Введение в анализ данных, майнор
Семинар №4

За основу взят аналогичный семинар с курса МО-1 ФКН НИУ ВШЭ

1 k ближайших соседей

Рассмотрим задачу классификации: $\mathbb{Y} = \{1, \ldots, K\}$. Пусть дана обучающая выборка $X = (x_i, y_i)_{i=1}^{\ell}$ и функция расстояния $\rho : \mathbb{X} \times \mathbb{X} \to [0, \infty)$. Мы не будем требовать, чтобы функция расстояния являлась метрикой — достаточно, чтобы она была симметричной и неотрицательной. Не будем строить модель на этапе обучения, а вместо этого просто запомним обучающую выборку. Пусть теперь требуется классифицировать новый объект u. Расположим объекты обучающей выборки X в порядке неубывания расстояний до u:

$$\rho(u, x_u^{(1)}) \leqslant \rho(u, x_u^{(2)}) \leqslant \ldots \leqslant \rho(u, x_u^{(\ell)}),$$

где через $x_u^{(i)}$ обозначается i-й сосед объекта u. Алгоритм k *ближайших соседей* (k nearest neighbours, kNN) относит объект u к тому классу, представителей которого окажется больше всего среди k его ближайших соседей:

$$a(u) = \arg\max_{y \in \mathbb{Y}} \sum_{i=1}^{k} [y_u^{(i)} = y]. \tag{1.1}$$

§1.1 Метод парзеновского окна

Проблема формулы (1.1) состоит в том, что она никак не учитывает расстояния до соседей. Действительно, если рассматривается 7 ближайших соседей, и для объекта u ближайшие два объекта находятся на расстоянии $\rho(u, x) \approx 2$, а остальные — на расстоянии $\rho(u, x) \geqslant 100$, то было бы логично обращать внимание только на первые два объекта. Чтобы добиться этого, можно ввести веса в модель:

$$a(u) = \arg\max_{y \in \mathbb{Y}} \sum_{i=1}^{k} w(i, u, x_u^{(i)})[y_u^{(i)} = y].$$

Веса можно делать затухающими по мере роста номера соседа i (например, $w(i, u, x) = \frac{k+1-i}{k}$), но лучше использовать расстояния при их вычислении.

Это делается в *методе парзеновского окна*:

$$a(u) = \arg\max_{y \in \mathbb{Y}} \sum_{i=1}^{k} K\left(\frac{\rho(u, x_u^{(i)})}{h}\right)[y_u^{(i)} = y],$$

где K — это ядро, h — ширина окна.

§1.2 Некоторые проблемы метода

Разберем особенности и проблемы метода k ближайших соседей, возникающие при использовании евклидовой метрики в качестве функции расстояния:

$$\rho(x, z) = \left(\sum_{j=1}^{d} |x_j - z_j|^2\right)^{1/2}.$$

1.2.1 «Проклятие размерности»

Пусть объекты выборки — это точки, равномерно распределенные в d-мерном кубе $[0, 1]^d$. Рассмотрим выборку, состоящую из 5000 объектов, и применим алгоритм пяти ближайших соседей для классификации объекта u, находящегося в начале координат. Выясним, на сколько нужно отступить от этого объекта, чтобы с большой вероятностью встретить пять объектов выборки. Для этого построим подкуб $[0; \varepsilon]^d \subset [0; 1]^d$, $\varepsilon \in (0, 1)$, и положим его объём равным $\delta = \varepsilon^d$. Найдем такое значение δ, при котором в этот подкуб попадет как минимум пять объектов выборки с вероятностью 0.95.

Небольшой пример. Рассмотрим единичный интервал [0,1]. 100 равномерно разбросанных точек будет достаточно, чтобы покрыть этот интервал с частотой не менее 0,01. Теперь рассмотрим 10-мерный куб. Для достижения той же степени покрытия потребуется уже 10^20 точек. То есть, по сравнению с одномерным пространством, требуется в 10^18 раз больше точек. Поэтому, например, использование переборных алгоритмов становится неэффективным при возрастании размерности системы.

Задача 1.1. *Запишите выражение для δ.*

Решение.

$$\min\left\{\delta \,|\, \sum_{k=5}^{5000}\binom{5000}{k}\delta^k(1-\delta)^{5000-k} \geqslant 0.95\right\} \approx 0.0018.$$

■

Таким образом, для того, чтобы найти пять соседей объекта u, нужно по каждой координате отступить на $0.0018^{1/d}$. Уже при $d = 10$ получаем, что нужно отступить на 0.53, при $d = 100$ — на 0.94. Таким образом, при больших размерностях объекты становятся сильно удалены друг от друга, из-за чего классификация на основе сходства объектов может потерять смысл. В то же время отметим, что в рассмотренном примере признаки объектов представляли собой равномерный шум, тогда как в реальных задачах объекты могут иметь осмысленные распределения, позволяющие построение модели классификации даже при больших размерностях.

§1.3 Примеры функций расстояния

1.3.1 Метрика Минковского

Метрика Минковского определяется как:

$$\rho_p(x,z) = \left(\sum_{j=1}^{d}|x_j - z_j|^p\right)^{1/p}$$

для $p \geqslant 1$. При $p \in (0,1)$ данная функция метрикой не является, но все равно может использоваться как мера расстояния.

Частными случаями данной метрики являются:

- Евклидова метрика ($p = 2$). Задает расстояние как длину отрезка прямой, соединяющей заданные точки.
- Манхэттенское расстояние ($p = 1$). Минимальная длина пути из x в z при условии, что можно двигаться только параллельно осям координат.
- Метрика Чебышева ($p = \infty$), выбирающая наибольшее из расстояний между векторами по каждой координате:

$$\rho_\infty(x,z) = \max_{j=1,\dots,d}|x_j - z_j|.$$

- «Считающее» расстояние ($p = 0$), равное числу координат, по которым векторы x и z различаются:

$$\rho_0(x, z) = \sum_{j=1}^{d} [x_j \neq z_j].$$

Отметим, что считающее расстояние не является метрикой.

Отметим, что по мере увеличения параметра p метрика слабее штрафует небольшие различия между векторами и сильнее штрафует значительные различия.

В случае, если признаки неравнозначны, используют взвешенное расстояние:

$$\rho_p(x, z; w) = \left(\sum_{j=1}^{d} w_j |x_j - z_j|^p \right)^{1/p}, \quad w_j \geqslant 0.$$

Задача 1.2. *Рассмотрим функцию $f(x) = \rho_2(x, 0; w)$. Что представляют из себя линии уровня такой функции?*

Решение. Распишем квадрат функции $f(x)$ (форма линий уровня от этого не изменится):

$$f^2(x) = \sum_{j=1}^{d} w_j x_j^2.$$

Сделаем замену $x_j = \frac{x'_j}{\sqrt{w_j}}$:

$$f^2(x') = \sum_{j=1}^{d} {x'_j}^2.$$

В новых координатах линии уровня функции расстояния представляют собой окружности с центром в нуле. Сама же замена представляет собой растяжение вдоль каждой из координат, поэтому в исходных координатах линия уровня являются эллипсами, длины полуосей которых пропорциональны $\sqrt{w_j}$.

■

Вывод: благодаря весам линии уровня можно сделать эллипсами с осями, параллельными осям координат. Это может быть полезно, если признаки имеют разные масштабы — благодаря весам автоматически будет сделана нормировка.

1.3.2 Расстояния между текстами

Пусть заданы векторы x и z. Известно, что их скалярное произведение и косинус угла θ между ними связаны следующим соотношением:

$$\langle x, z\rangle = \|x\|\|z\|\cos\theta.$$

Соответственно, косинусное расстояние определяется как

$$\rho_{\cos}(x,z) = \arccos\left(\frac{\langle x,z\rangle}{\|x\|\|z\|}\right) = \arccos\left(\frac{\sum_{j=1}^{d} x_j z_j}{\left(\sum_{j=1}^{d} x_j^2\right)^{1/2}\left(\sum_{j=1}^{d} z_j^2\right)^{1/2}}\right).$$

Косинусная мера часто используется для измерения схожести между текстами. Каждый документ описывается вектором, каждая компонента которого соответствует слову из словаря. Компонента равна единице, если соответствующее слово встречается в тексте, и нулю в противном случае. Тогда косинус между двумя векторами будет тем больше, чем больше слов встречаются в этих двух документах одновременно.

Один из плюсов косинусной меры состоит в том, что в ней производится нормировка на длины векторов. Благодаря этому она не зависит, например, от размеров сравниваемых текстов, измеряя лишь объем их схожести.

1.3.3 Расстояние Джаккарда

Выше мы рассматривали различные функции расстояния для случая, когда объекты обучающей выборки являются вещественными векторами. Если же объектами являются множества (например, каждый объект — это текст, представленный множеством слов), то их сходство можно измерять с помощью *расстояния Джаккарда*:

$$\rho_J(A,B) = 1 - \frac{|A\cap B|}{|A\cup B|}.$$

2 Методы поиска ближайших соседей

(данный материал является опциональным)

§2.1 Точные методы

Разберем методы поиска ближайших соседей для евклидовой метрики. Будем рассматривать задачу поиска одного ближайшего соседа, все методы несложно обобщаются на случай с $k > 1$.

Если просто перебирать все объекты обучающей выборки, выбирая наиболее близкий к новому объекту, то получаем сложность $O(\ell d)$.

Можно выбрать подмножество признаков, и сначала вычислить расстояние только по этим координатам. Оно является нижней оценкой на полноценное расстояние, и если оно уже больше, чем текущий наилучший результат, то данный объект можно больше не рассматривать в качестве кандидата в ближайшего соседа. Такой подход является чисто эвристическим и не гарантирует сублинейной сложности по размеру обучения.

kd-деревья. Одной из структур данных, позволяющих эффективно искать ближайших соседей к заданной точке, является *kd-дерево*. Оно разбивает пространство на области (каждая вершина производит разбиение по определенной координате), и каждый лист соответствует одному объекту из обучающей выборки. Обходя это дерево определенным образом, можно найти точку из обучения, ближайшую к заданной. Если размерность пространства небольшая (10-20), то данный подход позволяет находить ближайшего соседа за время порядка $O(\log \ell)$.

Экспериментально было установлено, что в пространствах большой размерности сложность поиска ближайшего соседа в kd-дереве сильно ухудшается и приобретает линейный порядок сложности.