Лекция 4. KNN (k-nearest neighbors) - метод k ближайших соседей

Начнем изучение алгоритмов машинного обучения с одного из самых простых и понятных метрических алгоритмов, а именно kNN - метод k ближайших соседей. Давайте сначала попробуем на простом языке объяснить как именно работает этот алгоритм. В первую очередь стоит проговорить, а что такое метрический алгоритм? Метрический алгоритм - это подход, при котором мы используем некоторый функционал метрики расстояния для наших объектов. Имея функционал этой самой метрики, мы можем вычислять так называемых “соседей”, т.е. те точки, которые расположены к рассматриваемому объекту максимально близко. В итоге можно сформулировать этот подход так: проанализируй принадлежность соседей вокруг тебя классам, и какой из классов доминирует, такому классу ты и относишься. Очень похоже на поговорку “Скажи мне кто твой друг и я скажу кто ты 🙂”

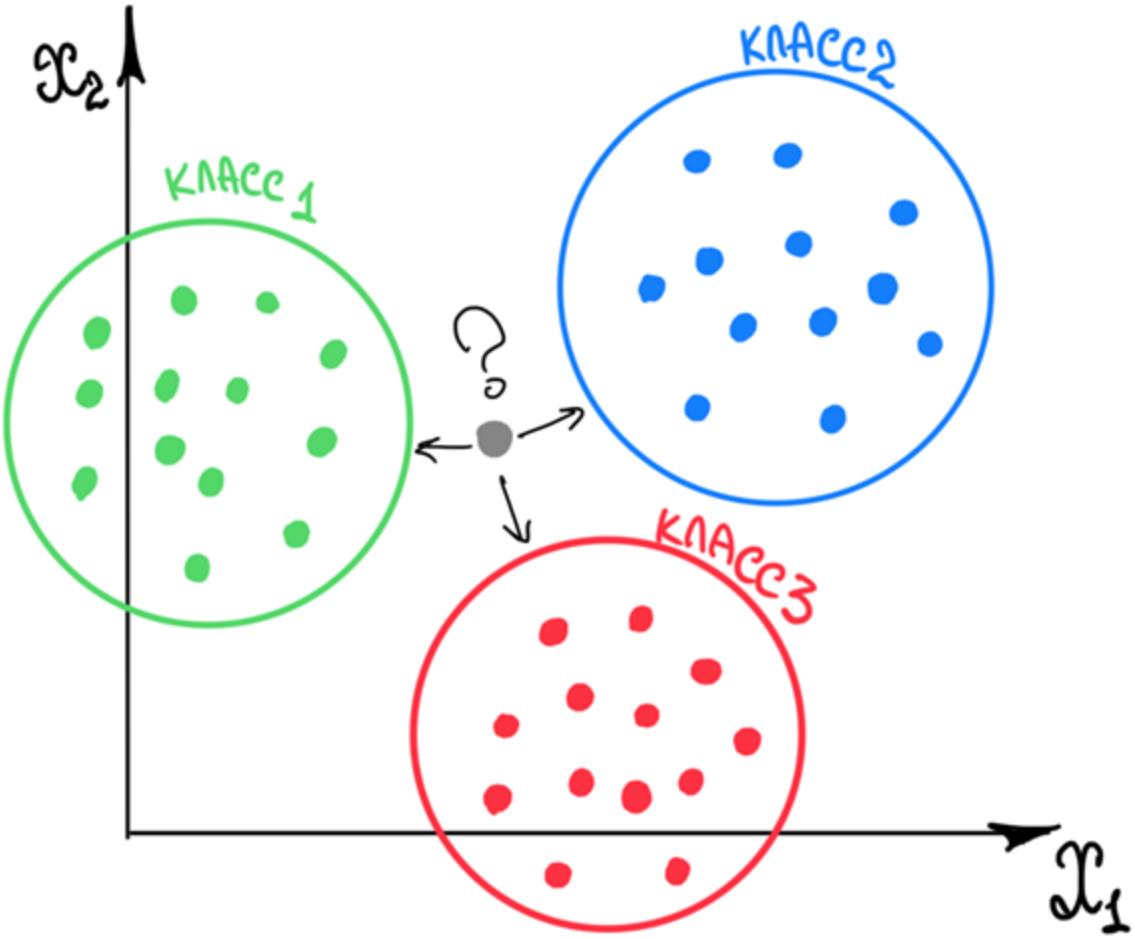
Мы сказали про “близкие” объекты, метрику расстояния, но что же такое близость между объектами в многомерном пространстве на разнотипных признаках и как ее можно формально описать и что такое метрика?

> ❗ **Метрика** - мера расстояния, неотрицательная, симметричная, и если она равно 0, то объекты совпадают. Также часто требуется, чтобы выполнялось неравенство треугольника.

Фундаментальной основой данного алгоритма выступает гипотеза компактности, вкратце она звучит так:

> ❗ Если метрика расстояния между объектами введена удачно, то похожие объекты гораздо чаще лежат в одном классе, чем в разных.

Обычно метрика задается как функция расстояния - метрическая функция от пары объектов, которая сопоставляет каждой паре неотрицательное число, которое как раз-таки и является расстоянием между объектами.



Одной из самой известных метрик расстояния является **метрика Минковского**

$$d_p(a,b) = (\sum_{i=1}^{n} |a_i - b_i|^p)^{\frac{1}{p}}, \text{ где } p \geq 1$$

Частными случаями данной метрики являются:

Евклидова метрика (p = 2). Задает расстояние как длину прямой, соединяющей заданные точки.

Манхэттенское расстояние (p = 1). Минимальная длина пути из x в y при условии, что можно двигаться только параллельно осям координат.

Также есть ещё ряд метрик:

Метрика Чебышева (p = ∞), выбирающая наибольшее из расстояний между векторами по каждой координате:

$$\rho(a,b) = max_{i=1...n}|a_i - b_i|$$

Считающее расстояние, равное числу координат, по которым векторы a и b различаются:

$$\rho(a,b) = \sum_{i=1}^{n}[a_i \neq b_i]$$

Расстояние Махалонобиса определяется следующим образом:

$$\rho(a,b) = \sqrt{(a-b)^T S^{-1}(a-b)}$$

где S симметричная положительно определенная матрица. Напомним, что собственным вектором матрицы S называется такой вектор x, что $Sx = \lambda x$ для некоторого λ. Если матрица S симметричная, то из ее собственных векторов можно составить ортонормированный базис.

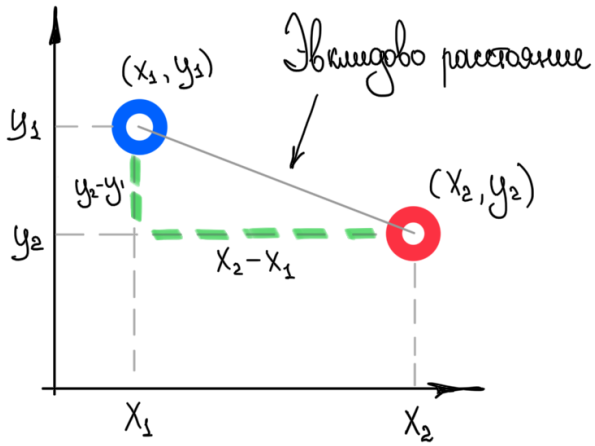
Косинусное расстояние определяется как

$$\rho_{cos}(a,b) = arccos(\frac{\langle a,b \rangle}{||a||\ ||b||}) = arccos(\frac{\sum_{i=1}^{n} a_i b_i}{(\sum_{i=1}^{n} a_i^2)^{\frac{1}{2}} (\sum_{i=1}^{n} b_i^2)^{\frac{1}{2}}})$$

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.

Самым часто используемым примером расстояния для вещественных признаков является евклидово расстояние в признаковом пространстве, которое вычисляется как сумма квадратов координатных разностей.



К сожалению, использование евклидовой метрики оправдано далеко не во всех задачах, многие объекты описываются признаками расстояние между которыми нельзя корректно описать при помощи этой метрики. Например, при анализе текстовых данных мы вынуждены использовать иные подходы для оценки “близости” и одним из возможных подходов является расстояние Левенштейна - это количество вставок и замен символов, которое необходимо произвести, чтобы привести одну строку в другую или же косинусная метрика.

Но вернемся к исходному примеру, где у нас вещественные признаки. Рассмотрим простейший случай, когда k=1, т.е. мы анализируем только одного соседа для выбранного объекта. Для классификация выбранного объекта, в первую очередь мы сформируем массив попарных расстояний выбранного объекта (не учитывая в выборке сам анализируемый объект, конечно) до всех остальных объектов, далее отсортируем данный массив по возрастанию и отнесем анализируемый объект к тому же классу, к которому принадлежит первый объект массива. Уверен, Вы уже хотите возразить, что брать в расчет только одного ближайшего соседа не лучший вариант, ведь намного лучше взять некоторую окрестность вокруг объекта, тем

самым считая класс по нескольким ближайшим соседям при помощи голосования большинством или же усредняя информацию по ближайшим соседям. И это верно! Собственно, из этой идеи и появилась идея данного алгоритма: а давайте упорядочим все объекты обучающей выборки по возрастанию расстояний до анализируемого объекта. В дополнение введем вес соседей $\omega(i, x)$, то есть это функция от объекта и от индекса в отсортированном массиве. Теперь, мы можем усреднить в каждом классе значение весов, и в результате посчитаем оценку близости объекта к классу.

Вот мы и подошли к финалу: для классификации нам достаточно отнести объект к тому классу, для которого эта оценка близости максимальна. Казалось бы, алгоритм прост и его можно реализовать на коленке. Вся сложность заключается в определении единственного параметра k - числа соседей. Когда соседей мало, то велика вероятность поймать некоторую аномальную точку, т.е. объект, который “случайно” попал не в свой класс, как мы убедились на примере. С другой стороны, если k будет очень большим, то это будет больше похоже на некоторый “константный” алгоритм, т.е. почти в любой точке пространства будет одна и та же метка. Основной проблемой метода ближайших соседей является то, что его обучение заключается лишь в запоминании выборки (и, возможно, построении структуры данных для эффективного поиска в ней). При этом не происходит никакой настройки параметров с целью максимизации качества, из-за чего метод не может приспособиться к ненормированным или шумовым признакам. В то же время метод работает с объектами лишь через функцию расстояния, что позволяет использовать его для работы с самыми разнообразными данными (векторами, множествами, строками, распределениями и т.д.).

Алгоритм

Для классификации каждого из объектов тестовой выборки необходимо последовательно выполнить следующие операции:

- Выбрать объект из исходной выборки
- Вычислить расстояние от выбранного объекта до каждого из объектов обучающей выборки
- Отсортировать объекты по возрастанию выбранной метрики расстояния и отобрать k объектов из получившегося отсортированного массива, посчитать веса

- Усреднить в каждом классе значение весов
- Присвоить объекту тот класс, для которого оценка близости максимальна