Сети Кохонена.

Ссылка на видеокурс от МИАН

В этой и следующей статье будет рассмотрено несколько методов автоматической классификации данных. Двое из рассмотренных методов связаны с фамилией финского ученого **Теуво Кохонена** (1934-2021), получившего известность благодаря изобретению своих карт, позволяющих визуализировать кластеризованные данные.

Постановка задачи кластеризации

Перед рассмотрением самого метода определим, в чем заключается задача классификация каких-либо объектов.

Имеем множество X, которое является непустым. Это множество можно представить как

$$X = \bigcup_{k=1}^{K} X_k,$$

где $X_i \cap X_j = \varnothing$, если $i \neq j, i = 1, \ldots, K, j = 1, \ldots, K$. Иными словами, множество X можно разбить на K попарно непересекающихся подмножеств и пронумеровать их. Говорят, что K — количество кластеров (классов) множества X.

Задача классификации состоит в том, что требуется построить функцию

$$F : X \to \{1, 2, \ldots, K\}.$$

Для любого $x \in X$ число $F(x)$ — номер класса, к которому относится этот элемент x. Заметим, что в некоторых задачах число классов может быть известно, а в других — неизвестно.

В машинном обучении существует два похода к построению функции F.

- **Обучение с учителем**.

  Имеем множества пар

  $$\{(x^m, k^m)\}_{m=1}^{M},$$

  где $x^m \in X, k^m \in \{1, 2, \ldots, K\}$. То есть имеется конечный набор мощности $|M|$ из элементов множества X, для которых известно, к какому классу они принадлежат. Иными словами, должно выполняться условие $F(x^m) = k^m$, $m = 1, \ldots, M$. Этот набор называется **обучающей выборкой**. Обычно она собирается непосредственно человеком. Модель настраивается на этой выборке, и по аналогии производится классификация элементов $x \in X$, для которых $F(x) =?$.

  По своей формулировке это в точности **задача интерполяции**. Но пользоваться стандартными методами интерполяции для ее решения мы, естественно, не можем: в нетривиальных задачах распознавания образов обучающая выборка даже близко не характеризует функцию F; эта функция очень сложна по своей природе: она не везде определенная, негладкая, наглядно представить ее почти нельзя. Поэтому строить ее нужно из других соображений.

  Классическая нейронная сеть относится к обучению с учителем.

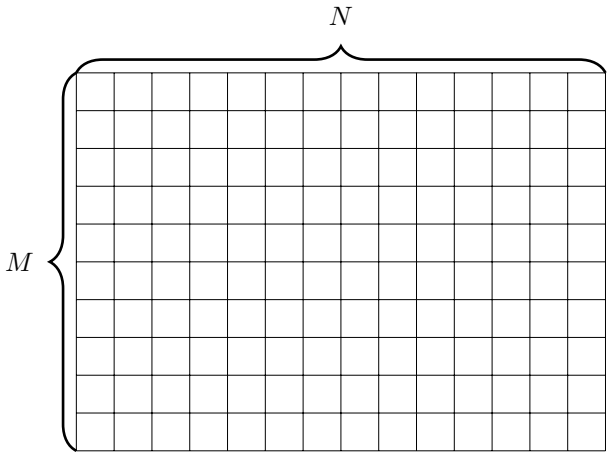
- **Обучение без учителя (автоматическая классификация)**.

  В обучении без учителя модели предоставляется большое количество многомерной информации, которую она должна обработать. Разделение объектов обучающей выборки на классы не задаётся, и требуется классифицировать объекты только **на основе их сходства друг с другом**. В этом случае обучающая выборка может совпадать со всем множеством X. Также очевидно, что в этом случае не ставится задача определения, какой у объекта номер класса — класс может быть любой, но он определен, можно выделить его характерные черты, дать ему трактовку и название (или номер).

  Нейронная сеть Кохонена, **самоорганизующиеся карты Кохонена** и **EM-алгоритм** относятся к обучению без учителя.

Задачи обучения без учителя

Объекты можно описать в виде многомерного вектора, компоненты которого отвечают за ту или иную характеристику. Характеристика — некоторое действительное число, которое может иметь единицу измерения. Удобно представить множество объектов с их характеристиками в виде таблицы.



Имеем M N-мерных векторов $x^m = (x_1^m, x_2^m, \ldots, x_N^m)$, $m = 1, 2, \ldots, M$, $X \subset \mathbb{R}^N$. M — количество объектов в обучающей выборке, N — количество характеристик каждого объекта. В каждом столбце матрицы описаны значения одной характеристики каждого объекта, а в каждой строке — все характеристики конкретного объекта. Это представление можно считать базой данных.

В ячейках данные могут и отсутствовать, но мы будем считать, что таблица заполнена полностью.

Задача заключается в том, что мы ищем близкие по параметрам объекты и объединяем их в один класс. Можно выделить 4 основных подзадачи, которые требуется решить:

1. Определить, сколько классов объективно есть в представленной выборке. Может быть представленные объекты никак нельзя кластеризовать. А может быть объекты настолько разнородны, что выделится M классов. Этот вопрос нужно выяснить.
2. Построить на обучающей выборке функцию $F(x^m) \in \{1, 2, \ldots, K\}$, $m = 1, \ldots, M$.
3. Продлить, если возможно, ее для любого $x^m \in X$, $m = M+1, M+2, \ldots$
4. Выделить для каждого класса **характерные значения** $\widetilde{x^k} \in \mathbb{R}^N$, $k = 1, \ldots, K$. По сути, $\widetilde{x^k}$ — это некий средний представитель каждого из кластеров.

Метод классификации с помощью сети Кохонена отлично справляется с данной задачей.

Алгоритм обучения нейронной сети Кохонена

Сеть Кохонена представляется в виде набора **весовых коэффициентов**

$$w^k = (w_1^k, w_2^k, \ldots, w_N^k), \quad k = 1, 2, \ldots, K.$$

Размерность этих векторов совпадает с размерностью исходных данных.

Рассмотрим сам алгоритм.

1. Поначалу удобно обезразмерить и **нормализовать** исходные данные. То есть привести их к значениям из отрезка $[0, 1]$. Но каким образом произвести нормализацию?

Ясно, что в строке значения никак не связаны друг с другом и могут иметь разные единицы измерения. Поэтому нормализацию проводим **по столбцам**.

Вычисляем

$$Max_n = \max_m x_n^m, \quad Min_n = \min_m x_n^m,$$
$$a_n = \frac{1}{Max_n - Min_n}, \quad b_n = \frac{-Min_n}{Max_n - Min_n},$$
$$n = 1, 2, \ldots, N$$

и нормируем

$$x_n^m = a_n x_n^m + b_n, \quad n = 1, 2, \ldots, N, \quad m = 1, 2, \ldots, M.$$

То есть проводим линейное преобразование исходных векторов. Если подставить в выражение Min_n, то получим 0, а если Max_n — 1. Каждый столбец будет содержать 0, 1 и значения из интервала $(0, 1)$.

Может оказаться так, что $Min_n = Max_n$, это значит, что все значения в столбце одинаковы. Этот признак следует исключить из рассмотрения, так как он не несет какой-либо информативности.

Вычисленные коэффициенты следует сохранить, для того, чтобы можно было денормировать измененные значения.

2. Пусть выбрано некоторое количество $K > 1$ для разбиения. Это будет параметр метода.

   Строим веса нейронной сети $\{w_n^k\}$ случайным образом по равномерному закону распределения

   $$w_n^k \sim R(0.1, 0.3).$$

   Равномерное распределение характеризуется тем, что любое значение из отрезка может выпасть с равной вероятностью. Границы отрезка распределения $[0.1, 0.3]$ выбраны из эмпирических соображений.

   В дальнейшем веса будут двигаться так, что займут характерные значения каждого класса.

3. Еще один параметр — **скорость обучения** (англ. learning rate)

   $$\lambda = 0.3.$$

4. Повторить L раз ($L = 10$):

   (a) Для каждого x^m находим **ближайший** w^{k^*} относительно нормы

   $$\|x^m - w^{k^*}\| \underset{\mathbb{R}^N}{\to} \min_{k=1,2,\ldots,K}.$$

   Нужно перебрать все весовые коэффициенты w^k и найти те, что близки по некоторой норме в $\mathbb{R}^N$.

   (b) Сдвигаем найденный w^{k^*} к x^m по формуле

   $$w^{k^*} = w^{k^*} + \lambda(x^m - w^{k^*}).$$

   Все операции записаны в векторной форме.

5. Уменьшаем длину сдвига весовых коэффициентов

   $$\lambda = \lambda - \Delta\lambda,$$

   где $\Delta\lambda = 0.05$.

6. Если $\lambda = 0$, то завершаем обучение, иначе переходим к шагу 4.

На выходе получим набор весов $\{w^1, w^2, \ldots, w^K\}$ — характерные значения. Они показывают средние значения компонент для каждого класса.

Для интересующего нас вектора x следует найти весовой вектор w^k который наиболее близок к x (прежде всего x необходимо нормализовать). Это и будет класс, к которому относится x.

Для получения натуральных значений весов следует провести процедуру **денормировки**

$$w_n^k = \frac{w_n^k - b_n}{a_n}.$$

Объективное количество кластеров исходного множества

При классификации с помощью сети Кохонена необходимо задать количество кластеров, на которые необходимо разбить исходное множество. При этом, по-видимому, существует объективное количество кластеров. Если это количество не известно, то можно использовать следующую процедуру

1. Задать начальное количество классов $K = 2$.
2. Использовать классификацию на основе сети Кохонена заданное (3-5) число раз. Если при разных запусках получаем различные классы, то конец процедуры — наше множество содержит $K - 1$ класс. Если же получаем одинаковые классы, то переход к следующему шагу.
3. Увеличить количество классов:
$$K = K + 1$$
и перейти к шагу 2.

Кроме этого можно использовать следующий подход:

1. Разбиваем множество на два класса.
2. Если какое-либо из подмножеств содержит большое количество элементов, то работаем с ним и переходим к пункту 1.

Получаем иерархическую кластеризацию.

Самоорганизующиеся карты Кохонена

При анализе данных и автоматической (без учителя) классификации многомерных данных большую популярность имеют самоорганизующиеся карты Кохонена (англ. self-organizing map (SOM)). Они могут быть полезны для кластеризации и представления многомерных данных.

Самоорганизующиеся карты позволяют проецировать многомерные данные на плоскость

$$\mathbb{R}^N \to \mathbb{R}^2,$$

при этом группируя их.

Главной особенностью карты является то, что ее нейроны имеют определенное геометрическое расположение. Поэтому можно задать некоторое расстояние между нейронами, которое повлияет на связи между ними.

Как и в прошлом методе мы имеем данные, которые представляют собой N-мерные вектора

$$x = (x_1, x_2, \ldots, x_N).$$

Построим карту, состоящую из M нейронов, которые расположены на плоской квадратной сетке. Эти нейроны мы занумеруем от 1 до M и определим геометрическое расстояние между их центрами

$$\rho(m', m''), \quad m', m'' \in \{1, 2, \ldots, M\}.$$

С каждым нейроном связан N-мерный вектор весов, размерность которого совпадает с размерностью анализируемых данных

$$w^m = (w_1^m, w_2^m, \ldots, w_N^m), \quad m = 1, 2, \ldots, M.$$

Приведем алгоритм для построения самоорганизующихся карт Кохонена.

1. Проводим нормировку обучающей выборки точно так, как и в предыдущем методе.

2. Инициализируем вектора w^m случайными числами, распределенными по равномерному закону на отрезке $[0, 1]$. Из-за этого при каждом запуске программы будут строиться различные виды карты, тем не менее ее информативность от этого меняться не будет.

3. Параметр времени положим $t = 1$.

4. Выбираем произвольный вектор x из исходных данных.

5. Находим нейрон, веса которого наиболее близки к выбранному x в метрике N-мерного пространства (нейрон-победитель, англ. Best Matching Unit (BMU)). В качестве меры близости удобно использовать Евклидову норму. Пусть нейрон-победитель имеет номер m^*.

6. Корректируем веса нейронов по следующей формуле

$$w_n^m = w_n^m + \eta(t)h(t, \rho(m, m^*))(x_n - w_n^m), \quad n = 1, \ldots, N, \quad m = 1, \ldots, M,$$

   где

$$\eta(t) = \eta_0 e^{-at}, \quad \eta_0 > 0, \quad a > 0,$$
$$h(t, \rho) = e^{-\frac{\rho^2}{2\sigma(t)}},$$
$$\sigma(t) = \sigma_0 e^{-bt}, \quad \sigma_0 > 0, \quad b > 0.$$

   Здесь $0 < \eta(t) < 1$ — обучающий сомножитель, который с ростом времени t будет плавно затухать (уменьшаться). Гауссовская функция $h(t, \rho)$ — функция плотности обучения. Ясно, что с ростом $\rho(m, m^*)$ (расстояния между нейронами на плоскости!) эта функция быстро затухает. Для нейрона-победителя ρ примет значение 0, и значение $h(t, \rho) = 1$. Также значение функции плотности обучения близко к 1 для близких к нейрону m^* нейронов, а для далеких — близко к 0. То есть помимо нейрона-победителя мы обучаем и ближайшие к нему нейроны. Также здесь функция соседства $0 < \sigma(t) < 1$ — это аналог дисперсии, затухающий с ростом t . В качестве η, h, σ могут быть подобраны и другие функции, главное, чтобы они имели похожие свойства с ростом t и ρ.

7. t = t + 1

8. Перейти к шагу 4.

Можно задать максимальное время $t = t_{max}$, в случае превышения которого обучение будет остановлено.

После построения по указанному алгоритму самоорганизующейся карты Кохонена ее необходимо градуировать. Для этого выбираем эталонные значения входных данных (необязательно из тех, которые использовались в обучении) и отмечаем на карте нейрон, веса которого наиболее близки к этим эталонным данным.

После градуировки для анализа нового вектора необходимо найти наиболее близкий к нему нейрон и отметить его на карте. По его расположению и расположению эталонных значений можно оценить этот вектор.

Пример. Классификация предприятий

В наборе данных примера каждый вектор характеризует российское предприятие.

Для характеристики предприятия используется 5 критериев:

1. Активы — денежная оценка всех ресурсов, которые компания использует в бизнесе.

2. Капитал — те средства, что остаются от активов компании после вычета всех обязательств.

3. Операционная прибыль — часть всей выручки компании за отчетный период за вычетом расходов на ведение бизнеса.

4. Чистая прибыль — часть выручки за вычетом всех расходов: на ведение бизнеса, проценты и налоги.

5. Число сотрудников.

Из этой информации можно выделить 3 группы предприятий: гиганты, крупные и средние.

В обучающей выборке представлено 10 компаний.

При $K = 2$ сеть Кохонена смогла сразу определить 2 предприятия-гиганта, остальные отнесла в отдельную группу. При $K = 3$ из второй группы выделено одно крупное предприятие. Остальные 7 отнесены в группу средних. При $K = 4$ появляется пустой класс, поэтому данные можно разбить только на 3 класса.

В группу средних попало одно крупное предприятие, в котором работает очень мало сотрудников — скорее всего сеть посчитала, что этот показатель относит предприятие к числу средняков. Запустим сеть только на тех предприятиях, которые попали в число средних. При $K = 2$ это крупное предприятие отделяется от остальных.

Итого, в нашей выборке 2 предприятия-гиганта, 2 крупных предприятия и 6 средних. Нужно понимать, что такое разбиение основано только на данных, которые доступны сети. Для более объективной (совпадающей с реальностью) кластеризации потребуется большее количество критериев, большее количество информации.

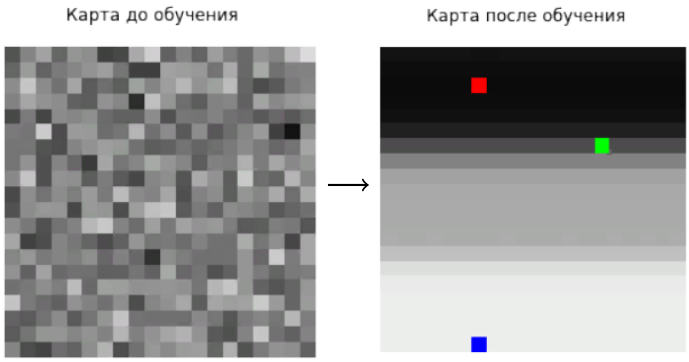
Теперь мы знаем, какие предприятия можно считать крупными, средними и гигантскими. Можно построить самоорганизующуюся карту Кохонена.

Будем использовать следующие параметры обучения:

$$\eta_0 = 0.1 \quad a = 5 \cdot 10^{-4}$$
$$\sigma_0 = 20 \quad b = 1.3 \cdot 10^{-3}$$
$$t_{max} = 1000$$

При градуировании карты мы используем следующие цвета:

- красный: среднее предприятие
- зеленый: крупное предприятие
- синий: предприятие-гигант



В финальной версии карты четко можно разглядеть, в какой области какой из классов находится.

Заключение

Использование метода классификации на основе сетей Кохонена позволяет эффективно кластеризовать многомерную информацию с целью получения новых знаний о множестве рассматриваемых объектов. Кроме того, используя методы самоорганизующихся карт Кохонена, можно получать наглядное представление этой информации.

Плюс обоих методов также в том, что для обучения часто не требуется огромная выборка. Эти методы удобно использовать тогда, когда информации, которой мы располагаем, не так много.

Как всегда, код методов и комментарии к нему представлены в приложении.