Математические Методы Распознавания образов, весна, ВМК МГУ
Семинар №1

Автор: Афанасьев Глеб

1 Введение

В прошлом семестре было упомянуто, что классические методы машинного обучения можно в общем разделить на три категории:

- обучение по размеченным данным (обучение с учителем, supervised Learning, SL). Обучающая выборка здесь состоит из пар (x, y), где x – описание объекта, y – его метка. Необходимо обучить модель y=f(x), которая по описанию x предсказывает метки.
- обучение с частично размеченными данными (Semi-Supervised Learning / SSL). Обучающая выборка в этом методе состоит из данных с метками и без меток (последних, как правило, существенно больше). Необходимо также обучить модель y=f(x), но здесь может помочь информация о том, как объекты располагаются в пространстве описаний.
- обучение по неразмеченным данным (обучение без учителя, unsupervised Learning, UL) – даны только объекты (без меток), необходимо эффективно описать, как они располагаются в пространстве описаний. Например, типичные задачи обучения по неразмеченным данным – кластеризация, понижение размерности, детектирование аномалий, оценка плотности и т.п

В первой части курса мы рассмотрим одну из задач обучения без учителя. А именно, кластеризацию.

§1.1 Постановка задачи кластеризации

Наиболее интуитивно похожей на кластеризацию задачей машинного обучения, которая была изучена ранее, является задача классификации. В ней целью было разделить предсказываемую выборку на классы на основании обучающей выборки.

Задача кластеризации также состоит в разделении данных на группы в пространстве признаков. Однако, в случае кластеризации, мы не имеем обучающую выборку, следовательно, у нас нет никакой информации об объективной правильности решения. Более того, мы не всегда знаем даже количество групп, на которые мы хотим разделить данные. Единственным ориентиром является требование чтобы точки, находящиеся в одной группе, были максимально "похожи в то время как точки в разных группах должны были максимально "непохожи".

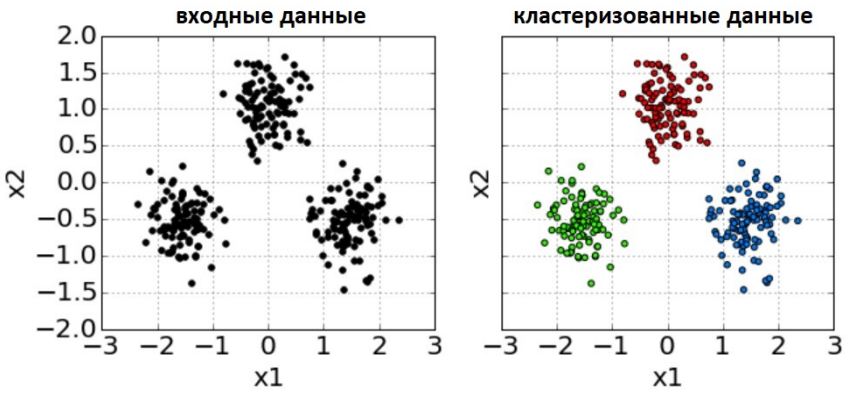


Рис. 1. Пример кластеризации

§1.2 Проблемы постановки задачи кластеризации

Главной проблемой постановки задачи является то что единой формализации понятия "похожести"объектов нет и быть не может. Его формализация зависит от факторов, специфичных для конкретной задачи, таких как:

- знаем ли мы требуемое количество кластеров, или нет?
- Есть ли у нас понимание "формы"кластеров (все ли они выпуклые, или могут быть произвольные)?
- Есть ли у нас ограничение по вычислительной сложности?
- И многие другие подобные факторы

Далее мы рассмотрим различные существующие методы кластеризации и некоторые решения обозначенных выше проблем.

2 Representative-based методы

Representative-based методы основаны на конструировании наиболее обобщенных представителей для каждого из кластеров. Кластеризация осуществляется путем отнесения объекта к кластеру, обобщенный представитель которого оказался к нему самым близким по метрике, определяемой конкретной задачей. Количество кластеров - гиперпараметр, задаваемый инженером.

Более формально, representative-based методы решают оптимизационную задачу следующего вида:

$$L(y_1, y_2, ..., y_N, \mu_1, ..., \mu_K) = \sum_{i=1}^{N} \rho(x_i, \mu_{y_i}) \to min$$

Где x_i - объект выборки, y_i - ассоциированный с ним кластер, μ_k - обобщенный представитель кластера k, $\rho(x, y)$ - метрика в признаковом пространстве.

§2.1 Метод K-means

Одним из самых известных Representative-based методов является K-means. В нем мы выбираем метрику ρ как евклидово расстояние между точками. Таким образом, задача принимает следующий вид:

$$\sum_{i=1}^{N} ||x_i - \mu_{y_i}||^2 \to min$$

Так как решать подобную задачу по всем параметрам сложно, обычно используют итерационный метод:

инициализируем $\mu_1, ..., \mu_K$

пока not converged

для i=1, 2, ..., N

$y_i = argmin_{j \in 1,2,...,K} ||\mu_j - x_i||^2$

для i=1, 2, ..., K

$\mu_i = \frac{1}{\sum_{j=1}^{N} I[y_j = \mu_i]} \sum_{k=1}^{N} I[y_j = \mu_i] x_k$

=0

Условиями сходимости могут выступать:

- достижение максимального количества итераций
- факт того что обобщенные представители кластеров перестали двигаться от итерации к итерации: $\sum_{i=1}^{K} ||\mu_i' - \mu_i||^2 = 0$
- факт того что обобщенные представители кластеров перестали двигаться на расстояние, большее чем заданное от итерации к итерации: $\sum_{i=1}^{K} ||\mu_i' - \mu_i||^2 \leqslant \epsilon$

Очевидно, данный алгоритм сильно зависит от начальной инициализации кластеров. Кроме того, из вида алгоритма следует что он предполагает что кластеры выпуклые.

k-means обязательно пытается отдать каждому кластеру какие-то объекты даже если по факту их меньше. Это значит, в том числе, что этот метод хорош для детектирования аномалий, так как если мы имеем дело с выбросом, то этот выброс будет единственным объектом кластера, обобщенным представителем которого будет он сам.

§2.2 Эллиптический метод кластеризации

Одной из характерных особенностей K-means, является то что метод в идеале хочет создать требуемое количество кластеров с одинаковой плотностью объектов внутри него. Следовательно, если мы имеем дело с кластерами разной плотности, K-means не всегда бует выдавать адекватное разбиение.

Одним из очевидных решений является замена евклидова расстояния на метрику, учитывающую соотношение данных в одном кластере, и настраивать ее параллельно с перенастройкой обобщенных представителей. Такой метрикой является расстояние Махаланобиса:

$$\rho(x, \mu_k)^2 = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$$

где Σ_k - матрица ковариации элементов кластера k, которая перенастраивается на каждом шаге перед пересчетом обобщенного представителя кластера k.
Таким образом, мы получаем возможность получать кластеры различного размера и плотности. Однако, они все еще обязаны быть выпуклыми.

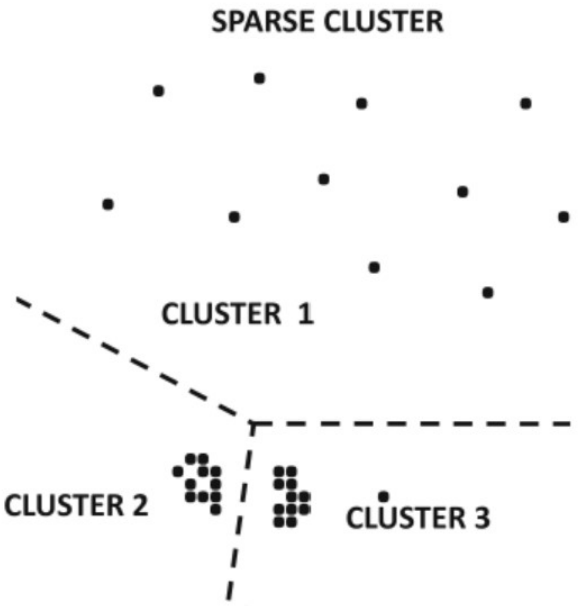


Рис. 2. Пример эллиптической кластеризации

§2.3 Ядерный K-mean

Вспомним теорию прошлого семестра. Конкретно, метод svm. В нем предлагалось с помощью ядерного трюка учитывать нелинейные зависимости данных линейными методами. Попробуем провернуть аналогичный трюк для задачи кластеризации. В kernel svm вся теория строилась на том факте, что для алгоритма требовалось лишь знание скалярного произведения между объектами, в то время как сами объекты не требовались. Здесь мы имеем аналогичную ситуацию. Давайте запишем это математически.

$$\rho(\mu_k, x) = ||\mu_k - x||^2 = \langle \mu_k - x, \mu_k - x \rangle = \langle \frac{1}{|C_k|} \sum_{i \in C_k} x_i - x, \frac{1}{|C_k|} \sum_{i \in C_k} x_i - x \rangle =$$

$$= \langle x, x \rangle - 2 * \frac{1}{|C_k|} \sum_{i \in C_k} \langle x_i, x \rangle + \frac{1}{|C_k|^2} \sum_{i,j \in C_k} \langle x_i, x_j \rangle$$

Как видно, для отнесения объекта к кластеру нам нужно лишь скалярное произведение. Следовательно, мы можем заменить его на ядро:

$$\rho(\mu_k, x) = K(x, x) - 2 * \frac{1}{|C_k|} \sum_{i \in C_k} K(x_i, x) + \frac{1}{|C_k|^2} \sum_{i,j \in C_k} K(x_i, x_j)$$

Важно заметить что в отличие от описанного выше алгоритма, когда мы поэтапно на каждом шаге сначала пересчитывали кластеры для каждого объекта, а постом пересчитывали обобщенных представителей каждого кластера, здесь мы совмещаем эти два шага в один. Пересчет обобщенных представителей уже зашит в расчет расстояния до них, так как на каждом этапе, согласно формуле выше, мы считаем расстояние уже до пересчитанного представителя.
Алгоритм выглядит следующим образом:
инициализируем $\mu_1, ..., \mu_K$

пока not converged
  для i=1, 2, ..., N
    $y_i = argmin_{j \in 1,2,...,K} \rho(\mu_j - x_i)^2$
  =0

Таким образом, данный алгоритм может выделять невыпуклые кластеры, однако определение количества кластеров все еще остается задачей инженера.

3 Иерархическая кластеризация

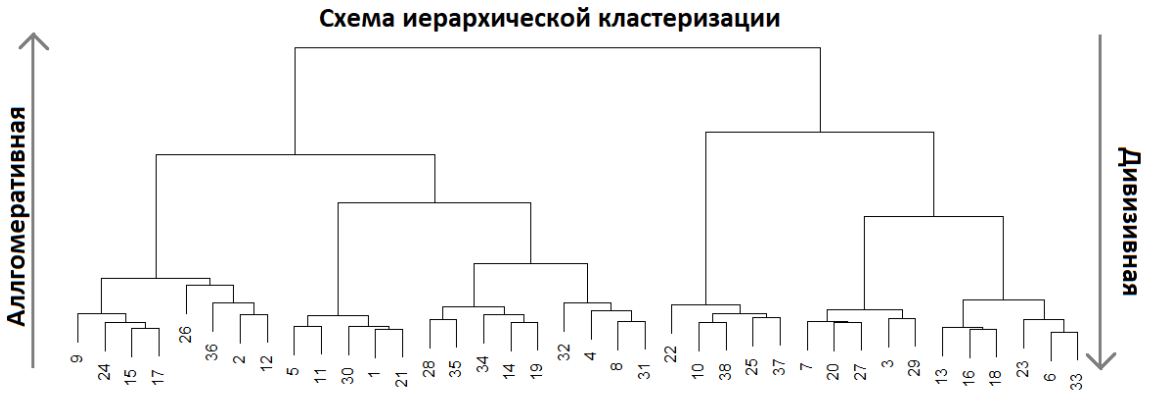
Рассмотрим иной подход к кластеризации, не требующий заранее задавать количество кластеров.
Алгоритмы, описанные выше, являлись так называемыми алгоритмами "плоской"кластеризации. В таких алгоритмах кластеры формировались как некоторое подмножество объектов без введенной на подмножестве структуры элементов. То есть все кластеры обладали одинаковыми свойствами.
В методах иерархической кластеризации предлагается вести структуру элементов на кластерах и выстраивать их таким образом, чтобы кластеры выстраивались в соответствии с некоторой иерархией. Таким образом, после отработки алгоритма у нас получается один кластер, содержащий все объекты, который делится на несколько подкластеров, которые, в свою очередь, также делятся на подкластеры и так далее. Самые маленькие подкластеры, очевидно, будут содержать по одному элементу и их количество будет совпадать с количеством элементов.
Существует два варианта иерархической кластеризации:

- агломеративная, в которой алгоритм на каждой итерации объединяет два меньших кластера в один;
- дивизивная, в которой алгоритм на каждой итерации разбивает один кластер на два более мелких;

§3.1 Агломеративная кластеризация

Алгоритм агломеративной кластеризации:

1. Инициализируем наше множество кластеров, каждая точка считается своим кластером. То есть для выборки размера N у нас на первой итерации будет N кластеров. Также входным параметром алгоритму подается метрика расстояния между двумя кластерами.



2. На каждой итерации мы объединяем два кластера в один. Объединяющиеся кластера выбираются в соответствии с наименьшим значением выбранной метрики. То есть в соответствии с выбранным нами расстоянием эти два кластера будут наиболее похожи и поэтому объединяются.

3. Предыдущий шаг повторяется вплоть до объединения всех точек один кластер.

В результате в данном подходе мы можем выбрать любое количество кластеров после завершения процедуры, просто остановив на нужном нам шаге. К тому же данный алгоритм гораздо менее чувствителен к выбору метрики между точками, тогда как другие алгоритмы сильно зависят от этого.
Рассмотрим существующие метрики метрика расстояния между кластерами:

- Расстояние между ближайшими элементами: $\rho(A, B) = min_{a \in A, b \in B} \rho(a, b)$

  1. Не зависит от формы кластера
  2. Чувствителен к выбросам
  3. $\rho(A \bigcup B, C) = min(\rho(A, C), \rho(B, C))$

- Расстояние между самыми дальними элементами: $\rho(A, B) = max_{a \in A, b \in B} \rho(a, b)$

  1. Не зависит от формы кластера
  2. Чувствителен к выбросам
  3. Кластеры получаются очень компактными
  4. $\rho(A \bigcup B, C) = max(\rho(A, C), \rho(B, C))$

- Расстояние между обобщенными представителями: $\rho(A, B) = \rho(\mu_A, \mu_B)$
- Усредненное расстояние между элементами: $\rho(A, B) = mean_{a \in A, b \in B} \rho(a, b)$

§3.2 Дивизивная кластеризация

Этот метод кластеризации менее популярен. Его алгоритм выглядит следующим образом:

1. Инициализируем множество кластеров одним кластером, содержащим все точки. Также инициализируем базовый метод плоской кластеризации и базовое число кластеров.

2. На каждой итерации кластеризуем внутри одного кластера. То есть в соответствии с выбранным алгоритмом кластеризации, делим кластер на небольшое число подкластеров.

3. Предыдущий шаг повторяется вплоть до разделения всех точек на кластеры, содержащие только по одной точке.

4 Плотностная кластеризация

Рассмотрим алгоритмы, основанные на анализе плотности элементов.

§4.1 DBScan

Первый плотностный алгоритм - это DBScan. Если дан набор объектов в некотором пространстве, алгоритм группирует вместе объекты, расположенные в области высокой плотности, и помечает как выбросы объекты, находящиеся в областях с малой плотностью (ближайшие соседи которых лежат далеко). Алгоритм имеет два основных гиперпараметра:

- eps — радиус рассматриваемой окрестности
- min_samples — число соседей в окрестности

Все точки делятся на основные точки, достижимые по плотности точки и выбросы следующим образом:

- Точка p является основной точкой, если по меньшей мере min_samples точек находятся от нее на расстоянии, не превосходящем eps.
- Точка q прямо достижима из основной точки p, если точка находится на расстоянии, не большем eps от точки p.
- Точка q достижима из основной точки p, если имеется путь $p_1, p_2, ..., p_n$ где $p_1 = p, p_n = q$ и каждая точка p_i достижима прямо из p_{i-1}. Все точки на пути должны быть основными, за исключением q.
- Все точки, не достижимые из основных точек, считаются выбросами.

Если p является основной точкой, то она формирует кластер вместе со всеми точками (основными или неосновными), достижимыми из этой точки. Каждый кластер содержит по меньшей мере одну основную точку. Неосновные точки

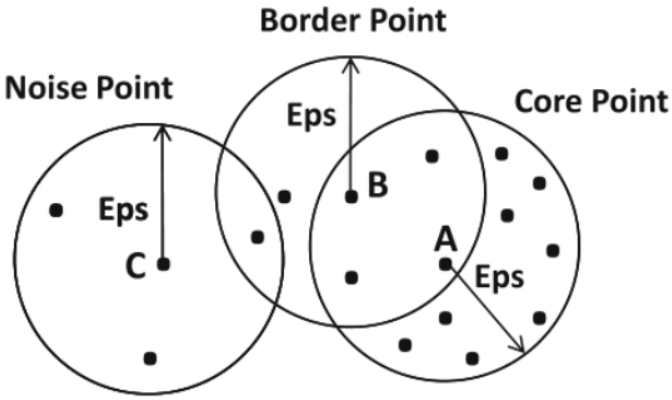


Рис. 3. DBScan

могут быть частью кластера, но они формируют его «край», поскольку не могут быть использованы для достижения других точек.
Данный алгоритм не требует выпуклости кластеров. Кроме того, он устойчив к выбросам и сам определяет количество кластеров. Однако, он плохо работает для кластеров различной плотности.

§4.2 Сеточная кластеризация

Сеточная кластеризация основывается на идее разделения пространства на небольшие области и анализе объектов, попавших в них.
Алгоритм выглядит следующим образом:

1. Инициализируем гиперпараметры p, k и r;
2. Делим каждую ось на p частей, таким образом получаем p^D D-мерых кубов;
3. Куб будем рассматривать полным, если в нем оказалось более k точек;
4. Формируем граф следующим образом: вершины - полные кубы. Между вершинами есть ребро, если два соответствующих куба имеют r или больше общих границ;
5. Компоненты связности графа образуют кластеры;

Как и DBScan данный метод не требует выпуклости кластеров, устойчив к выбросам и сам определяет количество кластеров. Однако, все так же плохо работает для кластеров различной плотности.

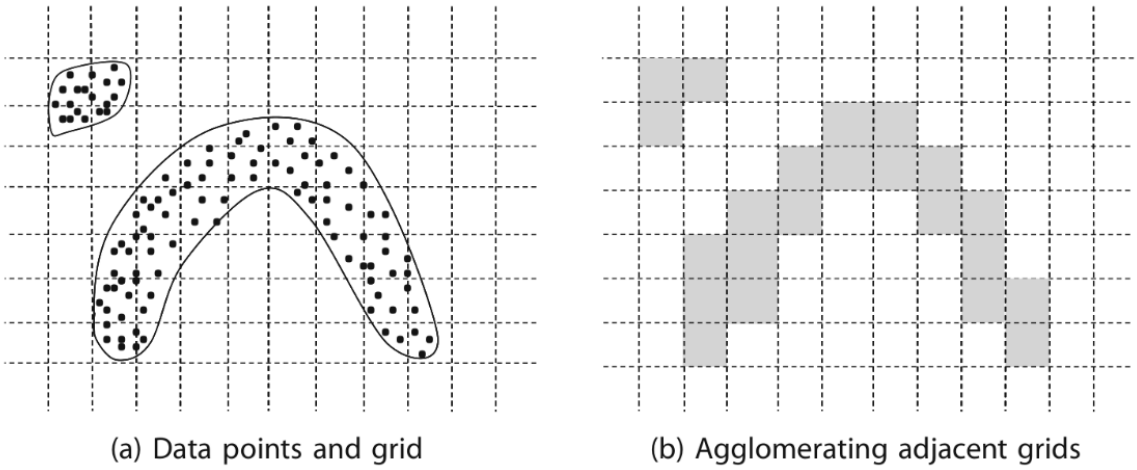


Рис. 4. Сеточная кластеризация

Список литературы

[1] *Лекция по кластеризаци от Виктора Китова* https://yadi.sk/i/5Uu3pOIO3W7KtE

[2] *Семинар по кластеризации от Евгения Соколова* https://github.com/esokolov/ml-course-hse/blob/master/2021-fall/seminars/sem11_clustering.ipynb