кластеру $\mathbf{Y}_j$, равна

$$P'(j) = \frac{|\mathbf{Y}_j|}{N}.$$

## Отрегулированная взаимная информация, AMI

Вычислим значения энтропии, соответствующие разбиениям $\mathbf{X}$ и $\mathbf{Y}$:

$$H(\mathbf{X}) = -\sum_{i=1}^{r} P(i)\log\big(P(i)\big);$$

$$H(\mathbf{Y}) = -\sum_{j=1}^{p} P'(j)\log\big(P'(j)\big).$$

## Отрегулированная взаимная информация, AMI

Коэффициент взаимной информации ($MI$) между двумя разбиениями рассчитывается следующим образом:

$$MI(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{r} \sum_{j=1}^{p} P(i,j) \log\left(\frac{P(i,j)}{P(i)P'(j)}\right);$$

где $P(i,j)$ - вероятность того, что объект принадлежит одновременно кластерам $\mathbf{X}_i$ и $\mathbf{Y}_j$: $P(i) = \frac{|\mathbf{X}_i \cap \mathbf{Y}_j|}{N}$.

## Отрегулированная взаимная информация, AMI

Аналогично $ARI$, коэффициент $MI$ нужно отрегулировать:

$$AMI = \frac{MI - E[MI]}{\max\big(H(\mathbf{X}), H(\mathbf{Y})\big) - E[MI]},$$

где $E[MI]$ - ожидаемое значение взаимной информации.

$$0 \leq AMI \leq 1$$

# Отрегулированная взаимная информация, AMI

| | $\mathbf{Y}_1$ | $\mathbf{Y}_2$ | … | $\mathbf{Y}_p$ | $\Sigma$ |
|---|---|---|---|---|---|
| $\mathbf{X}_1$ | $N_{11}$ | $N_{12}$ | … | $N_{1p}$ | $a_1$ |
| $\mathbf{X_2}$ | $N_{21}$ | $N_{22}$ | … | $N_{2p}$ | $a_2$ |
| … | … | … | … | … | … |
| $\mathbf{X}_r$ | $N_{r1}$ | $N_{r2}$ | … | $N_{rp}$ | $a_r$ |
| $\Sigma$ | $b_1$ | $b_2$ | … | $b_p$ | $N$ |

## Отрегулированная взаимная информация, AMI

$$E[MI] = \sum_{i=1}^{r}\sum_{j=1}^{p}\sum_{N_{ij}=(a_i+b_j-N)^+}^{\min(a_i,b_j)} \frac{N_{ij}}{N}\log\left(\frac{N\cdot N_{ij}}{a_i b_j}\right)\times$$

$$\times\frac{a_i!\,b_j!\,(N-a_i)!\,(N-b_j)!}{N!\,N_{ij}!\,(a_i-N_{ij})!\,(b_j-N_{ij})!\,(N-a_i-b_j+N_{ij})!},$$

где $(a_i+b_j-N)^+ = \max(1, a_i+b_j-N)$.

## Силуэт

Значение силуэта является мерой того, насколько похож объект на другие объекты из своего кластера в сравнении с объектами из других кластеров.

Значение силуэта находится в диапазоне от -1 до 1: значения, близкие к 1 указывают на то, что объект хорошо «вписывается» в свой кластер и плохо – в другие.

## Силуэт

Если большинство объектов имеют высокое значение силуэта, то кластеризация выполнена хорошо; если у многих объектов низкое или отрицательное значение силуэта, это может указывать на слишком большое или слишком малое число кластеров.

## Силуэт

Предположим, данные были разделены на кластеры.
Пусть для объекта $i \in C_i$ из кластера $C_i$

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

будет означать среднее расстояние между $i$ и другими объектами из того же кластера; $d(i, j)$ - мера расстояния.

## Силуэт

Далее, пусть для объекта $i \in C_i$ из кластера $C_i$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

будет означать наименьшее среднее расстояние между $i$ и объектами во всех других кластерах. Кластер, для которого $b(i)$ является наименьшим, называется «соседним» кластером.

## Силуэт

Теперь мы можем определить силуэт для объекта $i$:

$$s(i) = \begin{cases} \dfrac{b(i) - a(i)}{\max\{a(i), b(i)\}}, & if\ |C_i| > 1; \\ 0, & if\ |C_i| = 1. \end{cases}$$

Отметим, что в случае $|C_i| = 1$, $a(i) = 0$.

## Силуэт

Среднее значение $s(i)$ для всех объектов из набора данных показывает, насколько хорошо была выполнена кластеризация:

$$\tilde{s} = \frac{1}{N}\sum_{i=1}^{N} s(i)\,.$$

Если имеются несколько разбиений с разным числом кластеров (например, с использованием $k$-means), максимальное значение $\tilde{s}_k$ позволит определить оптимальное число кластеров.

## Индекс Дэвиса-Болдина, DBI

$DBI$, возможно, одна из самых используемых мер оценки качества кластеризации.

Она вычисляет компактность как расстояние от объектов кластера до их центроидов, а отделимость - как расстояние между центроидами.

## Индекс Дэвиса-Болдина, DBI

Предположим, имеется разбиение данных на $K$ кластеров, и в нем каждый кластер $C_i$ имеет размер $|C_i| = T_i$ и центроид $A_i$. Также, пусть объекты $X_j$ принадлежат кластеру $C_i$.

Мерой компактности кластера $C_i$ назовем величину

$$S_i = \left( \frac{1}{T_i} \sum_{j=1}^{T_i} |X_j - A_i|^p \right)^{1/p}$$

обычно $p = 2$

## Индекс Дэвиса-Болдина, DBI

Мерой отделимости кластеров $C_i$ и $C_j$ назовем величину

$$M_{ij} = \|A_i - A_j\|_p,$$

обычно $p = 2$

т.е. расстояние между центроидами.

## Индекс Дэвиса-Болдина, DBI

Введем величину $R_{ij} = \frac{S_i + S_j}{M_{ij}}$ и найдем

$$D_i = \max_{j \neq i} R_{ij} .$$

Тогда

$$DBI = \frac{1}{K} \sum_{i=1}^{K} D_i .$$

## Индекс Дэвиса-Болдина, DBI

$DBI$ принимает значения больше 0 и, по сути, является средним отношением внутрикластерных разбросов к расстояниям между кластерами.

Наилучшее разбиение на кластеры минимизирует $DBI$.