Метод опорных векторов
Лекция № 7 курса
«Алгоритмы для Интернета»

Юрий Лифшиц*

9 ноября 2006 г.

Содержание

1. Постановка задачи классификации

В этой лекции мы будем рассматривать только бинарную классификацию, то есть имеется только два класса — принадлежит категории и не принадлежит категории (оранжевые точки/зеленые точки). Каждый *объект* классификации является вектором (точкой) в n-мерном пространстве. Каждая координата вектора — это некий признак, и она тем больше, чем больше этот признак выражен у данного объекта. Чем меньше эта координата, тем меньше этот признак соответствует объекту.

Учебная коллекция — это множество векторов $[x_1..x_n] \in \mathbb{R}^n$ и чисел $[y_1..y_n] \in \{-1, 1\}$. Число y_i равно 1 в случае принадлежности соответствующего вектора x_i категории и -1 в противном случае, то есть уже известно, какого цвета точки $[x_1..x_n]$.

Метод опорных векторов (SVM) — это алгоритм, обучающийся различать объекты двух классов.

Постановка задачи: мы имеем коллекцию оранжевых и зеленых точек. Необходимо найти такое правило, по которому новую, неокрашенную точку мы могли бы покрасить в один из этих двух цветов.

*Законспектировал Каширин Виктор.

2. Оптимальная разделяющая гиперплоскость

2.1. Разделение прямой

Линейный классификатор — это первый способ решения поставленной задачи. Идея заключается в следующем: найти прямую, которая отделяет все оранжевые точки от зеленых точек. Если удастся найти такую прямую, то классифицировать каждую новую точку можно будет следующим образом: если точка лежит выше прямой, то она оранжевая, если ниже — зеленая. Формализуем эту классификацию: необходимо найти вектор w такой, что для некоторого граничного значения b и новой точки x_i выполняется:

$$w \cdot x_i > b \Rightarrow y_i = 1;$$
$$w \cdot x_i < b \Rightarrow y_i = -1.$$

Уравнение $w \cdot x_i = b$ описывает гиперплоскость, разделяющую классы в пространстве $\mathbb{R}^n$.

Если скалярное произведение вектора w на x_i больше допускающего значения b, то новая точка принадлежит первой категории, если меньше — второй. На самом деле вектор w перпендикулярен искомой разделяющей прямой, а значение b зависит от кратчайшего расстояния между разделяющей прямой и началом координат.

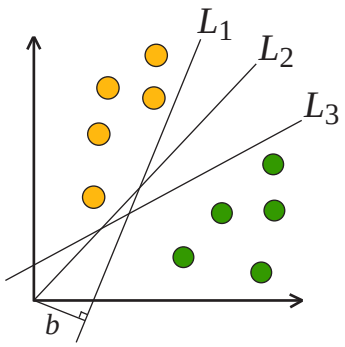

Рис. 1. Пример классифицирующих разделяющих прямых

Рассмотрим пример на рисунке 1. Для прямой L_2 граница b равна 0, а для прямой L_1 — длине перпендикуляра, опущенного на L_1 из начала координат.

2.2. Разделение полосой

Поскольку мы свободны в выборе разделяющей гиперплоскости, то нужно этим воспользоваться для улучшения классификации. Постараемся расположить разделяющую прямую так, чтобы она максимально далеко отстояла от ближайших к ней точек обоих классов, то есть найдем такие вектор w и число b, что для некоторого $\varepsilon > 0$ выполняется:

$$w \cdot x_i \geqslant b + \varepsilon \Rightarrow y_i = 1;$$
$$w \cdot x_i \leqslant b - \varepsilon \Rightarrow y_i = -1.$$

Алгоритм классификации не изменится, если w и b одновременно умножить на одну и ту же константу. Можно воспользоваться этим и выбрать константу таким образом, чтобы для всех пограничных (то есть ближайших к разделяющей гиперплоскости) точек выполнялись условия

$$w \cdot x_i - b = y_i.$$

Это возможно сделать, так как при оптимальном положении разделяющей гиперплоскости все пограничные объекты находятся от нее на одинаковом расстоянии, а остальные объекты находятся дальше.

Домножим тогда неравенства на $\frac{1}{\varepsilon}$ и выберем ε равным единице. Таким образом, для всех векторов x_i из учебной коллекции:

$$w \cdot x_i - b \geqslant 1, \text{ если } y_i = 1;$$
$$w \cdot x_i - b \leqslant -1, \text{ если } y_i = -1.$$

Условие $-1 < w \cdot x_i - b < 1$ задает полосу, разделяющую классы. Ни одна из точек обучающей выборки не может лежать внутри этой полосы. Границами полосы являются две параллельные гиперплоскости с направляющим вектором w. Точки, ближайшие к разделяющей гиперплоскости, лежат точно на границах полосы. Пример разделяющей полосы изображен на рисунке 2.

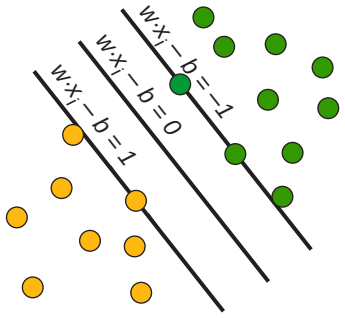

Рис. 2. Разделяющая полоса

Ширина полосы равна $\frac{2}{\|w\|}$ [3]. Чем шире полоса, тем увереннее можно классифицировать документы, соответственно, самая широкая полоса является лучшей.

2.3. Случай линейной разделимости

Сформулируем условия задачи поиска оптимальной разделяющей полосы.

Имеются ограничения: $y_i(w \cdot x_i - b) \geqslant 1$. Вообще здесь нужно писать два неравенства, но так как $y_i \in \{-1, 1\}$, то подставляя y_i в формулу, автоматически получаем необходимое неравенство. При этих ограничениях y_i и x_i константы, так как это элементы учебной коллекции, w и b являются переменными.

Мы хотим найти такие w и b, чтобы выполнялись все линейные ограничения, и при этом как можно меньше была норма вектора w (следовательно, шире разделяющая полоса), то есть необходимо минимизировать:

$$\|w\|^2 = w \cdot w.$$

Такая задача называется задачей *квадратичной оптимизации* — при линейных ограничениях найти минимум квадратичной функции.

2.4. Случай отсутствия линейной разделимости

У рассматриваемого метода есть два существенных недостатка:

- метод не работает, в случае если классы линейно не разделимы;
- предположим, что в учебной коллекции есть ошибка — неправильно классифицирован один или несколько элементов. Из-за этих элементов результирующая разделяющая полоса может сильно отличаться от той, которая получилась бы в случае с корректной учебной коллекцией.

Позволим алгоритму допускать ошибки на обучающих документах. Введем набор дополнительных переменных $\xi_i \geqslant 0$, характеризующих величину ошибки на объектах $x_i \in [x_1..x_n]$. Это позволяет смягчить ограничения-неравенства:

$$y_i(w \cdot x_i - b) \geqslant 1 - \xi_i.$$

Предполагается, что если $\xi_i = 0$, то на документе x_i ошибки нет. Если $\xi_i > 1$, то на документе x_i допускается ошибка. Если $0 < \xi_i < 1$, то объект попадает внутрь разделительной полосы, но относится алгоритмом к своему классу.

Переформулируем задачу поиска оптимальной разделяющей: при данных ограничениях минимизировать сумму

$$\|w\|^2 + C \sum \xi_i.$$

Коэффициент C — это параметр настройки метода, который позволяет регулировать отношение между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки. Этот параметр выбирается вручную для каждой ситуации отдельно.

Заметим, что эта задача осталась задачей *квадратичного программирования.*

Итак, это первый полученный результат, который говорит о том, какой линейный классификатор необходимо выбирать. Мы пришли к выводу, что чем шире полоса и чем меньше штрафов, тем классификатор лучше. Таким образом, задача обучения классификатора свелась к задаче оптимизации, поиску линейного классификатора, который лучше всего подходит к тем учебным данным, которые у нас имеются. Естественно, возникает вопрос, как решать поставленную задачу оптимизации. Мы разберемся в этом в следующей части.

3. Обучение машины опорных векторов

Как известно, чтобы найти минимум функции, необходимо исследовать ее производную. Однако в нашем случае помимо функции нам заданы линейные ограничения, которые необходимо учитывать при минимизации. Множество точек, удовлетворяющих ограничениям, в n-мерном пространстве представляет собой многогранник: пространство делится несколькими гиперплоскостями или полуплоскостями в зависимости от того, стоят в линейных ограничениях соответственно знаки равенства или неравенства. Пересечение нескольких полупространств и дает многогранник, который может также в некотором месте продолжаться в бесконечность. В этом ограниченном пространстве и необходимо найти минимум.

Решить эту задачу можно с помощью метода Лагранжа. Лагранж попытался свести задачу поиска условного минимума (с ограничениями) к задаче поиска безусловного минимума (без ограничений), чтобы затем воспользоваться стандартным методом поиска минимума функции. Для этого необходимо изменить целевую функцию, то есть ту функцию, которую необходимо минимизировать.

Метод Лагранжа при нескольких условиях-равенствах

Если условия представляют собой равенства, то множество, на котором ищется минимум, есть пересечение нескольких гиперплоскостей, и является неким подпространством. Введем дополнительные переменные λ_i, называемые множителями Лагранжа, и вместо исходной целевой функции F будем рассматривать такую:

$$F - \sum_i \lambda_i G_i,$$

где G_i — уравнения ограничений. Лагранж доказал, что в той точке, в которой у функции F достигается условный минимум, при фиксированном наборе λ_i у новой функции также достигается минимум, но при этом уже по всем переменным x, без ограничений. Зная это, можно воспользоваться следующим алгоритмом поиска минимума:

1. Берем все производные новой целевой функции по переменным x и приравниваем их к нулю.
2. С помощью получившихся уравнений выражаем переменные x через $\lambda_1 \dots \lambda_n$.

3. Так как целевая функция имеет минимум, и при этом выполнены линейные ограничения, то переменные x, выраженные через λ_i, подставляем в n уравнений для $G_1 \dots G_n$. Подставив, получаем систему из n уравнений для n неизвестных. Решив систему, найдем значения λ_i, следовательно, найдем значения переменных x при которых достигается условный минимум.

Приведем пример. Будем минимизировать $F(x) = x_1 + x_2$ при условии $G(x)$: $x_1^2 + x_2^2 = 1$. Составим новую целевую функцию: $x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 1)$. Нам известно, что при некотором λ в той точке, в которой минимизируется $F(x)$, минимизируется и получившаяся целевая функция. Рассмотрим частные производные по x_1 и x_2, равные соответственно $1 - 2\lambda x_1$ и $1 - 2\lambda x_2$. Приравняем их к нулю, и, выразив x_1 и x_2 через λ, получим, что $x_1 = x_2 = \frac{1}{2}\lambda$. Подставим получившиеся значения x_1 и x_2 в уравнение ограничения: $\frac{1}{4}\lambda^2 + \frac{1}{4}\lambda^2 = 1$, и получим $\lambda = \pm\sqrt{2}$. Теперь необходимо исследовать оба случая и рассмотреть, чему будет равна функция. Оказывается, что в одной точке функция достигает максимума, в другой — минимума. Задача решена.

Метод Лагранжа при нескольких условиях-неравенствах

Будем рассматривать целевую функцию

$$F - \sum_i \lambda_i G_i$$

при условии, что все неравенства $G_i \geqslant 0$, и все $\lambda_i \geqslant 0$.

Теорема Лагранжа, в случае если ограничения являются неравенствами, звучит следующим образом. Если в точке x достигается условный минимум исходной целевой функции, то при условии, что выполняется равенство нулю производных по x новой целевой функции, существует такой набор λ_i, что в этой же точке x достигается минимум новой целевой функции, но уже глобально по всем x. При этом для каждого λ_i верно следующее: либо λ_i равно нулю, и соответствующее ограничение не активно, либо λ_i не равно нулю, и соответствующее ограничение выполняется, но является при этом уже равенством.

Теперь алгоритм поиска минимума следующий:

1. Взять все производные новой целевой функции по переменным x и приравнять их к нулю.
2. С помощью получившихся уравнений выразить x через $\lambda_1 \dots \lambda_n$. Для каждого λ_i верно следующее:
   - или λ_i равно нулю, и соответствующее ограничение G_i оказывается неактивным, оно не участвует в целевой функции;
   - или, если λ_i не равно нулю, то соответствующее активное ограничение G_i является равенством.
3. Подставить x в активные ограничения. Для n переменных решить n «или-уравнений».

Вернемся к исходной задаче: при ограничениях-неравенствах

$$y_i(w \cdot x_i - b) \geqslant 1 - \xi_i,\ \xi_i \geqslant 0$$

минимизировать квадратичную функцию

$$\frac{1}{2}\|w\|^2 + C\sum_i \xi_i.$$

Формулируя эту задачу в терминах метода Лагранжа, получаем, что необходимо найти минимум по w, b, ξ_i и максимум по λ_i функции

$$\frac{1}{2}w \cdot w + C\sum_i \xi_i - \sum_i \lambda_i(\xi_i + y_i(w \cdot x_i - b) - 1)$$

при условиях

$$\xi_i \geqslant 0,\ \lambda_i \geqslant 0.$$

Эта новая целевая функция называется *Лагранжианом*. Необходимым условием метода Лагранжа является равенство нулю производных Лагранжиана по переменным w и b. Взяв производную целевой функции по w, выражаем вектор w через множители Лагранжа:

$$w = \sum_i \lambda_i y_i x_i.$$

Из этого следует, что искомый вектор w должен быть линейной комбинацией учебных векторов, причем только тех, для которых $\lambda_i \neq 0$.

Если $\lambda_i > 0$, то документ обучающей коллекции x_i называется *опорным вектором* (*support vector*).

Теперь уравнение разделяющей гиперплоскости выглядит так:

$$\sum_i \lambda_i y_i \; x_i \cdot x - b = 0,$$

где x_i — это документ, который мы хотим классифицировать.

В случае если бы не были введены штрафы, значение b можно было бы найти как среднее между значениями скалярного произведения w на вектора первой категории и второй. Если же штрафы введены, то необходимо найти полосу, при которой сумма штрафов минимальна.

Взяв производную целевой функции по b, получаем что

$$\sum_i \lambda_i y_i = 0.$$

Подставляя w, выраженное через λ_i, в Лагранжиан, получаем новую, «двойственную» задачу: решению соответствуют такие значения множителей, при которых достигается максимум

$$\sum_i \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j)$$

при условиях

$$\sum_i \lambda_i y_i = 0,$$

$$0 \leqslant \lambda_i \leqslant C,$$

где второе ограничение — это дополненное исходное ограничение на λ_i, так как штрафы ξ_i тоже выражаются через λ_i.

Полезно заметить, что матрица коэффициентов при $\lambda_i \lambda_j$ положительно определена, из чего следует, что полученная функция выпуклая, а значит любой локальный максимум (минимум) этой функции является ее глобальным максимумом (минимумом).

В итоге, исходная задача квадратичной оптимизации была переформулирована в задачу для множителей Лагранжа. Целевая функция зависит не от самих x_i, а от скалярных произведений между ними. Само скалярное произведение есть некоторая метрика близости, которая, например, для векторов показывает количество совпадений признаков, и во многих практических случаях использование скалярных произведений учебных векторов гораздо удобнее манипуляций самими учебными векторами.

Решение двойственной задачи: метод последовательных оптимизаций

Алгоритм решения двойственной задачи таков:

1. Начать с набора λ_i, удовлетворяющих ограничениям.
2. С помощью хитрых эвристик выбрать из набора пару улучшаемых коэффициентов $\lambda_i \lambda_j$.
3. При фиксированных значениях остальных множителей из набора и имеющихся ограничениях $y_i \lambda_i + y_j \lambda_j = y_i \lambda_i^{old} + y_j \lambda_j^{old}$ и $0 \leqslant \lambda_i, \lambda_j \leqslant C$ выбрать оптимальную пару значений (*мини-оптимизация*).
4. Продолжать процесс, повторяя шаги 2 и 3, до наступления *стоп-условий*.

Таким образом, на каждом шаге мы изменяем два коэффициента так, чтобы оставались выполненными ограничения, и при этом функция возросла. Множество таких пар $\lambda_i \lambda_j$, удовлетворяющих ограничениям, лежит на диагонали в прямоугольнике со сторонами равными C. Действительно, из ограничения следует, что $y_i \lambda_i + y_j \lambda_j = const$, что в свою очередь задает уравнение прямой. Ее отрезок, лежащий в квадрате, в котором λ_i и λ_j изменяются в пределах от 0 до C, и является множеством точек, удовлетворяющих ограничениям. Среди этого множества нужно выбрать ту точку, при которой максимизируемая функция принимает наибольшее значение. В нашем случае имеются всего две переменные, и нужно просто найти максимум квадратичной функции.

Стоп-условие — это некоторая эвристика. Нам известно, что на каждом шаге максимизируемая функция должна возрастать, тогда в качестве стоп-условия может быть выбрано следующее: за последние сто шагов функция возросла менее чем на какое-нибудь число ε. Исследуемая функция выпуклая, и после того, как она перестанет расти и достигнет максимума, можно быть уверенными, что этот максимум глобальный.

4. Расширение пространства признаков

Метод классификации разделяющей полосой имеет два недостатка:

- в поиске разделяющей полосы существенное значение имеют только пограничные точки;
- во многих случаях найти оптимальную разделяющую полосу невозможно.

Следовательно, необходимо как-то улучшить метод. Как мы помним, условия задачи оптимизации, сформулированной с помощью множителей Лагранжа, зависели только от скалярных произведений между учебными документами. Поэтому для улучшения метода можно попытаться изменить скалярное произведение. Здесь на помощь приходит идея *расширенного пространства*.

Построение машины опорных векторов:

1. Выберем отображение $\phi(x)$ векторов x в новое, *расширенное* пространство.
2. Автоматически получается новая функция скалярного произведения $K(x, y) = \phi(x) \cdot \phi(y)$. На практике обычно выбирают не отображение $\phi(x)$, а сразу функцию $K(x, y)$, которая могла бы быть скалярным произведением при некотором отображении $\phi(x)$. Функция $K(x, y)$ называется *ядром*. Эта функция есть главный параметр настройки машины опорных векторов.
3. Находим разделяющую гиперплоскость в новом пространстве: с помощью функции $K(x, y)$ мы составляем новую матрицу коэффициентов для задачи оптимизации, подставляя вместо $(x_i \cdot x_j)$ значение $K(x_i, x_j)$, и решаем новую задачу оптимизации.
4. Найдя w и b, получаем классифицирующую поверхность $w \cdot \phi(x) - b$ в новом, расширенном пространстве.

5. Примеры

Пример 1

Рассмотрим наглядный пример перехода к расширенному пространству, изображенный на рисунке 3. Как видно, эти оранжевые и зеленые точки не разделяются никакой полосой. Если же мы перенесем

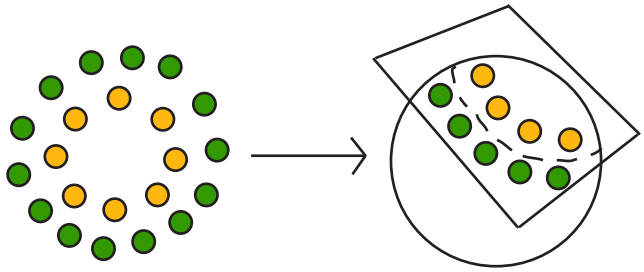

Рис. 3. Пример перехода к расширенному пространству

эти точки на сферу, то тогда они разделяются плоскостью, которая срезает часть сферы вместе с оранжевыми точками. Таким образом, выгнув пространство с помощью отображения ϕ, можно легко найти разделяющую гиперплоскость.

Пример 2

$$\phi(x) = \phi((x_1, x_2)) \to (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$
$$K(x, y) = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 = (x \cdot y)^2$$

Мы видим, что не обязательно считать $\phi(x)$ и $\phi(y)$, чтобы сосчитать скалярное произведение. Можно сразу сказать, что новое скалярное произведение есть старое скалярное произведение в квадрате. Это очень полезно, так как в данном примере пространство увеличилось всего на одну координату, а в практических ядрах оно может увеличиться во много раз, и в таких случаях считать скалярное произведение через функцию ϕ может быть очень невыгодно.

Пример 3

$$\phi(x) = (1, c_{10\ldots0}x_1, \ldots, c_{110\ldots0}x_1x_2, \ldots, c_{0\ldots0d}x_k^d)$$

В данном случае отображение делает следующее: оно берет вектор x, и делает столько координат, сколько бывает мономов степени не больше d, добавляя некоторые коэффициенты. Моном степени d есть произведение некоторых координат вектора в таких степенях, что их сумма не превышает d. Мы придумали это отображение, уже имея в виду, что хотим получить скалярное произведение вида:

$$K(x, y) = (1 + x \cdot y)^d$$

Раскроем скалярное произведение $x \cdot y = x_1y_1 + \ldots + x_ny_n$ и подставим в скобку $(1 + x \cdot y)$. Возведем ее в степень d и получим большую сумму, из которой можно заметить, что коэффициент при соответствующем мономе-координате равен корню из числа слагаемых в этой сумме, в которых присутствует этот моном.

Общая формула для коэффициента координаты-монома вида $x_1^{\alpha_1} \ldots x_k^{\alpha_k}$ такова:

$$C_{\alpha_1\ldots\alpha_k} = \sqrt{\frac{d!}{\alpha_1! \ldots \alpha_k!(d - \alpha_1 - \ldots - \alpha_k)!}}.$$

Итак, с помощью введенного отображения мы смогли найти разделяющую гиперплоскость $w \cdot \phi(x) - b$. На самом деле это многочлен от координат исходного вектора x, причем w_i — это коэффициент этого многочлена. В расширенном пространстве мы ищем w_i, для которых гиперплоскость $w \cdot \phi(x) - b$ будет разделением, однако в исходном пространстве мы ищем полином, который опишет кривую разделения первого и второго классов. Поэтому, выбрав соответствующее ядро, мы выбираем класс разделяющих поверхностей. Ядру $K(x, y) = (1 + x \cdot y)^d$ соответствуют все полиномы, то есть мы ищем *оптимальную полиномиальную разделяющую поверхность*.

Итоги

Преимущества SVM:

- на тестах превосходит другие методы;
- при различных выборах ядер можно эмулировать другие подходы. Например, большой класс нейронных сетей можно представить в виде SVM с определенными ядрами;
- теоретическое обоснование: итоговое правило мы выбираем не с помощью некоторых эвристик, а согласно оптимизации некоторой функции.

Недостатки:

- мало параметров для настройки: после того как мы зафиксировали ядро, единственным варьируемым параметром остается коэффициент ошибки C;
- не очень понятно, как выбирать ядро;
- медленное обучение.

Метод опорных векторов сводит обучение классификатора к задаче квадратичной оптимизации.

Задача квадратичной оптимизации решается эвристическими алгоритмами путем последовательного уменьшения целевой функции.

Для построения нелинейных классификаторов используется отображение исходных объектов в расширенное пространство признаков.

Источники

[1] Wikipedia, Support Vector machine
http://en.wikipedia.org/wiki/Support_vector_machine

[2] CJC Burges. A Tutorial on Support Vector Machines for Pattern Recognition
http://www.music.mcgill.ca/~rfergu/adamTex/references/Burges98.pdf

[3] Константин Воронцов. Лекция по методу опорных векторов
http://www.ccas.ru/voron/download/SVM.pdf

[4] John Platt. Sequential Minimal Optimization
http://research.microsoft.com/users/jplatt/smo.html

[5] Страница курса
http://logic.pdmi.ras.ru/~yura/internet.html