ЦИФРОВАЯ КУЛЬТУРА
УНИВЕРСИТЕТ ИТМО

Метод опорных векторов (SVM)

Высшая Школа Цифровой Культуры
Университет ИТМО
dc@itmo.ru

Содержание

1 Метод опорных векторов (SVM): линейно разделимая выборка

1.1 Введение

Здравствуйте, уважаемые слушатели. В этой лекции мы рассмотрим еще один эффективный метод, используемый для решения задачи классификации, – так называемый метод опорных векторов или SVM (Support Vector Machines). Сам по себе метод был разработан в 1990-ых годах и с тех пор набрал заслуженную популярность. Но давайте, прежде чем переходить к строгому описанию, попробуем понять идею предлагаемого подхода на интуитивном и геометрическом уровнях.

1.2 Некоторые начальные соображения

Итак, везде далее в этой лекции мы будем предполагать, что рассматриваются лишь два класса. Иными словами, мы будем обсуждать двухклассовую классификацию. Как уже не раз отмечалось в разговорах про различные классификаторы, вводимое таким образом ограничение хоть и является достаточно обременительным (то есть явно сужает класс задач, которые могут быть решены обсуждаемым методом), все равно позволяет рассматривать широкий спектр вопросов и проблем – проблем, допускающих лишь два возможных ответа: «да - нет», плюс-минус, и подобные. Кроме того, существуют методы обобщения двухклассовых классификаторов до многоклассовых. В этой лекции мы не будем касаться этих методов, вы можете с ними ознакомиться в дополнительных материалах.

Сейчас же вернемся к нашей цели – на картинках и простых примерах понять, что за идея заложена в SVM. Для этого давайте обратимся к следующему рисунку, к рисунку 1. Перед нами 10 тренировочных данных x_1, x_2, ..., x_{10}, каждое из которых обладает двумя атрибутами (условно, $x_i = (x_{i1}, x_{i2})$) и одним откликом, условно «$+1$» и «-1». Для наглядности, данные с откликом «$+1$» обозначены красными кружками, а данные с откликом «-1» – синими. Так, например, к классу «$+1$» относятся данные с предикторами $(1, 0)$, $(2, 1)$, $(1, 2)$, $(0.5, 4)$, а к классу «-1» данные с предикторами $(4, 1)$, $(3.5, 3)$, $(2.5, 5)$ ну и так далее. Для большей наглядности данные приведены в следующей таблице:

Объект	Значения X_1	Значения X_2	Отклик
x_1	$x_{11} = 0.5$	$x_{12} = 4$	$+1$
x_2	$x_{21} = 1$	$x_{22} = 0$	$+1$
x_3	$x_{31} = 1$	$x_{32} = 2$	$+1$
x_4	$x_{41} = 2$	$x_{42} = 1$	$+1$
x_5	$x_{51} = 2.5$	$x_{52} = 5$	-1
x_6	$x_{61} = 3$	$x_{62} = 7$	-1
x_7	$x_{71} = 3.5$	$x_{72} = 3$	-1
x_8	$x_{81} = 4$	$x_{82} = 1$	-1
x_9	$x_{91} = 4$	$x_{92} = 6$	-1
x_{10}	$x_{10\ 1} = 5$	$x_{10\ 2} = 4$	-1

Что мы видим, исходя из наглядных соображений? А то, что красные достаточно четко отделены от синих, причем данные можно разделить прямой (то есть, как говорят, выборка линейно разделима). Но как эту прямую провести? Посмотрите, на рисунке 2 предложено несколько вариантов постро-

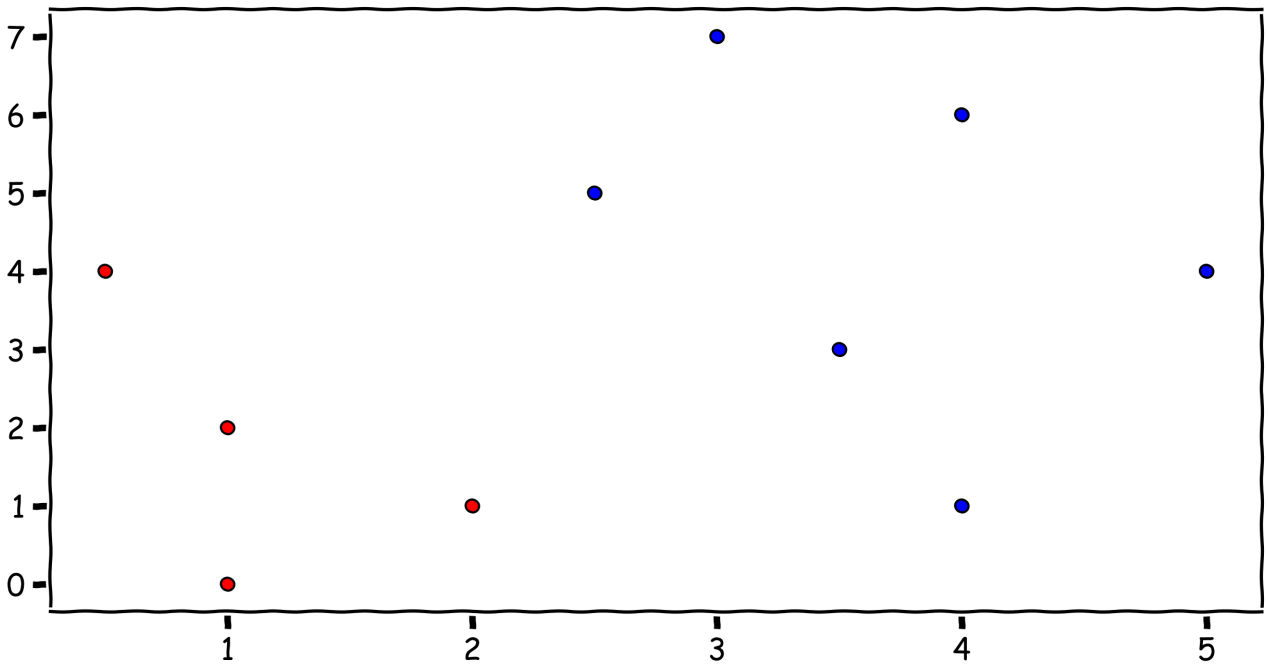


Рис. 1: Линейно разделимая выборка. Красные – класс «+1», синие – класс «−1»

ения такой прямой. Какое построение кажется более удачным? Ну смотрите, конечно же логично выдвинуть следующее требование: чтобы элементы разных классов находились как можно дальше от разделяющей прямой. Именно в этом случае, с точки зрения логики и здравого смысла, мы «лучше всего» различаем представленные классы. Значит, скорее всего оранжевая прямая

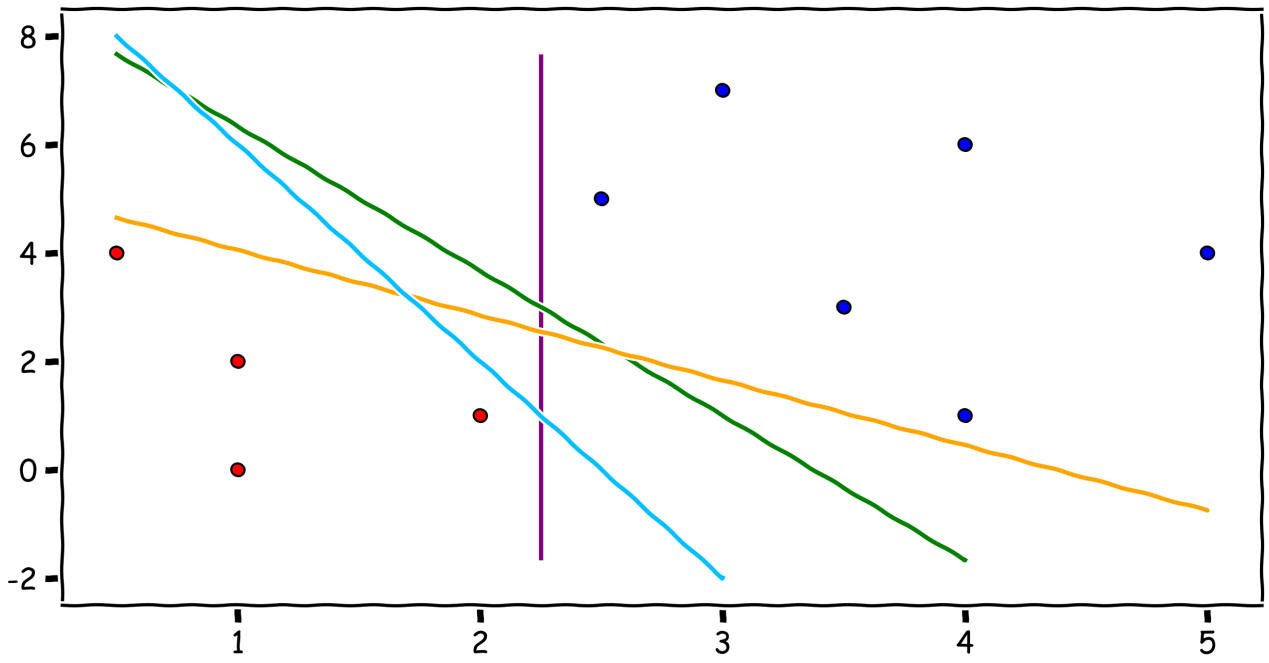


Рис. 2: Несколько возможных прямых, разделяющих два класса

– не самый лучший кандидат. Не смотря на то, что она, конечно же, разделяет красных и синих, расстояния до нее от ближайших представителей чрезвычайно мало (ближайшим представителем синих является объект с координатами $(4, 1)$, а ближайшим представителем красных является объект с координатами $(0.5, 4)$).

А что по поводу прямой голубого цвета? С ней, похоже, тоже не все хорошо. Дело в том, что расстояние от нее до ближайшего представителя красных необоснованно меньше, чем до ближайшего представителя синих. Почему она так сильно сдвинута вниз? Чем это подкрепляется? Немного подумав, опираясь на геометрические соображения, мы приходим ко второму требованию: расстояние от разделяющей прямой до ближайшего представителя как одного, так и другого класса, должно быть одинаковым (грубо говоря, прямая должна проходить «посередине» между классами). Значит, из предложенных нами вариантов в рассмотрении остаются только две прямые: зеленая и фиолетовая. Давайте поговорим о них подробнее, а заодно выясним и третье, финальное требование к желаемой разделяющей прямой.

Обратимся к вновь модифицированному рисунку, к рисунку 3. На нем фиолетовым и зеленым пунктиром проведены прямые, проходящие через ближайших представителей классов до фиолетовой и зеленой разделяющей прямых, соответсвенно.

Интуитивно понятно, что зеленая прямая лучше, ведь расстояние между зелеными пунктирными линиями больше, чем между фиолетовыми, а значит и классификация «более уверенная». Вот мы и поняли третье требование –

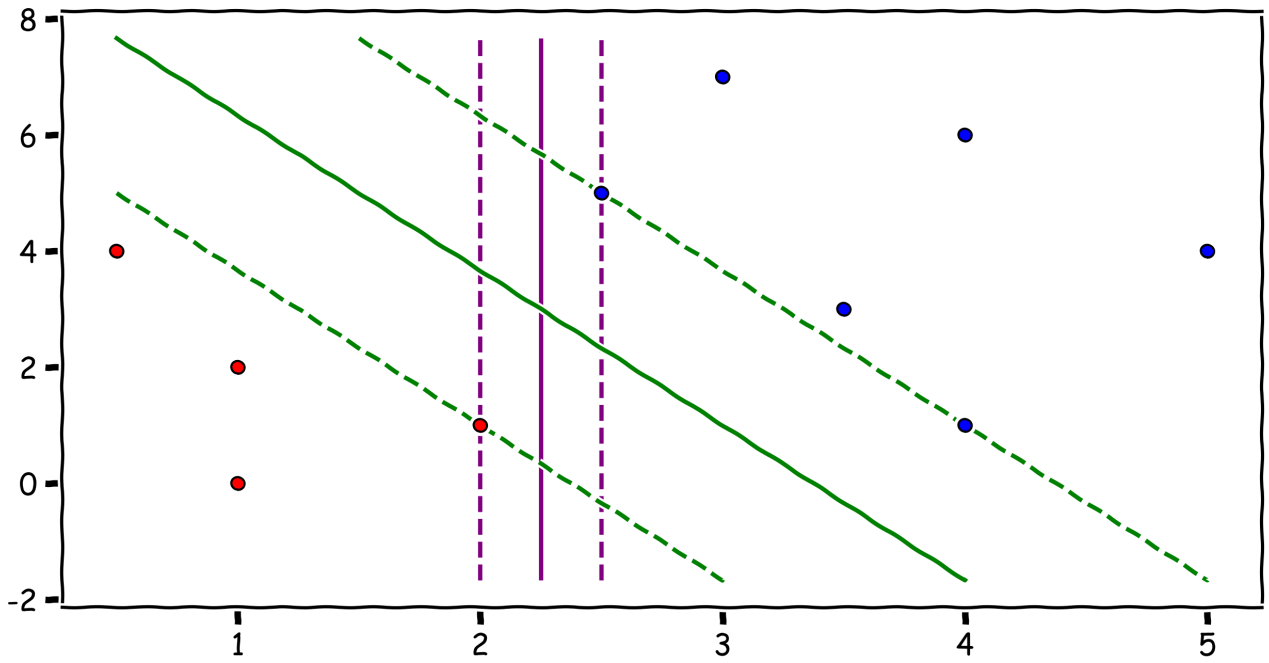


Рис. 3: Разделяющие полосы

получающаяся разделяющая полоса должна иметь наибольшую ширину.

Совсем на бытовом уровне разделяющую прямую и пунктирные линии по ее бокам можно представлять для себя, как проспект в каком-нибудь городе. Разделяющая прямая – это сплошная линия, делящая проезжую часть на две равные части, а пунктир – это граница проезжей части с тротуаром. Конечно, чем расстояние между тротуарами по обе стороны проезжей части больше, тем движению удобнее – может ехать больше машин, а так как движения в одну и другую сторону равноправны, то разделительная полоса расположена ровно посередине. С другой стороны, расстояние между пунктирными линиями не может быть сколь угодно большим, ведь тогда на проезжей части окажутся «пешеходы» – наши тренировочные данные.

Итак, эти геометрические соображения и ведут к основной идее SVM в случае линейно разделимой выборки – построению разделяющей гиперплоскости, отстающей на максимальные, но равные расстояния от ближайших к ней тренировочных представителей данных классов. Но как эта задача объясняется компьютеру, как работать в многомерных пространствах, когда наглядности, как в рассмотренном нами примере, нет, и как все-таки решать задачу классификации, когда разделяющая плоскость построена? Ну что, давайте разбираться: начнем с классификации и многомерности.

1.3 Немного о гиперплоскости и классификации с ее помощью

Изучая линейную и логистическую регрессии, мы уже сталкивались с понятием гиперплоскости. В двумерном пространстве гиперплоскость l – это множество точек, описываемых уравнением

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 = 0,$$

в котором хотя бы один из коэффициентов, стоящих перед X_1 и X_2, отличен от нуля, то есть $\theta_1^2 + \theta_2^2 \neq 0$. Как известно из средней школы, написанное уравнение есть не что иное, как уравнение прямой на плоскости, а вектор $n = (\theta_1, \theta_2)$ – это вектор нормали этой прямой. Вектор нормали прямой обладает следующим полезным свойством: он перпендикулярен прямой. Дословно это означает следующее: если точки A и B лежат на l, то векторы AB и n перпендикулярны (ортогональны). В терминах скалярного произведения, которое тоже обсуждалось ранее, это означает, что

$$(AB, n) = 0.$$

В трехмерном пространстве гиперплоскость l – это множество точек, описываемых уравнением

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 = 0,$$

в котором хотя бы один из коэффициентов, стоящих перед X_1, X_2 и X_3, отличен от нуля, то есть $\theta_1^2 + \theta_2^2 + \theta_3^2 \neq 0$. Опять же, легко понять, что написанное уравнение – это уравнение плоскости в пространстве, а вектор $n = (\theta_1, \theta_2, \theta_3)$ – это вектор нормали этой плоскости. Вектор нормали перпендикулярен плоскости, и понимать это нужно в том же самом смысле, как и в случае прямой: если точки A и B лежат на l, то векторы AB и n перпендикулярны (ортогональны). В терминах скалярного произведения получаем, что

$$(AB, n) = 0.$$

Наверное, дальнейшая логика ясна, поэтому можно принять общее определение.

Определение 1.3.1 *В p-мерном пространстве гиперплоскостью l называется множество точек, координаты которых удовлетворяют уравнению вида*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

в котором $\theta_1^2 + \theta_2^2 + ... + \theta_p^2 \neq 0$.

Кроме того, логично принять следующее определение

Определение 1.3.2 *Пусть дана гиперплоскость l, задаваемая уравнением*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0.$$

Вектор $n = (\theta_1, \theta_2, ..., \theta_p)$ называется нормалью гиперплоскости l.

Можно доказать, что нормаль гиперплоскости перпендикулярна ей ровно в том смысле, что был озвучен уже дважды ранее: вектор, составленный по любым двум точкам, лежащим на гиперплоскости, перпендикулярен (ортогонален) к её нормали.

Хорошо, а как проводить классификацию, используя гиперплоскость? Оказывается, очень легко, ведь, как не сложно понять геометрически, гиперплоскость разбивает пространство на две части (представьте себе прямую, разбивающую плоскость, и плоскость, разбивающую трехмерное пространство). Осталось только понять, в какую из двух частей попадает тестовое наблюдение. В случае, когда размерность пространства больше, чем 3, сделать это наглядно достаточно проблематично, поэтому используют следующий алгебраический подход.

Пусть гиперплоскость l задается уравнением

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0.$$

Рассмотрим функцию

$$f(X) = f(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p.$$

Если точка X лежит на гиперплоскости, то, согласно определению последней, $f(X) = 0$. Если же точка X не лежит на гиперплоскости, то либо $f(X) > 0$, что означает, что точка лежит по одну сторону от гиперплоскости (то есть попадает в один класс, который мы будем в дальнейшем обозначать «+1»), либо $f(X) < 0$, что означает, что точка лежит по другую сторону от гиперплоскости (и попадает в другой класс – класс «−1»). В итоге, чтобы определить, с какой стороны от гиперплоскости находится тестовое наблюдение (и к какому классу принадлежит тестовая точка), достаточно просто вычислить значение функции $f(X)$ на нем.

Замечание 1.3.1 *В этот момент полезно остановиться и задуматься: знак выражения $f(X)$ при одном и том же X зависит от того, какое направление нормали выбрано, а значит от этого зависит и нумерация классов. Смотрите, пусть гиперплоскость l задается выражением*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0.$$

Тогда при любом $k \neq 0$ уравнение

$$k\left(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p\right) = 0$$

задает ту же самую гиперплоскость. Но если $k < 0$, то нормаль этой гиперплоскости с координатами $(k\theta_1, k\theta_2, ..., k\theta_p)$ имеет направление, противоположное направлению нормали $n = (\theta_1, \theta_2, ..., \theta_p)$.

В частности по этой причине положительным классом мы будем называть точки той части пространства после разделения, в сторону которой направлена нормаль (так было, кстати, и при рассмотрении логистической регрессии).

Давайте поясним сказанные слова на примере. Пусть даны тренировочные данные, что и раньше, и предложено 3 варианта разделения всей плоскости на две части (каждому варианту отвечает прямая своего цвета). Будем счи-

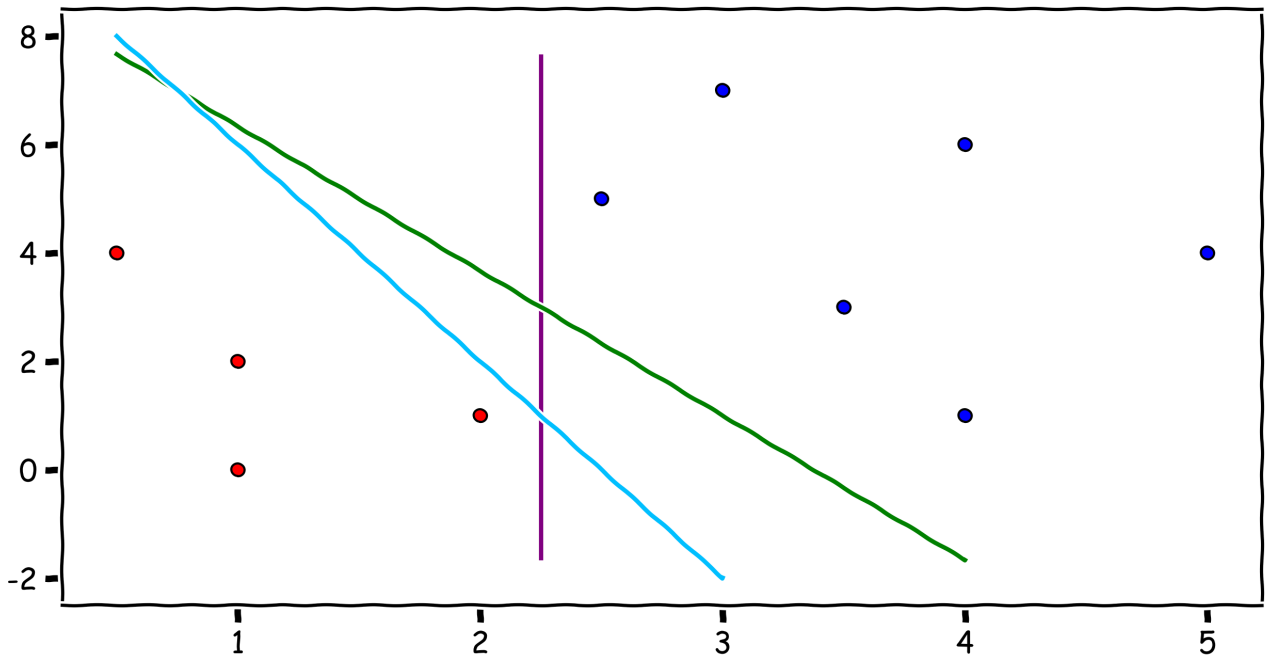


Рис. 4: Линейно разделимая выборка. Красные – класс «+1», синие – класс «−1»

тать, что красные точки отвечают классу «+1», а синие – классу «−1». При этом прямые задаются следующими уравнениями: голубая задается уравнением $-4X_1 - X_2 + 10 = 0$, фиолетовая уравнением $X_1 - 2.25 = 0$, а зеленая уравнением $-2.667X_1 - X_2 + 8.999 = 0$. Правомочно ли так считать, если мы выбрали нумерацию классов?

Смотрите, нормаль голубой прямой имеет коэффициенты $(-4, -1)$ и направлена как раз в сторону красных точек. Нормаль фиолетовой прямой имеет координаты $(1, 0)$ и направлена в сторону синих точек. Если мы хотим

сохранить нумерацию классов при классификации, то уравнение фиолетовой прямой стоит переписать, как $-X_1 + 2.25 = 0$ (всё уравнение просто умножено на -1, чтобы изменить направление нормали). Нормаль зеленой прямой имеет координаты $(-2.667, -1)$, направлена в сторону красных точек, значит никаких проблем нет. Итак, для классификации возникает три функции (относительно каждой из прямых):

$$f_{\text{голубая}}(X_1, X_2) = -4X_1 - X_2 + 10,$$

$$f_{\text{фиолетовая}}(X_1, X_2) = -X_1 + 2.25,$$

$$f_{\text{зеленая}}(X_1, X_2) = -2.667X_1 - X_2 + 8.999.$$

Теперь давайте произведем классификацию тестового оранжевого объекта с координатами $(2, 3)$. По рисунку сразу понятно: относительно голубой прямой

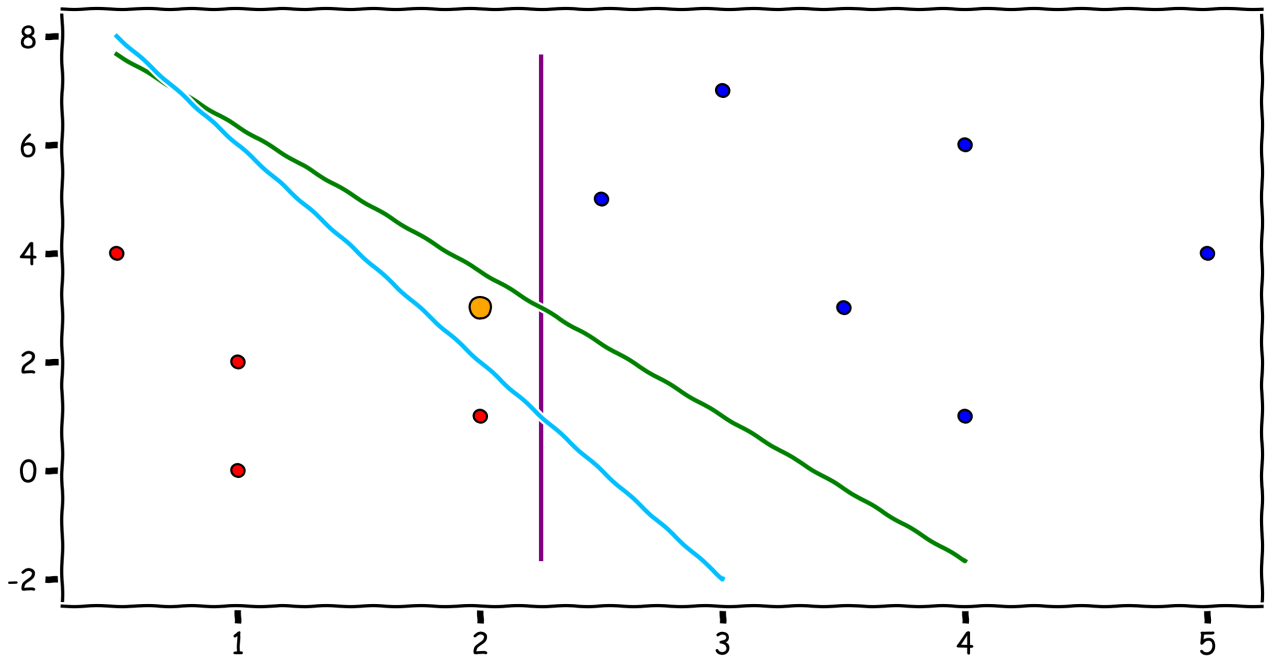


Рис. 5: Классификация нового объекта с координатами $(2, 3)$

объект будет отнесен к классу «-1», ведь он находится с той стороны, с которой находятся синие объекты, а относительно других прямых он будет отнесен к классу «$+1$». Иллюстрации возможны, как мы уже отмечали, не всегда, а потому давайте проведем и аналитические выкладки. Для голубой прямой

$$f_{\text{голубая}}(2, 3) = -8 - 3 + 10 = -1 < 0$$

получаем отрицательное значение, значит назначаемый класс – это класс «-1». Для других прямых

$$f_{\text{фиолетовая}}(2, 3) = -2 + 2.25 = 0.25 > 0,$$

$$f_{\text{зеленая}}(X_1, X_2) = -2.667 \cdot 2 - 3 + 8.999 = 0.665 > 0$$

объект относится, как мы и предсказывали, к классу «+1».

В окончание этого пункта полезно сделать следующие выводы, обобщающие сказанное выше:

1. Гиперплоскость

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p$$

   разделяет рассматриваемое p-мерное пространство на 2 части.

2. Точки одной части пространства принадлежат одному классу, а другой – другому.

3. Нормаль гиперплоскости $n = (\theta_1, \theta_2, ..., \theta_p)$ направлена в сторону той части пространства, на точках которой классификатор дает положительные значения. Эту часть пространства часто называют классом «+1», другую же – классом «−1».

4. Сам классификатор задается следующим аналитическим выражением

$$f(X) = f(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p.$$

5. Для классификации тестового данного x^*, имеющего значения предикторов $(x_1^*, x_2^*, ..., x_p^*)$, вычисляется значение f:

$$f(x^*) = \theta_0 + \theta_1 x_1^* + \theta_2 x_2^* + ... + \theta_p x_p^*.$$

   Если значение $f(x^*)$ положительно, то объект x^* относят к классу «+1», если отрицательно – к классу «−1», а если равно нулю, то объект лежит на гиперплоскости и может быть отнесен к любому классу.

6. Ну и последнее. Чем больше для тестового объекта x^* значение $|f(x^*)|$, тем дальше объект расположен от разделяющей гиперплоскости, и тем «увереннее» классификация, то есть тем меньше шанс ошибиться и присвоить объекту неправильный класс.

Теперь мы готовы перейти к последнему поднятому, но еще пока нерешенному вопросу: а как же построить ту самую оптимальную гиперплоскость, про которую мы говорили в самом начале? Давайте разбираться.

1.4 Математическое построение оптимальной разделяющей гиперплоскости

Предположим, что нам дано n тренировочных данных – суть n откликов Y от p предикторов $X_1, X_2, ..., X_p$, а также предположим, что данные линейно разделимы. Каждый тренировочный объект x_i, $i \in \{1, 2, ..., n\}$ можно отождествить с $(p+1)$-им числом: это набор

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\}$$

из p предикторов и отклик y_i, причем последний равен либо $+1$, либо -1, в зависимости от того, к какому классу принадлежит тренировочный объект. Будем искать разделяющую гиперплоскость l в виде

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0$$

при дополнительном условии, что

$$\theta_1^2 + \theta_2^2 + ... + \theta_p^2 = \sum_{i=1}^{p} \theta_i^2 = 1.$$

Последнее уловие, по сути дела, – это требование, чтобы длина $|n|$ нормали плоскости $n = (\theta_1, \theta_2, ..., \theta_p)$ была равна единице. Этого всегда можно добиться для любой гиперплоскости домножением всего равенства

$$\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0$$

на $\left(\theta_1^2 + \theta_2^2 + ... + \theta_p^2\right)^{-1/2} = |n|^{-1/2}$, так что вводимое ограничение нисколько не нарушает общности. Чтобы не вводить дополнительных обозначений, будем впредь предполагать, что коэффициенты уравнения гиперплоскости удовлетворяют приведенному требованию. Напомним следующую теорему, известную из курса аналитической геометрии.

Теорема 1.4.1 *Пусть гиперплоскость l задана уравнением*

$$l: \ \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

причем $|n| = 1$, $n = (\theta_1, \theta_2, ..., \theta_p)$. Тогда расстояние $\rho(X, l)$ от произвольной точки $X = (X_1, X_2, ..., X_p)$ до гиперплоскости l может быть вычислено, как

$$\rho(X, l) = |\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p|.$$

Замечание 1.4.1 *Полезно заметить, что последнее выражение – это модуль значения классификатора $f(X)$, построенного по гиперплоскости l:*

$$f(X) = f(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p,$$

вычисленного на объекте X.

Мы хотим построить такую гиперплоскость, нормаль которой направлена в сторону той части пространства, в которой расположены объекты класса «+1» (у который отклик +1). Тогда на тренировочных данных выражение для расстояния до гиперплоскости может быть записано следующим образом:

$$\rho(x_i, l) = y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right), \quad i \in \{1, 2, ..., n\}.$$

Теперь давайте запишем наши три сформулированные ранее требования к гиперплоскости, но математическим языком. Итак, мы хотим, чтобы каждое тренировочное данное было правильно классифицировано, и расстояние от него до гиперплоскости было как можно больше. Это означает, что мы ищем такое наибольшее неотрицательное M, для которого расстояние от каждого тренировочного данного до гиперплоскости удовлетворяет неравенству

$$\rho(x_i, l) = y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right) \geq M,$$

варьируя (или изменяя) параметры $\theta_0, \theta_1, ..., \theta_p$. Если говорить точнее, то мы ищем, конечно, гиперплоскость, а значит как раз-таки коэффициенты, но исходя из условия, что максимизируется число M.

Итого, все условия могут быть записаны следующим образом:

$$\max_{\theta_0, \theta_1, ..., \theta_p} M,$$

$$y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right) \geq M,$$

$$\theta_1^2 + \theta_2^2 + ... + \theta_p^2 = \sum_{i=1}^{p} \theta_i^2 = 1.$$

Эта оптимизационная задача решается численно, а мы приходим к следующему определению.

Определение 1.4.1 *Гиперплоскость l, уравнение которой задается соотношением*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

в котором коэффициенты $\theta_0, \theta_1, ..., \theta_p$ определяются из оптимизационной задачи, описанной выше, называется оптимальной разделяющей гиперплоскостью (optimal margin hyperplane).

Для дальнейшего изучения описанного метода, перепишем задачу в менее очевидных, но более удобных обозначениях. Пусть $w = (\theta_1, \theta_2, ..., \theta_p)$ – вектор нормали гиперплоскости, $w_0 = -\theta_0$, $X = (X_1, X_2, ..., X_p)$. Уравнение гиперплоскости в таком случае перепишется в виде

$$(w, X) - w_0 = 0.$$

Для получения более удобной оптимизационной задачи, поделим на M неравенство

$$y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right) \geq M,$$

снимем ограничение, что $|w| = 1$ и учитывая, что мы максимизируем M, и что умножение на положительную константу не влияет на минимизацию, придем к следующей постановке задачи:

$$\begin{cases} \frac{1}{2}(w, w) \to \min\limits_{w, w_0} \\ y_i((w, x_i) - w_0) \geq 1, \quad i \in \{1, 2, ..., n\} \end{cases}$$

Итак, мы минимизируем половину квадрата длины вектора нормали, изменяя нормаль гиперплоскости и свободный член, или, что то же самое, параметры гиперплоскости $\theta_0, \theta_1, ..., \theta_p$, но таким образом, что выполняется n условий вида

$$y_i((w, x_i) - w_0) \geq 1, \quad i \in \{1, 2, ..., n\},$$

каждое из которых означает, что все тренировочные данные находятся вне (или на границе) разделяющей полосы.

К такой постановке задачи можно было прийти и сразу, исходя из геометрических соображений, которые озвучивались и использовались ранее, приведем их снова. Второе условие

$$y_i((w, x_i) - w_0) \geq 1 \Leftrightarrow -1 \leq (w, x_i) - w_0 \leq 1$$

определяет полосу, которая разделяет два класса.

На рисунке 6 таких полосы две. Никакая точка тренировочных данных не может попасть внутрь рассматриваемой полосы, полоса ограничена двумя гиперплоскостями с нормалью w. Точки, ближайшие к гиперплоскости, находятся на границе полосы (их две в случае фиолетовой полосы и три в случае зеленой), а сама разделяющая гиперплоскость делит полосу ровно пополам. Совершенно очевидно, что чтобы разделяющая гиперплокость как можно дальше отстояла от объектов каждого класса, ширина полосы должна быть как можно больше. Давайте вычислим ширину. Пусть x_- и x_+ – это два таких представителя классов «+1» и «−1», которые лежат на границе разделяющей полосы. Тогда ширина полосы (рисунок 7) может быть найдена,

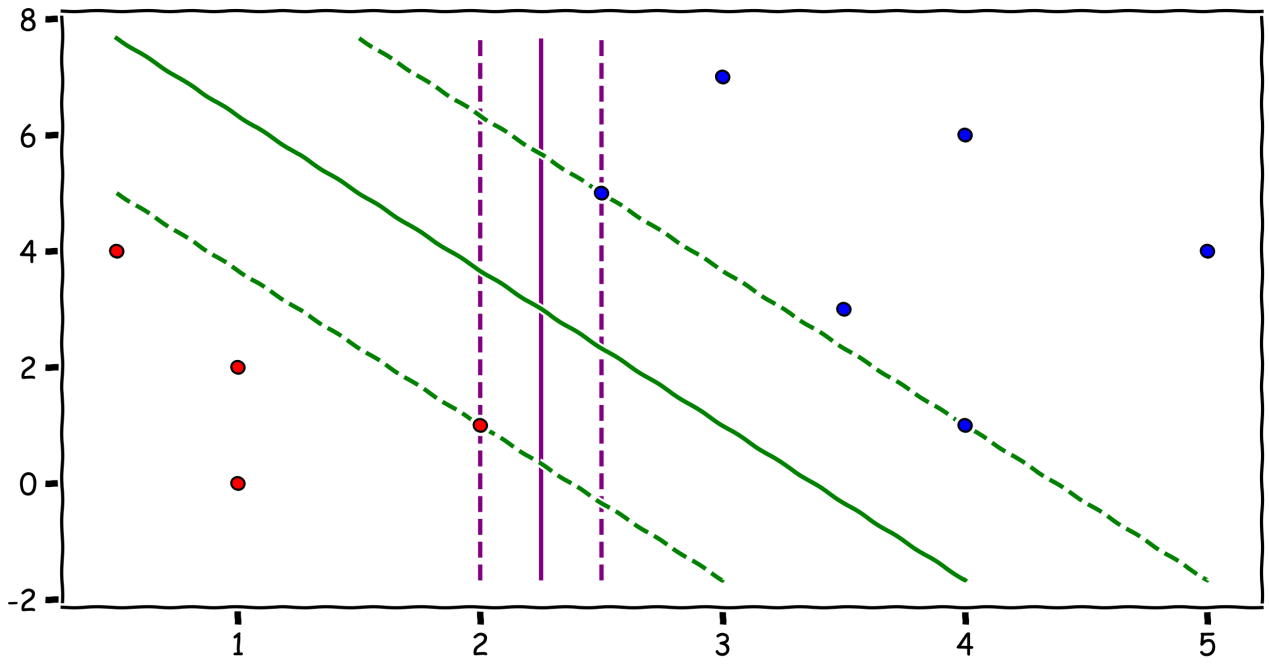


Рис. 6: Две разделяющие полосы

как

$$\left(x_+ - x_-, \frac{w}{|w|}\right) = \frac{(x_+, w) - (x_-, w)}{|w|} = \frac{(w_0 + 1) - (w_0 - 1)}{|w|} = \frac{2}{|w|}.$$

А значит, ширина полосы максимальная в случае, когда длина $|w|$ нормали w минимальна (или половина квадрата длины минимальна), откуда и получается ранее озвученная задача:

$$\begin{cases} \frac{1}{2}(w, w) \to \min\limits_{w, w_0} \\ y_i((w, x_i) - w_0) \geq 1, \quad i \in \{1, 2, ..., n\} \end{cases}.$$

Максимальность ширины разделяющей полосы, что мы не раз уже отмечали, как мы надеемся, обеспечивает нам наиболее точное разделение тренировочных данных на два класса, так как в этом случае данные расположены максимально далеко (насколько это возможно при линейном разделении) друг от друга.

1.5 Опорные векторы и условия Куна-Таккера

По сути дела, описанная выше задача – это задача поиска условного экстремума, ведь нужно найти минимум некоторой функции, но с учетом каких-то ограничений (условий). Для решения такого типа задач можно воспользоваться так называемыми условиями Куна-Таккера (обобщение стандартного метода множителей Лагранжа в задачах поиска условного экстремума). Опи-

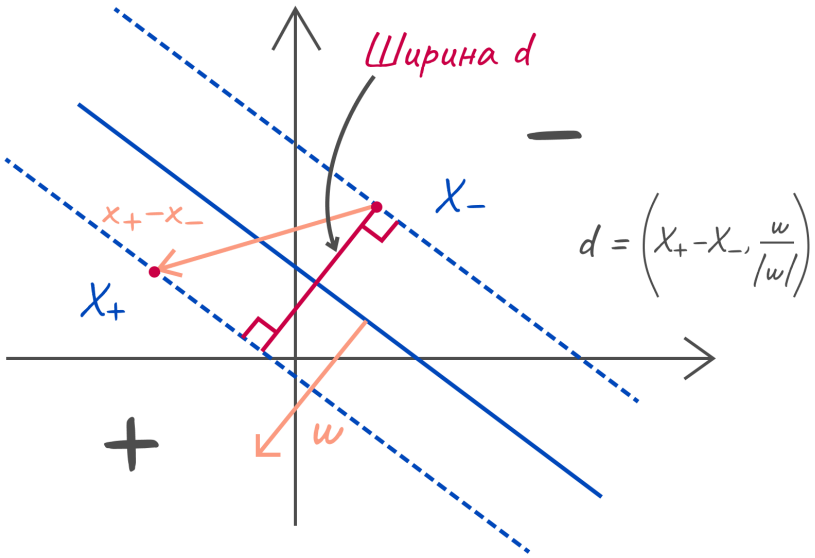


Рис. 7: Ширина полосы

санная задача, используя теорему Куна-Таккера, может быть переформулирована в задачу поиска седловой точки так называемой функции Лагранжа:

$$\begin{cases} L(w, w_0, \lambda) = \frac{1}{2}(w, w) - \sum\limits_{i=1}^{n} \lambda_i \left(y_i((w, x_i) - w_0) - 1\right) \to \min\limits_{w, w_0} \max\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1, 2, ..., n\} \\ \lambda_i = 0 \text{ или } (w, x_i) - w_0 = y_i, \quad i \in \{1, 2, ..., n\} \end{cases}.$$

Первое выражение $L(w, w_0, \lambda)$ в написанной системе часто называют функцией Лагранжа, вторая и третья же строчки устанавливают условия на аргументы функции Лагранжа. Отдельно стоит отметить третье условие: оно говорит, что слагаемое, отвечающее тренировочному данному x_i в функции Лагранжа не равно нулю только тогда, когда $(w, x_i) - w_0 = y_i$, то есть когда объект x_i лежит на границе разделяющей полосы!

Необходимым условием для седловой точки функции Лагранжа является, как обычно, равенство нулю частных производных, откуда получаются следующие соотношения:

$$\frac{\partial}{\partial w} L(w, w_0, \lambda) = w - \sum_{i=1}^{n} \lambda_i y_i x_i = 0,$$

$$\frac{\partial}{\partial w_0} L(w, w_0, \lambda) = \sum_{i=1}^{n} \lambda_i y_i = 0.$$

Из первого соотношения получаем, что

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i,$$

то есть искомый вектор w является линейной комбинацией тренировочных данных x_i, причем только тех, которые лежат на границе разделяющей полосы (иначе коэффициенты λ_i равны нулю).

Определение 1.5.1 *Тренировочные данные, для которых выполняется условие*

$$(w, x_i) - w_0 = y_i,$$

называются опорными векторами (support vector).

Итак, по большому счету, для построения разделяющей гиперплоскости нам нужны только опорные векторы. К сожалению, какие векторы опорные, а какие – нет, понять сходу удается не всегда, поэтому тренировочные данные прорядить «сильно» удается не всегда.

Вернемся к нашей задаче и совершим последний рывок, подставив полученные соотношения

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i, \quad \sum_{i=1}^{n} \lambda_i y_i = 0$$

в функцию Лагранжа. Сначала рассмотрим первое слагаемое, оно преобразуется следующим образом:

$$\frac{1}{2}(w, w) = \frac{1}{2}\left(\sum_{i=1}^{n} \lambda_i y_i x_i,\ \sum_{j=1}^{n} \lambda_j y_j x_j\right) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_y y_j (x_i, x_j).$$

Теперь рассмотрим второе слагаемое, оно преобразуется так:

$$\sum_{i=1}^{n} \lambda_i (y_i((w, x_i) - w_0) - 1) = \sum_{i=1}^{n} \lambda_i y_i (w, x_i) - w_0 \sum_{i=1}^{n} \lambda_i y_i - \sum_{i=1}^{n} \lambda_i.$$

Второе слагаемое равно нулю за счет условия, что $\sum\limits_{i=1}^{n} \lambda_i y_i = 0$. Подставим в первое слагаемое выражение для w, получим:

$$\sum_{i=1}^{n} \lambda_i (y_i((w, x_i) - w_0) - 1) = \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i, x_j) - \sum_{i=1}^{n} \lambda_i.$$

Итого, функция Лагранжа преобразуется к виду

$$L(\lambda) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j (x_i, x_j) + \sum_{i=1}^{n}\lambda_i.$$

В конечном итоге, мы приходим к эквивалентной задаче, которая записывается в следующем виде:

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n}\lambda_i + \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\lambda_i\lambda_j y_i y_j (x_i, x_j) \to \min\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n}\lambda_i y_i = 0 \end{cases},$$

где ищется не максимум функции $L(\lambda)$, а минимум функции $-L(\lambda)$, что, конечно же, то же самое. Задача, записанная в таком виде, намного лучше предыдущей. Здесь минимизируется квадратичный функционал с неотрицательно определенной квадратичной формой, откуда следует, что он выпуклый. Область ограничений же, определямая неравенствами и одним равенством, тоже выпукла, а значит поставленная задача имеет и при том единственное решение!

Решив полученную задачу, для построения разделяющей гиперплоскости можно действовать по следующему алгоритму:

1. Определим w из соотношения

$$w = \sum_{i=1}^{n}\lambda_i y_i x_i.$$

2. Найдем w_0. Для этого достаточно взять произвольный опорный вектор x и из соотношения $(w, x_i) - w_0 = y_i$ найти

$$w_0 = (w, x_i) - y_i.$$

3. Классификатор зададим аналитическим выражением

$$f(X) = (w, X) - w_0.$$

Дальнейшая классификация проводится по общим правилам, описанным в конце пункта 1.3.

Определение 1.5.2 *Построенный таким образом классификатор часто называется классификатором с оптимальным зазором (optimal margin classifier) или классификатором с жестким зазором.*

Возвращаясь к нашему примеру, оказывается, что построенная ранее зеленая прямая – и есть оптимальная разделяющая гиперплоскость. Мы видим, что 3 тренировочных данных являются опорными векторами – это представитель красных с координатами $(2, 1)$ и представители синих с координатами $(4, 1)$, $(2.5, 5)$. Гиперплоскость задается уравнением (с примерными коэффи-

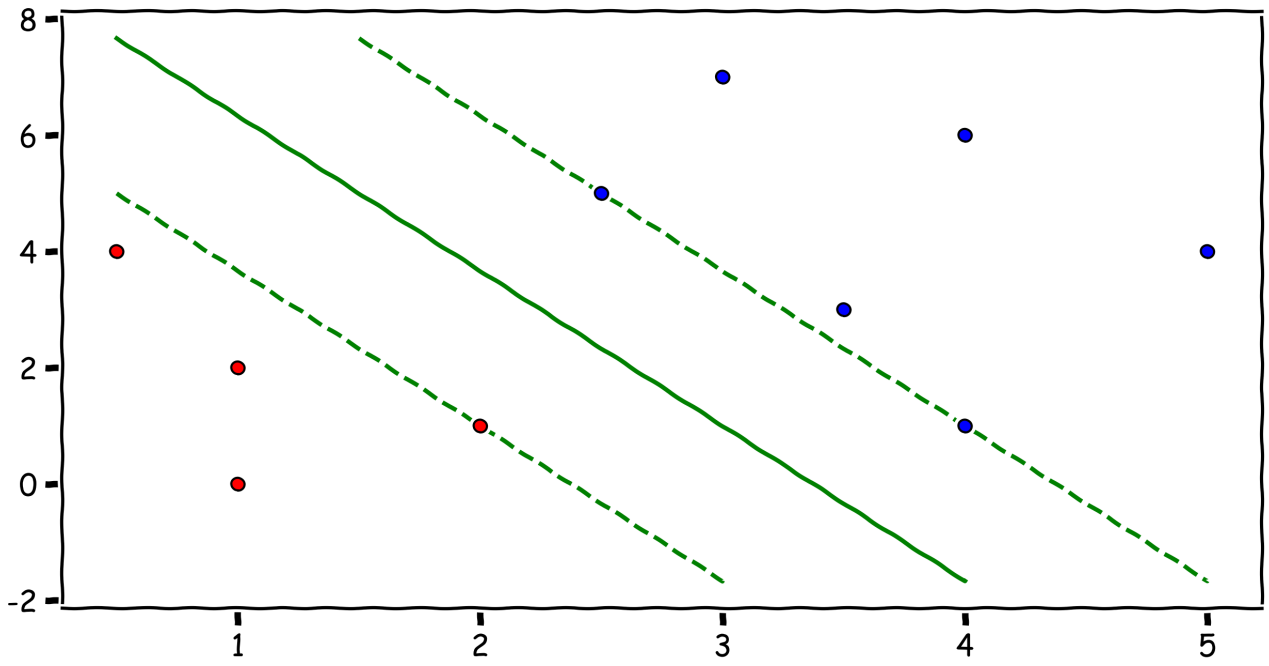


Рис. 8: Оптимальная разделяющая гиперплоскость

циентами)

$$-X_1 - 0.375X_2 + 3.375 = 0,$$

а классификатор выражением

$$f(X_1, X_2) = -X_1 - 0.375X_2 + 3.375.$$

Выполним классификацию двух новых объектов с координатами $(4, 0)$ – оранжевый и $(1.5, 1)$ – фиолетовый. Геометрически понятно, что оранжевого следует отнести к классу синих («-1»), а фиолетового к классу красных («$+1$»). Кроме того, классификация оранжевого объекта не является очень «уверенной», ведь объект попал внутрь разделяющей полосы.

Аналитически наши выводы тоже подкрепляются, ведь для оранжевого объекта

$$f(4, 0) = -4 + 3.375 = -0.625 < 0 \Rightarrow \text{класс } « - 1»,$$

а для фиолетового

$$f(1.5, 1) = -1.5 - 0.375 + 3.375 = 1.5 > 0 \Rightarrow \text{класс } « + 1».$$

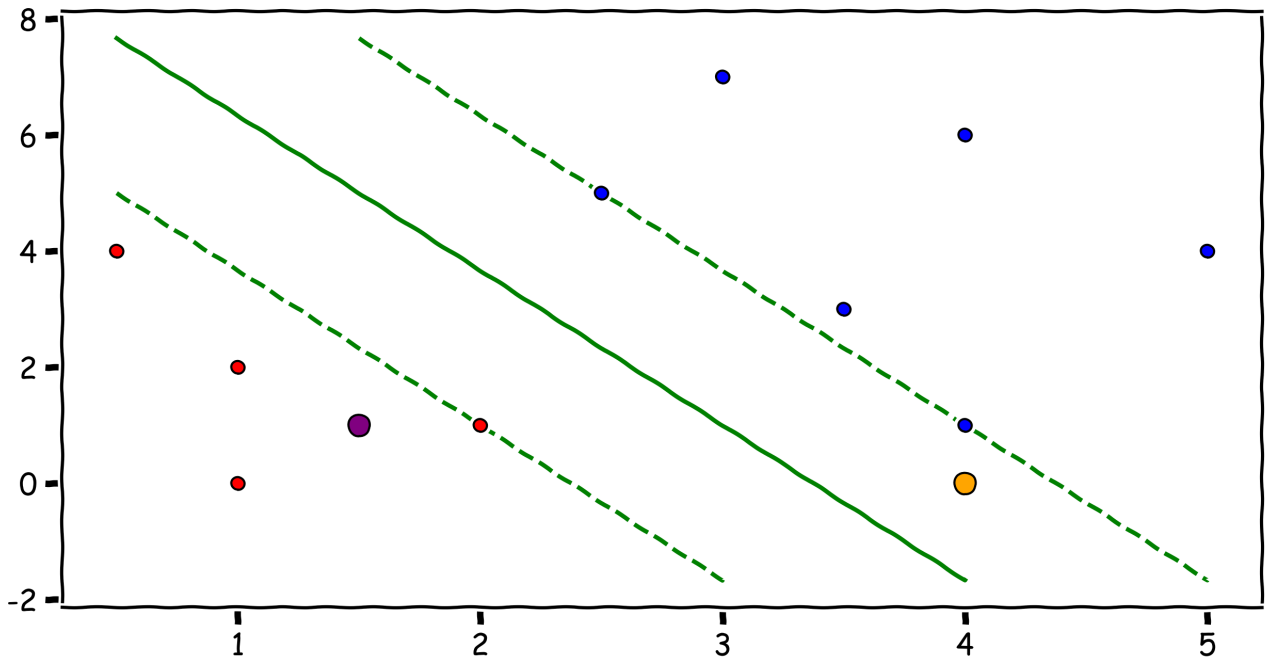


Рис. 9: Классификация новых объектов

Итак, мы научились проводить классификацию в случае линейно разделимых тренировочных данных, но у такого подхода есть совершенно очевидные недостатки: во-первых, описанный подход работает лишь в том случае, когда данные могут быть разделены гиперплоскостью, а во-вторых – такой подход очень чувствителен к выбросам. О том, как решить эти вопросы, мы расскажем дальше, а сейчас рассмотрим некоторый вычислительный пример, чтобы закрепить сказанное.

1.6 Пример подробного расчета «на пальцах»

Для того, чтобы формулы не казались столь пугающими, проведем разбор чрезвычайно синтетического примера «на пальцах», аналитически. Пусть класс «-1» (синий) содержит точки $x_1 = (0, 0)$, $x_2 = (1, 0)$, а класс «$+1$» (красный) содержит точки $x_3 = (2, 0)$ и $x_4 = (0, 2)$. Для удобства представим данные следующей таблицей:

Объект	X_1	X_2	Отклик
x_1	0	0	-1
x_2	1	0	-1
x_3	2	0	$+1$
x_4	0	2	$+1$

и изобрасим на плоскости на рисунке 10. Попробуйте догадаться просто из наглядного представления данных, какие из них будут опорными векторами, и где пройдет прямая. Наверное, нетрудно понять, что опорными векторами,

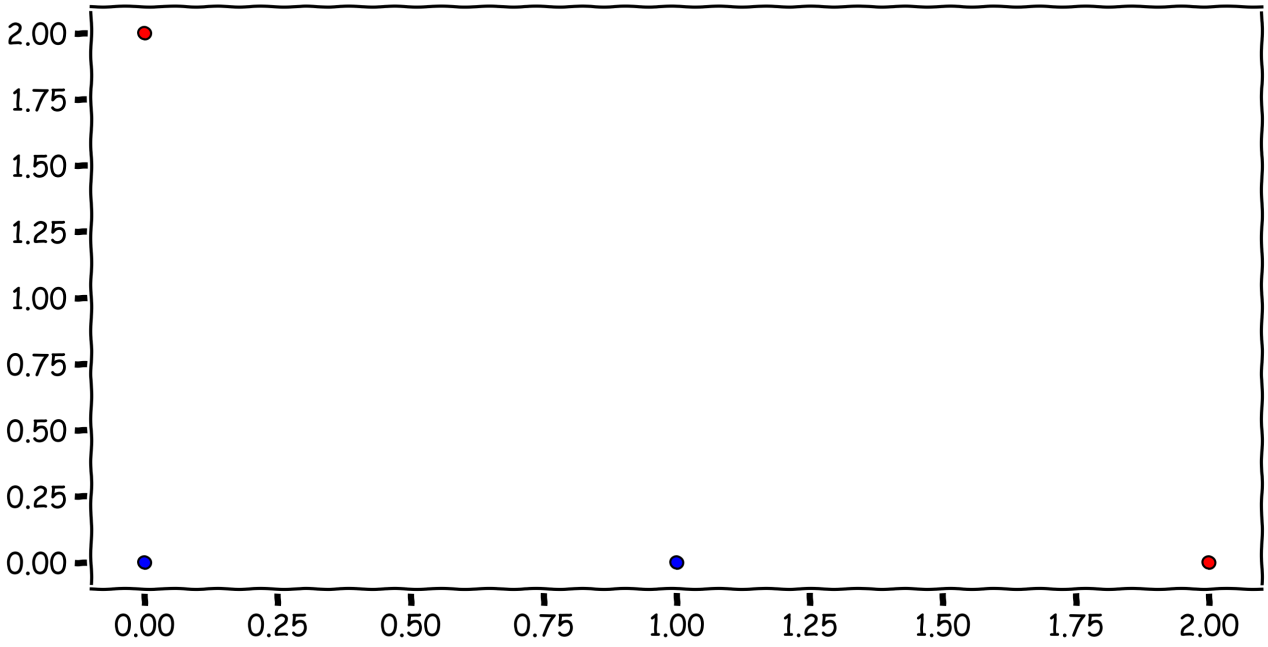


Рис. 10: Тренировочные данные

скорее всего, будут x_2, x_3, x_4, но давайте это проверим!

Итак, во веденных обозначениях,

$$x_1 = (0,0),\ x_{11} = 0,\ x_{12} = 0,$$

$$x_2 = (1,0),\ x_{21} = 1,\ x_{22} = 0,$$
$$x_3 = (2,0),\ x_{31} = 2,\ x_{32} = 0,$$
$$x_4 = (0,2),\ x_{41} = 0,\ x_{42} = 2$$

и скалярное произведение элемента x_i на элемент x_j вычисляется, как

$$(x_i, x_j) = x_{i1}x_{j1} + x_{i2}x_{j2},\ i,j \in \{1,2,3,4\}.$$

Так как имеется четыре тренировочных объекта, то есть $n = 4$, причем

$$(x_1, x_1) = (x_1, x_2) = (x_1, x_3) = (x_1, x_4) = 0,$$

$$(x_2, x_2) = 1,\ (x_2, x_3) = 2,\ (x_2, x_4) = 0,$$
$$(x_3, x_3) = 4,\ (x_3, x_4) = 0,$$
$$(x_4, x_4) = 4,$$

то в нашем случае минимизируемое выражение

$$-L(\lambda) = -\sum_{i=1}^{n} \lambda_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j y_i y_j (x_i, x_j)$$

примет следующий вид:

$$-L(\lambda) = -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 + \frac{1}{2}\left(\lambda_2^2 + 4\lambda_3^2 + 4\lambda_4^2 - 4\lambda_2\lambda_3\right)$$

и, кроме условий $\lambda_i \geq 0$, $i \in \{1, 2, 3, 4\}$, в силу последнего условия оптимизационной задачи

$$\sum_{i=1}^{n} \lambda_i y_i = 0,$$

возникает еще одно условие

$$\lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 = 0.$$

Итого, мы приходим к следующей задаче:

$$\begin{cases} -L(\lambda) = -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 + \frac{1}{2}\left(\lambda_2^2 + 4\lambda_3^2 + 4\lambda_4^2 - 4\lambda_2\lambda_3\right) \to \min\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1, 2, 3, 4\} \\ \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 = 0 \end{cases}.$$

Полученная задача – это стандартная задача дифференциального исчисления функций многих переменных: поиск наименьшего значения заданной функции в заданной области при некоторых ограничениях (условиях). У непрерывной функции это значение может достигаться либо во внутренней точке области, либо на ее границе.

Необходимым условием локального экстремума во внутренней точке области является равенство нулю частных производных, откуда приходим к следующей системе:

$$\begin{cases} -\frac{\partial L(\lambda)}{\partial \lambda_1} = 0 \\ -\frac{\partial L(\lambda)}{\partial \lambda_2} = 0 \\ -\frac{\partial L(\lambda)}{\partial \lambda_3} = 0 \\ -\frac{\partial L(\lambda)}{\partial \lambda_4} = 0 \end{cases} \Leftrightarrow \begin{cases} -1 = 0 \\ -1 + \lambda_2 - 2\lambda_3 = 0 \\ -1 + 4\lambda_3 - 2\lambda_2 = 0 \\ -1 + 4\lambda_4 = 0. \end{cases}$$

Ясно, что уже первое уравнение системы не имеет ни одного решения, а значит и вся система решений не имеет. Тем самым, наименьшее значение функции стоит искать на границе области.

Пусть $\lambda_1 = 0$, тогда третье уравнение перепишется в виде $\lambda_2 = \lambda_3 + \lambda_4$ и функция, которую мы исследуем на наименьшее значение, перепишется в виде

$$-L(\lambda_3, \lambda_4) = -2\lambda_3 - 2\lambda_4 + \frac{1}{2}\left((\lambda_3 + \lambda_4)^2 + 4\lambda_3^2 + 4\lambda_4^2 - 4(\lambda_3 + \lambda_4)\lambda_3\right),$$

а область, в которой ищется наименьшее значение, будет задаваться соотношениями $\lambda_3, \lambda_4 \geq 0$.

Вычислив частные производные по λ_3 и λ_4, получим систему

$$\begin{cases} -2 + \lambda_3 + \lambda_4 + 4\lambda_3 - 4\lambda_3 - 2\lambda_4 = 0 \\ -2 + \lambda_3 + \lambda_4 + 4\lambda_4 - 2\lambda_3 = 0 \end{cases} \Leftrightarrow \begin{cases} \lambda_3 - \lambda_4 = 2 \\ -\lambda_3 + 5\lambda_4 = 2 \end{cases},$$

откуда $\lambda_3 = 3$, $\lambda_4 = 1$, $\lambda_2 = 4$ и $\lambda_1 = 0$, а $-L(3,1) = -4$. Можно показать, что это и будет наименьшее значение рассматриваемой функции в заданной области.

Проверяя аналогичным образом остальные границы получим, что на них у рассматриваемой функции наименьшее значение больше, чем -4, а значит, как мы и предполагали ранее, элементы x_2, x_3, x_4 являются опорными векторами, ведь именно отвечающие им коэффициенты λ отличны от нуля.

Для того чтобы построить оптимальную разделяющую гиперплоскость, найдем для начала вектор нормали w:

$$w = \sum_{i=1}^{4} \lambda_i y_i x_i = 0 \cdot (-1) \cdot x_1 + 4 \cdot (-1) \cdot x_2 + 3 \cdot 1 \cdot x_3 + 1 \cdot 1 \cdot x_4 =$$

$$= -(0,0) - 4(1,0) + 3(2,0) + (0,2) = (0,0) + (4,0) + (6,0) + (0,2) = (2,2).$$

Осталось вычислить w_0. Для этого возьмем какой-нибудь опорный вектор, например $x_2 = (1,0)$, и вычислим

$$w_0 = (w, x_2) - y_2 = 2 - (-1) = 3.$$

Итого, оптимальная разделяющая гиперплоскость определяется уравнением

$$2X_1 + 2X_2 - 3 = 0,$$

а классификатор задается выражением

$$f(X) = f(X_1, X_2) = 2X_1 + 2X_2 - 3.$$

На рисунке 11 построена разделяющая гиперплоскость, исходные данные, а также разделяющая полоса. Все наши ожидания оправдались. Пусть теперь мы хотим понять, к какому классу относится тестовое наблюдение $x^* = (1,2)$. Для этого вычисляем $f(x^*)$ и получаем

$$f(x^*) = f(1,2) = 2 + 4 - 3 = 3 > 0,$$

откуда тестовое наблюдение относится к классу «+1», то есть к красным. Подтверждение этому факту можно увидеть на рисунке 12, где точка, отвечающая тестовому данному, окрашена в оранжевый цвет, и располагается в той же полуплоскости, что и красные.

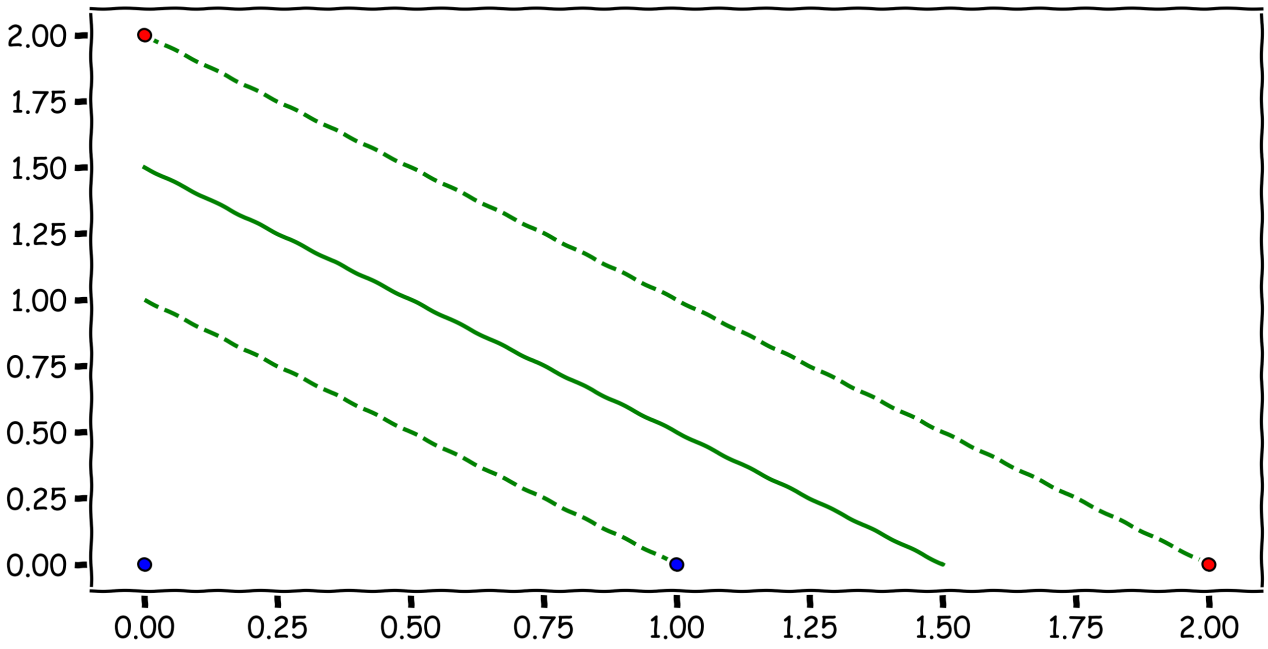


Рис. 11: Оптимальная разделяющая гиперплоскость

2 Метод опорных векторов (SVM): линейно неразделимая выборка

2.1 Гиперплоскость в случае, когда данные линейно неразделимы

Классификатор, который мы строили до настоящего момента, отлично работает в случае, когда данные могут быть разделены гиперплоскостью. Такое бывает не всегда. Обратите внимание, что изменение класса лишь одного из тренировочных данных может полностью нарушить линейную разделимость. При этом, что весьма обидно, легко может оказаться, что это выбивающееся наблюдение просто-напросто ошибочно, то есть является выбросом, рисунок 13. Кроме того, иногда, может быть, можно пожертвовать ошибками на некоторых тренировочных данных, если профит (скажем, ширина разделяющей полосы), становится куда больше?

Оказывается, подход, развитый нами в предыдущих пунктах, можно расширить: можно строить оптимальную почти-разделяющую гиперплоскость (или гиперплоскость с мягким зазором), которая хоть и допускает ошибки в классификации даже на тренировочных данных (еще раз, так как сами по себе данные просто не могут быть линейно разделены), но все же поддерживает «почти-линейное» разделение. Конечно, так мы не решим всех проблем, ведь данные могут быть даже близко линейно неразделимыми (рисунок 14), но об этом мы поговорим еще чуть позже. Итак, в этом пункте

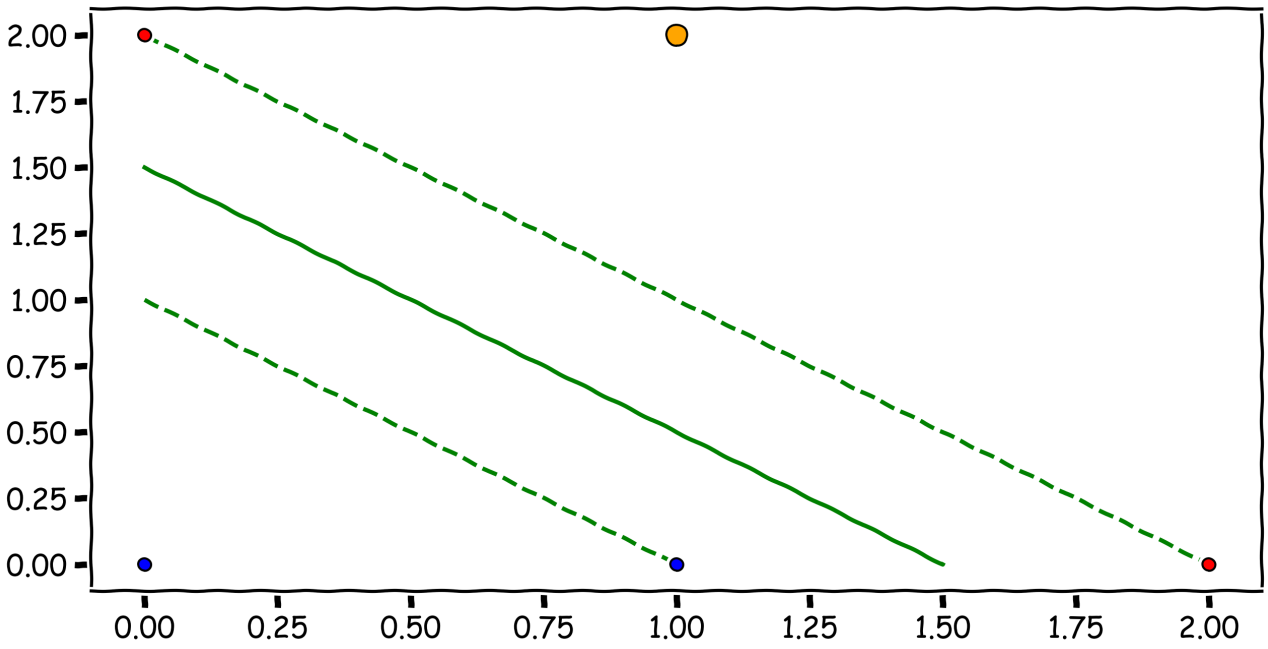


Рис. 12: Классификация тестового объекта

мы зададимся целью построить классификатор, основанный на построении почти-разделяющей гиперплоскости, хоть и не идеально разделяющей данные, но в то же время обладающей следующими приятными и полезными бонусами:

1. Большая устойчивость к конкретным наблюдениям, а значит и к выбросам.

2. Лучшая классификация «большинства» тренировочных наблюдений.

3. Легкая интерпретация правила разделения (классификатора).

В результате наш классификатор, конечно, будет допускать ошибки на тренировочных данных, но не «слишком много». В то же время, он должен будет работать лучше при классификации оставшихся наблюдений.

Введем дополнительный набор переменных $\xi_i \geq 0$, $i \in \{1, 2, ..., n\}$, каждая из которых характеризует величину ошибки на тренировочном данном x_i. В поставленной ранее задаче ослабим ограничения и придем к следующей оптимизационной задаче:

$$\begin{cases} \frac{1}{2}(w, w) + C \sum\limits_{i=1}^{n} \xi_i \to \min\limits_{w, w_o, \xi} \\ y_i((w, x_i) - w_0) \geq 1 - \xi_i, \ \ i \in \{1, 2, ..., n\} \\ \xi_i \geq 0, \ \ i \in \{1, 2, ..., n\} \end{cases},$$

где C – некоторый положительный параметр.

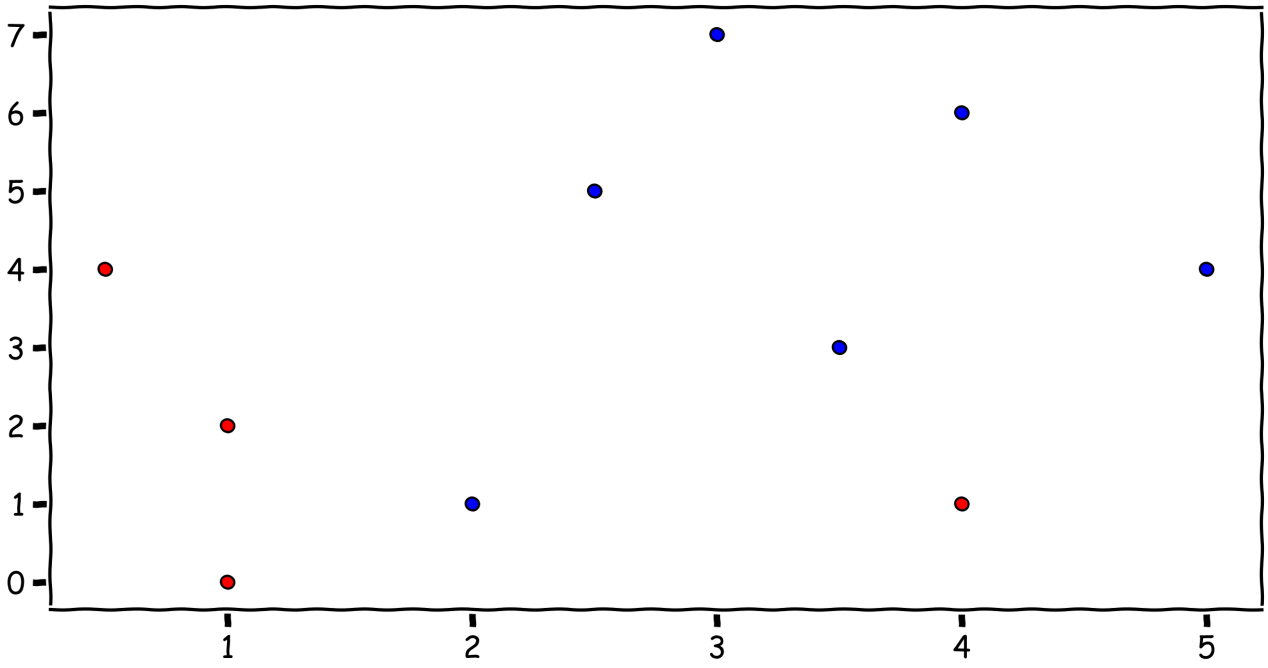


Рис. 13: Пример линейно неразделимых данных

Давайте посмотрим на второе условие и поймем, что оно означает. На самом деле оно как раз-таки формально описывает положение каждого тренировочного данного x_i относительно почти-разделяющей гиперплоскости. Итак, для тренировочного данного x_i возможно 3 различных ситуации:

1. $\xi_i = 0$, то есть $y_i((w, x_i) - w_0) \geq 1$. В этом случае классификатор корректно классифицирует наблюдение x_i, оно лежит вне разделяющей полосы. В случае, если в неравенстве достигается равенство, то есть если $y_i((w, x_i) - w_0) = 1$, то объект лежит на границе разделяющей полосы.

2. $0 < \xi_i \leq 1$, то есть $0 \leq y_i((w, x_i) - w_0) < 1$. В этом случае наблюдение x_i все так же корректно классифицировано, однако находится либо внутри разделительной полосы, если левое неравенство строгое, либо лежит на разделяющей гиперплоскости, если $y_i((w, x_i) - w_0) = 0$

3. $\xi_i > 1$, то есть $y_i((w, x_i) - w_0) < 0$. В этом случае наблюдение x_i классифицировано неверно.

Смотрите, первое условие оптимизационной задачи все так же стремится максимизировать ширину разделяющей полосы (путем минимизации (w, w)), но, к тому же, и сумму ошибок ξ_i с некоторым положительным весом C. Параметр C является управляющим параметром модели, отдается на откуп исследователю и контролирует баланс между максимизацией разделяющей полосы и суммарной ошибкой в классификации тренировочных данных.

По аналогии с тем, как было сделано ранее, введем определение.

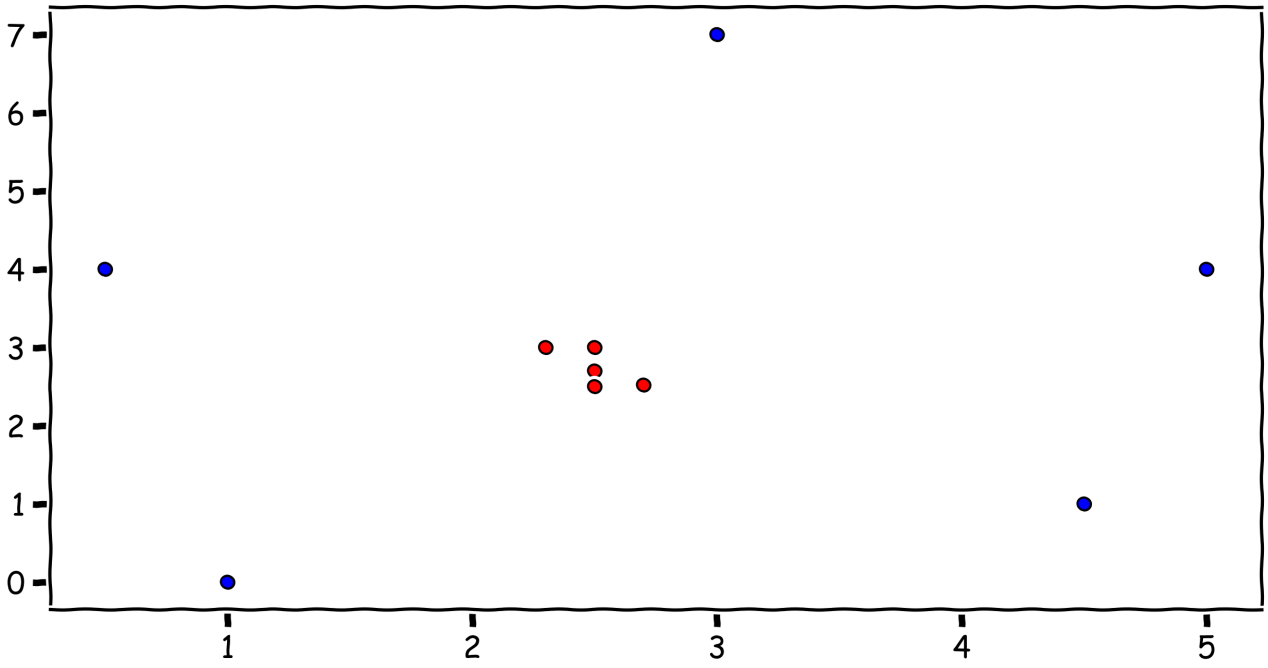


Рис. 14: Пример совсем линейно неразделимых данных

Определение 2.1.1 *Гиперплоскость l, уравнение которой задается соотношением*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

в котором коэффициенты $\theta_0, \theta_1, ..., \theta_p$ определяются из оптимизационной задачи, описанной выше, называется разделяющей гиперплоскостью с мягким зазором (soft margin hyperplane) или почти-разделяющей гиперплоскостью.

Как и ранее, сведем описанную задачу к задаче поиска седловой точки функции Лагранжа. Функция Лагранжа в нашем случае перепишется в виде

$$L(w, w_0, \xi, \lambda, \eta) = \frac{1}{2}(w, w) - \sum_{i=1}^{n} \lambda_i (y_i((w, x_i) - w_0) - 1) - \sum_{i=1}^{n} \xi_i (\lambda_i + \eta_i - C),$$

где η_i, не вдаваясь в детали, так называемая двойственная переменная к ξ_i. Итого, используя условия Куна-Таккера, мы переходим к задаче:

$$\begin{cases} L(w, w_0, \xi, \lambda, \eta) \to \min\limits_{w, w_0, \xi} \max\limits_{\lambda, \eta} \\ \xi_i, \eta_i, \lambda_i \geq 0, \ i \in \{1, 2, ..., n\} \\ \lambda_i = 0 \text{ или } y_i(w, x_i) - w_0) = 1 - \xi_i, \ i \in \{1, 2, ..., n\} \\ \eta_i = 0 \text{ или } \xi_i = 0, \ i \in \{1, 2, ..., n\} \end{cases}.$$

Дальнейшие шаги практически полностью повторяют шаги, проделанные нами ранее. Необходимое условие седловой точки – равенство нулю частных

производных функции Лагранжа, откуда

$$\frac{\partial}{\partial w}L(w, w_0, \xi, \lambda, \eta) = w - \sum_{i=1}^{n} \lambda_i y_i x_i = 0,$$

$$\frac{\partial}{\partial w_0}L(w, w_0, \xi, \lambda, \eta) = \sum_{i=1}^{n} \lambda_i y_i = 0,$$

$$\frac{\partial}{\partial \xi_i}L(w, w_0, \xi, \lambda, \eta) = -\lambda_i - \eta_i + C = 0.$$

Из этих равенств получаем важные соотношения, часть из которых (а точнее – первые два) нам уже известна:

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i,$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0,$$

$$\lambda_i + \eta_i = C,\ i \in \{1, 2, ..., n\}.$$

Подставив это, как и ранее, в функцию Лагранжа, приходим к задаче поиска минимума следующего вида:

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n} \lambda_i + \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i, x_j) \to \min\limits_{\lambda} \\ 0 \le \lambda_i \le C,\ i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n} \lambda_i y_i = 0 \end{cases}.$$

Понятно, что ранее решаемая нами задача отличается от данной лишь второй строчкой с дополнительным ограничением, что $\lambda_i \le C$ при $i \in \{1, 2, ..., n\}$. По тем же причинам, что были озвучены ранее, данная задача имеет, и при том единственное решение.

Решив поставленную задачу, то есть найдя параметры λ_i, следует, как и ранее, построить классификатор. Но для этого полезно бы сделать выводы о том, какие объекты и как классифицированы, а также ввести необходимую терминологию. Не поясняя детально причин, отметим следующие факты:

1. Если $\lambda_i = 0$, то объект x_i, как и ранее, классифицирован правильно и находится на достаточном удалении от почти-разделяющей гиперплоскости (то есть вне разделяющей полосы).

2. Если $0 < \lambda_i < C$, то объект x_i лежит на границе разделяющей полосы, то есть является опорным вектором.

3. Если $\lambda_i = C$, то сразу выводы, касаемые объекта x_i, сделать нельзя: объект x_i либо правильно классифицирован, но лежит внутри разделяющей полосы, либо лежит на почти-разделяющей гиперплоскости, либо классифицирован неправильно.

Достаточно общепринятым считается следующее определение.

Определение 2.1.2 *Если $\lambda_i = C$, то объект x_i называется нарушителем.*

Итак нарушитель – это либо объект, который классифицирован неверно, либо объект, который оказался внутри почти-разделяющей полосы!

Как же теперь построить классификатор? Для его построения можно действовать по следующему алгоритму (аналогичному алгоритму в случае линейно разделимой выборки):

1. Определим w из соотношения

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i.$$

2. Найдем w_0. Для этого достаточно взять произвольный опорный вектор x и из соотношения $(w, x_i) - w_0 = y_i$ найти

$$w_0 = (w, x_i) - y_i.$$

3. Классификатор зададим аналитическим выражением

$$f(X) = (w, X) - w_0.$$

Дальнейшая классификация проводится по общим правилам, описанным в конце пункта 1.3.

Определение 2.1.3 *Построенный таким образом классификатор часто называется классификатором с мягким зазором (soft margin classifier).*

Итак, построим почти-разделяющую гиперплоскость для наших данных, представленных в самом начале данного пункта, если параметр $C = 3$. Мы видим на рисунке 15, что два представителя синих и один представитель красных являются опорными векторами, при этом классификация допускает ошибки на данных обоих классов: один красный попал не в ту область, как и один синий. Почти-разделяющая гиперплоскость задается уравнением

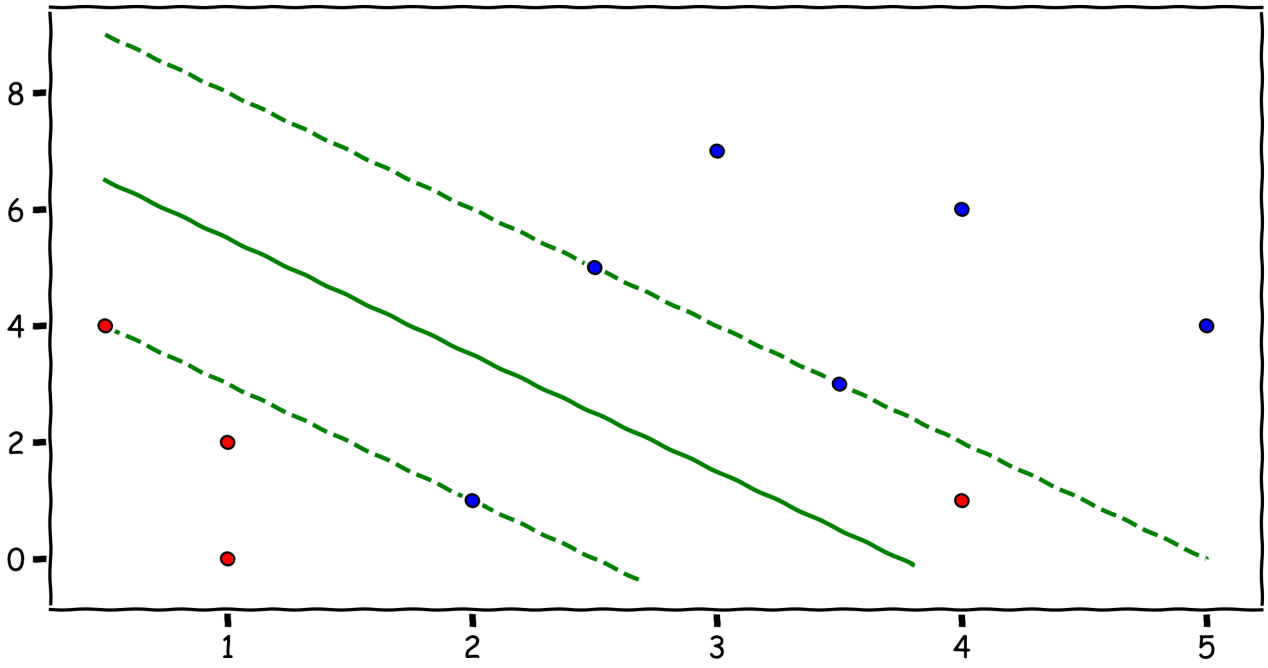


Рис. 15: Классификация с мягким зазором

$$-0.8X_1 - 0.4X_2 + 3 = 0,$$

а классификатор, в свою очередь, выражением

$$f(X) = f(X_1, X_2) = -0.8X_1 - 0.4X_2 + 3.$$

При $C = 1$ ситуация несколько меняется, она изображена оранжевыми линиями на рисунке 16. Разделяющая полоса стала шире, но, в то же время, классификация красных тренировочных данных стала менее уверенной – один из красных, который ранее являлся опорным вектором, стал хоть и правильно классифицированным, но нарушителем. Верхняя же граница у обеих почти-разделяющих полос одинаковая и на рисунке слилась.

Замечание 2.1.1 *Имея в руках построенных классификатор, можно понять, какие объекты-нарушители все же классифицированы правильно, а какие – нет. Для этого для тренировочных объектов x_i, у которых $\lambda_i = C$, достаточно вычислить $f(x_i)$ и сравнить знак полученного значения со знаком класса. Если знаки совпадают, то классификация выполнена верно, но объект находится внутри почти-разделяющей полосы, а значит классификацию нельзя считать «уверенной». Если же знаки отличаются, то классификация и вовсе проведена неверно. Если $f(x_i) = 0$, то объект оказывается лежащим на почти-разделяющей гиперплоскости и про его класс однозначно сказать ничего нельзя.*

Наверное, у внимательного слушателя все еще осталась некоторое внутреннее недовольство: ведь совсем не логично все на свете делить гиперплоскостью,

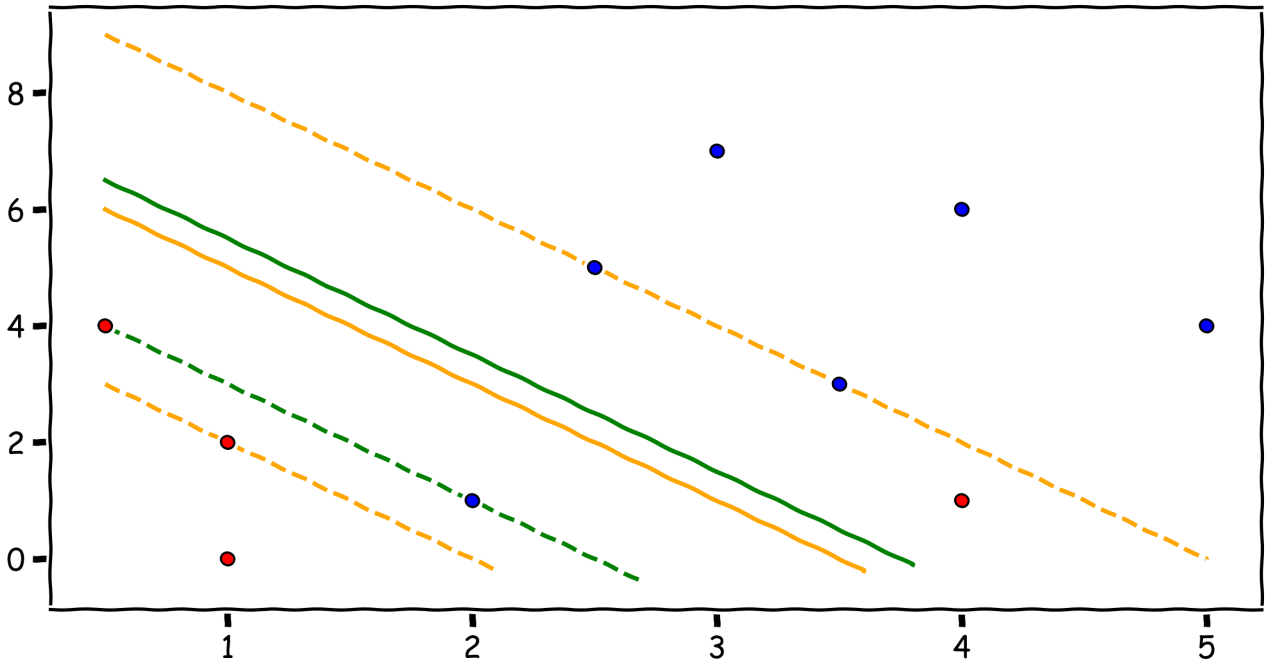


Рис. 16: Классификация с мягким зазором

хоть и допуская ошибки. Это правда. Рассмотрению этого вопроса и посвящен следующий пункт.

2.2 Ядра и преобразования пространств: общая теория

В заключение лекции осветим еще один подход к линейно неразделимым выборкам, когда и подход почти-разделяющей гиперплоскости оказывается не самым естественным. Скажем, для набора данных, изображенного на рисунке 17, разделение прямой не кажется логичным.

Идея происходящего далее проста, но технически весьма трудоемка. Она заключается в том, чтобы перейти к пространству признаков более высокой размерности, в котором данные окажутся линейно разделимы. Если ψ – такое преобразование, то в новом пространстве объекту x_i будет отвечать объект $\psi(x_i)$, а дальнейшее построение классификатора можно провести по ранее описанной схеме с тем только отличием, что теперь скалярное произведение (x_i, x_j) заменится на какое-то другое скалярное произведение элементов $\psi(x_i)$ и $\psi(x_j)$ в пространстве большей размерности. Итак, преобразование производится при помощи функции $\psi : \mathbb{R}^p \to H$, где H – пространство со скалярным произведением $(\cdot, \cdot)_H$. Вот только вопрос: а откуда взять ψ, и что это за пространство H? Начнем с некоторых безобидных определений.

Определение 2.2.1 *Пространство H в случае, если в нем исходный набор данных оказывается линейно разделимым, называется спрямляющим.*

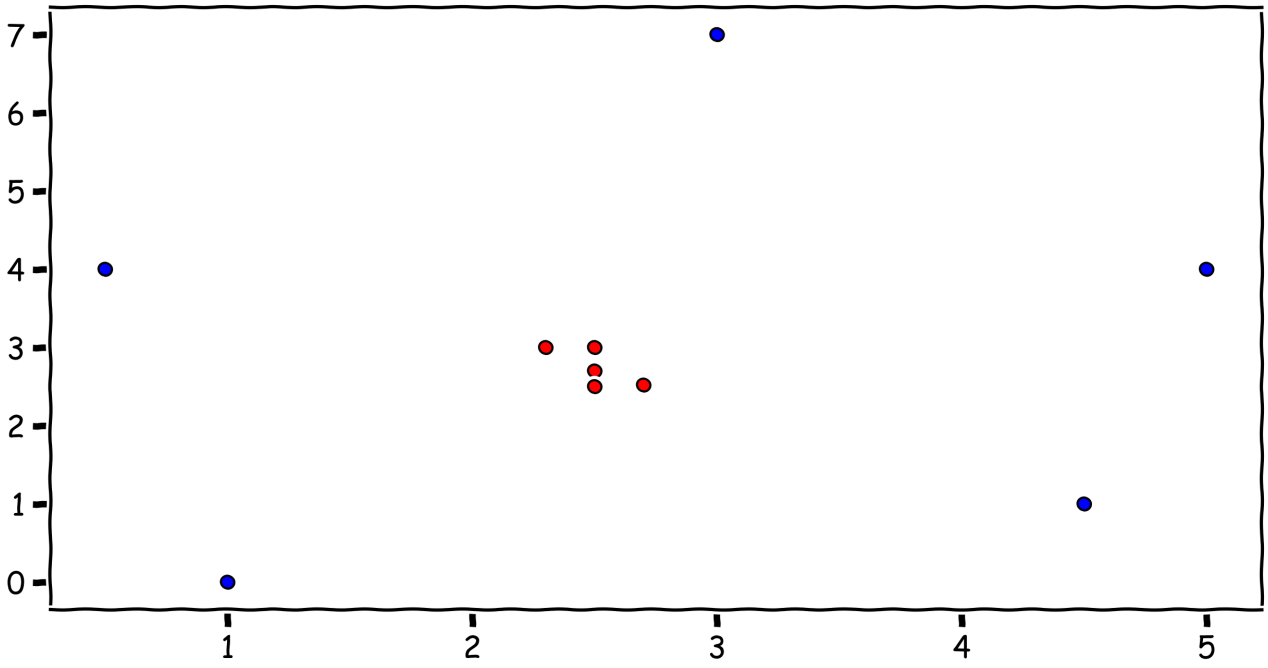


Рис. 17: Пример совсем линейно неразделимых данных

Замечание 2.2.1 *Ясно, что вместо линейной разделимости можно снова требовать почти-линейную разделимость, как было описано в предыдущем пункте лекции.*

Полезно еще раз взглянуть и освежить в память те оптимизационные задачи, которые мы решаем. Если это сделать, то видно, что постановки этих задач зависят только от скалярных произведений объектов, а не от их описаний. Отсюда мы и приходим к понятию ядра.

Определение 2.2.2 *Пусть $\psi : \mathbb{R}^p \to H$, где H – пространство со скалярным произведением $(\cdot,\cdot)_H$ – функция, переводящая объект $x_i \in \mathbb{R}^p$ в пространство более высокой размерности. Функция $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ называется ядром, если она представима в виде*

$$K(x, x') = (\psi(x), \psi(x'))_H.$$

Из этого определения, похоже, сразу понятно, что оптимизационная задача, которая требует решения, с использованием ядер в случае линейно разделимой в H выборки перепишется, как

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n} \lambda_i + \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \to \min\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n} \lambda_i y_i = 0 \end{cases}$$

а в случае линейно неразделимой, как

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n} \lambda_i + \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \to \min\limits_{\lambda} \\ 0 \leq \lambda_i \leq C, \ i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n} \lambda_i y_i = 0 \end{cases}.$$

Как обычно, для определения w достаточно вычислить

$$w = \sum_{i=1}^{n} \lambda_i y_i \psi(x_i),$$

а для определения w_0 достаточно вычислить

$$w_0 = (w, \psi(x_i))_H - y_i$$

на опорном векторе $\psi(x_i)$. Классификатор же задается соотношением

$$f(X) = (w, \psi(X)) - w_0.$$

Важно отметить, что ядра оставляют «след» и в исходном пространстве: они просто меняют разделяющую кривую, смотрите. Используя ядра, выражение для классификатора можно переписать в виде

$$f(X) = (w, \psi(X)) - w_0 = \left(\sum_{i=1}^{n} \lambda_i y_i \psi(x_i), \psi(X) \right) - w_0 = \sum_{i=1}^{n} \lambda_i y_i K(x_i, X) - w_0.$$

Приравнивая это выражение к нулю, мы получим задание геометрического места точек пространства, разделяющего исходные данные. Давайте теперь напоследок обратимся к конкретным примерам ядер.

2.3 Конкретный пример разделения при помощи ядер

Давайте на примере поясним, какие бывают ядра и как происходит разделение.

Пусть $p = 2$, то есть мы работаем в двумерном пространстве признаков. Рассмотрим функцию $K(x, y) = (x, y)^2$ и покажем, что приведенная функция является ядром. Пусть $x = (x_1, x_2)$, $y = (y_1, y_2)$, тогда

$$K(x, y) = (x, y)^2 = (x_1 y_1 + x_2 y_2)^2 = x_1^2 y_1^2 + 2 x_1 y_1 x_2 y_2 + x_2^2 y_2^2.$$

Введем в рассмотрение пространство $H = \mathbb{R}^3$ и функцию $\psi : \mathbb{R}^2 \to H$, действующую по правилу:

$$\psi(x) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2), \quad x = (x_1, x_2).$$

Легко увидеть, что при перемножении векторов $\psi(x)$ и $\psi(y)$, как векторов в H, мы получим, что

$$(\psi(x), \psi(y))_H = x_1^2 y_1^2 + 2x_1 y_1 x_2 y_2 + x_2^2 y_2^2,$$

то есть что

$$K(x, y) = (\psi(x), \psi(y))_H.$$

Итак, введенная функция и правда является ядром. Посмотрим, как она работает и чем может нам помочь на уже знакомых данных. Описанное преобра-

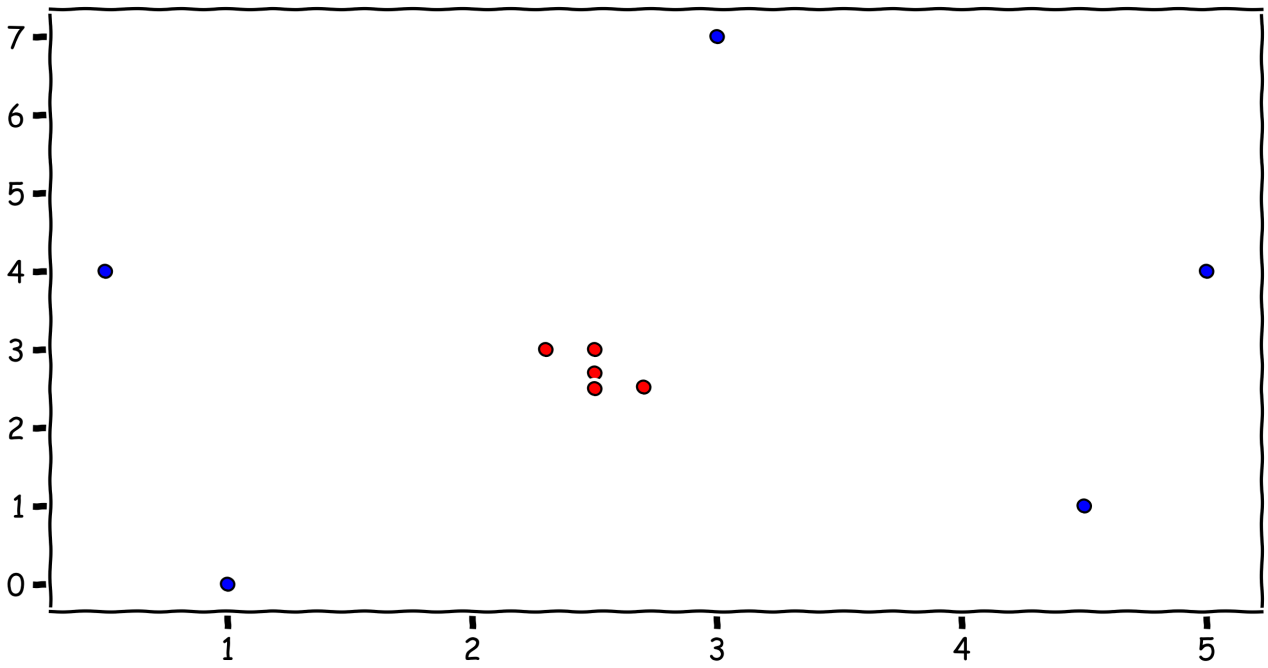


Рис. 18: Пример совсем линейно неразделимых данных

зование переводит набор данных в трехмерное пространство. Как видно, данные становятся линейно разделимыми Интересно посмотреть, что за фигура будет разделять наши данные на плоскости. Картина получается такая. Как мы можем видеть, наши данные разделились на плоскости гиперболой. Классификация геометрически тоже понятна: то, что находится «между» двумя ветками гиперболы относится к красным, то есть к классу «+1», а то, что «внутри» веток – к синим, к классу «−1».

Вроде все неплохо, хотя один вопрос все так же остался без ответа: откуда брать эти самые ядра, как их придумывать?

2.4 Немного о том, какие еще бывают ядра и какими свойствами они обладают

Про теорию получения ядер, а так же про то, какими свойствами они обладают, какая функция может быть ядром, а какая нет и все эти вопросы

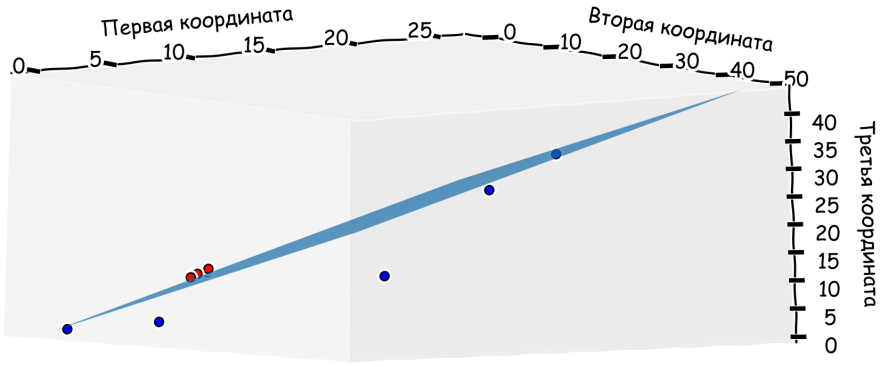


Рис. 19: Разделение данных в пространстве большей размерности

можно говорить не одну лекцию, и точно не простым языком. Мы позволим здесь себе сделать лишь небольшой обзор тех ядер, которые «на слуху», и имеются практически во всех математических пакетах и библиотеках. За более детальным изучением описанного вопроса настоятельно рекомендуем обратиться к дополнительной литературе.

Существует достаточно много стандартных ядер, которые, если их внимательно рассмотреть, как оказывается, приводят к известным алгоритмам: полиномиальному разделению, потенциальным функциям, двухслойным нейронным сетям и так далее. В итоге ядра выглядят очень многообещающе: они описывают (или заменяют) множество разных известных и используемых алгоритмов обучения с учителем. В то же время, как ни парадоксально, до сих пор нет эффективного общего подхода к их подбору в конкретных задачах. Давайте приведем несколько стандартных ядер, которые вшиты во множество пакетов и библиотек, а также опишем их преимущества.

1. Обобщим ранее приведенный пример. Пусть пространство признаков – это $\mathbb{R}^p$. Часто бывает удобно рассмотреть ядро

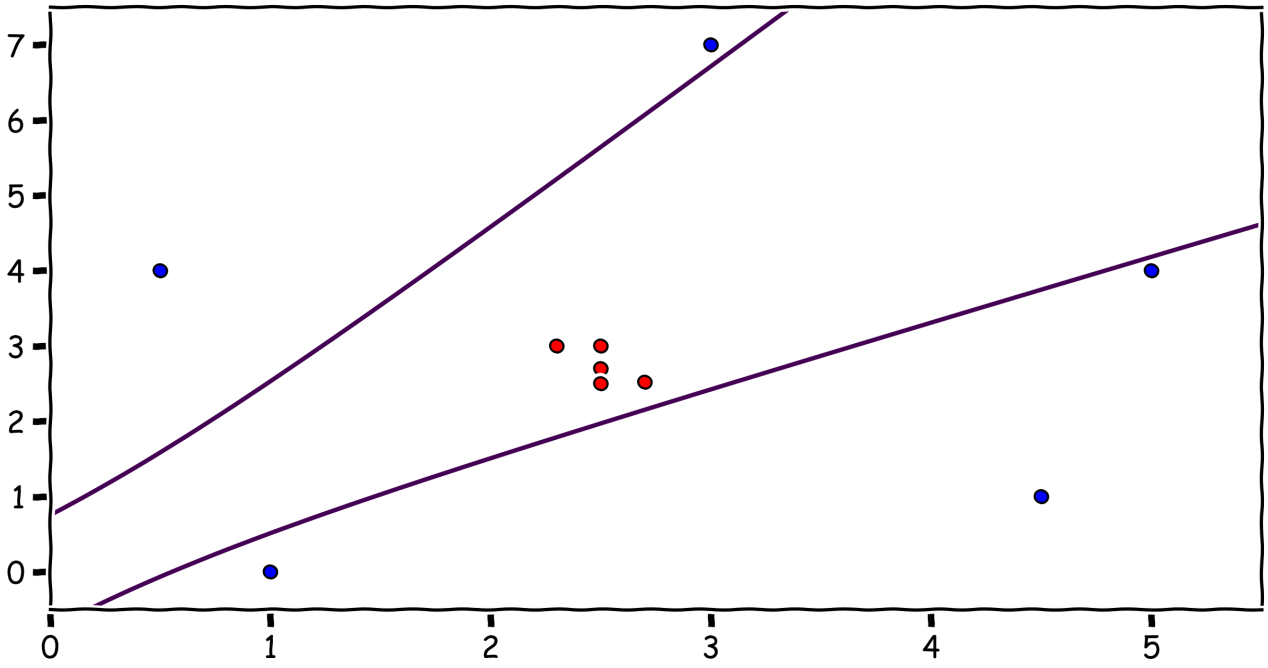
$$K(x, x') = (x, x')^d, \ d \in \mathbb{N}.$$

Это ядро удобно при разделении внутренности гиперэллипсоида от внешности, как в нашем примере ранее.

2. Пусть пространство признаков – это $\mathbb{R}^p$. Часто бывает удобно рассмотреть ядро

$$K(x, x') = (1 + (x, x'))^d, \ d \in \mathbb{N}.$$

Это – так называемое полиномиальное ядро. Разделение данных в новом про-



странстве эквивалентно разделению их полиномиальной разделимости в исходном пространстве.

3. Пусть пространство признаков – это $\mathbb{R}^p$. Часто вводят в рассмотрение так называемое радиальное или гауссово ядро (RBF – radial basis function)

$$K(x, x') = e^{-\beta\|x-x'\|^2}, \beta > 0.$$

Радиальное ядро обладает свойством локальности, а именно: если тестовое наблюдение находится далеко от тренировочного наблюдения, то значение радиального ядра чрезвычайно мало. Итак, ядро чувствительно лишь к близким обьектам.

Существует и множество других ядер, некоторые из которых разработаны под конкретные типы данных – целая наука. Но на эмпирическом уровне в инструментах, что находятся в руках, можно поступать следующим образом: всегда стоит сначала пробовать линейную разделимость или линейное ядро. В случае не очень большого обучающего набора даных имеет смысл использовать и гауссово ядро. Все дальнейшие исследования, как обычно, остаются на откуп самому исследователю.

2.5 Примеры применения

Давайте посмотрим на применение ядер к достаточно известному набору данных – ирисам Фишера. Так как для визуализации удобно использовать данные лишь с двумя предикторами, то оставим только два первых атрибута: длина наружной доли околоцветника и ширина наружной доли околоцветника и оставим (так как у нас двуклассовая классификация) лишь два типа цветков: setosa и versicolor.

Давайте сначала просто посмотрим на данные, рисунок 20. Заметим, что мы предварительно провели так называемую стандартизацию данных. По рисунку четко видно, что классы линейно разделимы. Итак, используя линей-

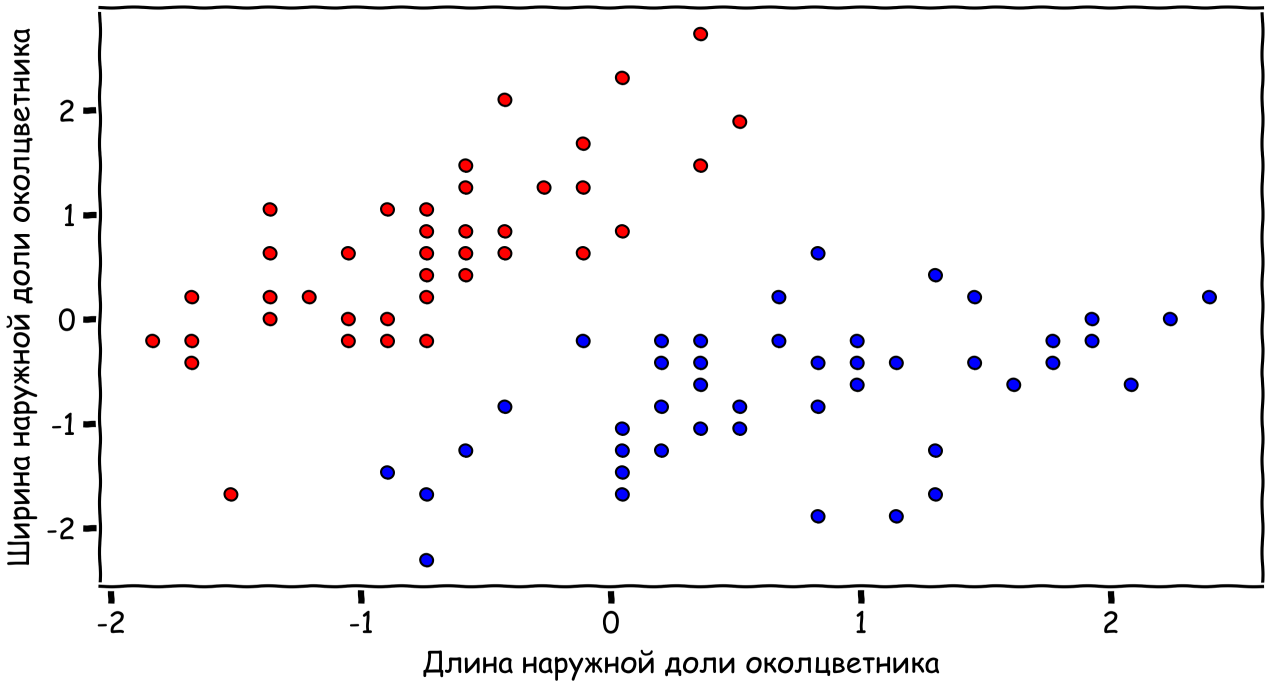


Рис. 20: Первые два атрибута цветков setosa и versicolor

ное разделение, получим следующий рисунок, рисунок 21. На нем же видно 4 опорных вектора, они обведены черными окружностями. А что, если попробовать разделить наши данные не гиперплоскостью? Давайте посмотрим. На рисунке 22 показано, как происходит разделение, используя полиномиальное ядро третьей степени. Видно, что данные остаются разделимыми. Опорные векторы изменились, они снова обведены. Следующий рисунок, рисунок 23 показывает, что происходит при использовании ядра RBF. Видно, как разделяющая «полоса» обвела две группы данных.

Теперь возьмем цветки versicolor и virginica и те же самые атрибуты. Посмотрите, данные перестали быть линейно разделимы, рисунок 24. Наверное, вообще сложно даже предположить: есть ли разумный способ их разделить. Классификатор, как видно, тоже с этим справляется плохо. Следующие три рисунка – это попытка разделить данные линейно (допуская ошибки), полиномиальным ядром третьей степени и ядром RBF. Обведены все объекты-нарушители. Как видно, ни одного «приличного» разделения нами так и не получено.

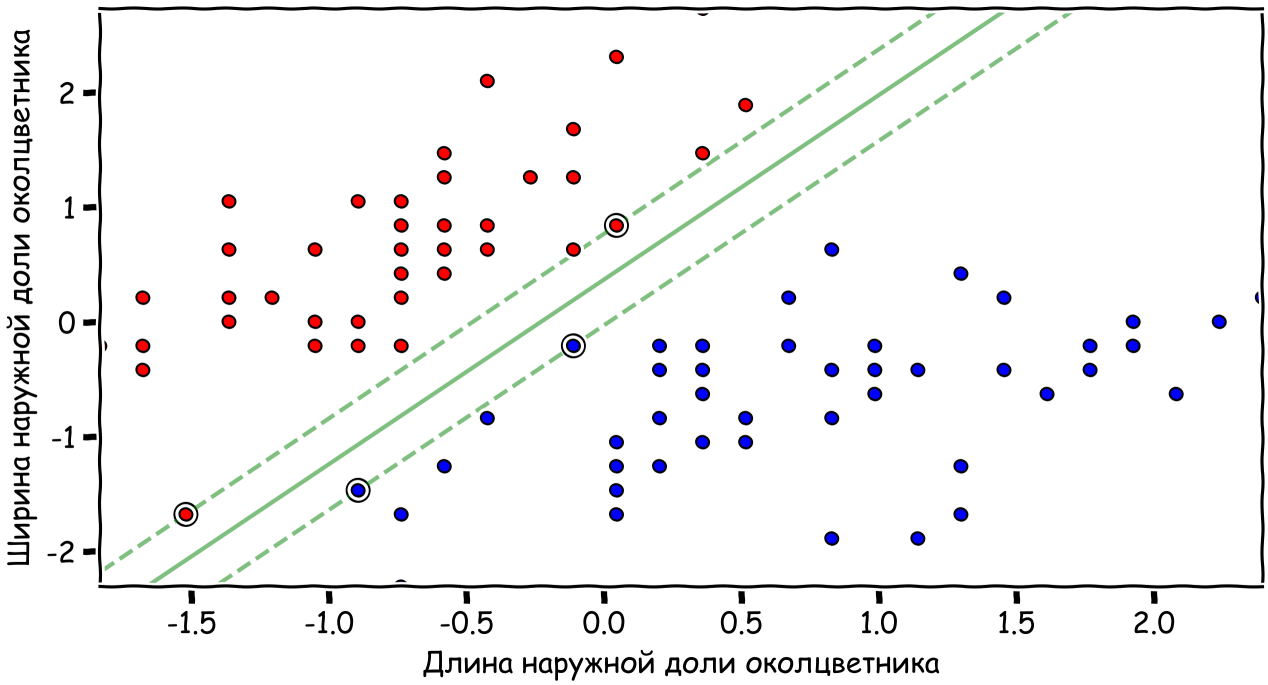


Рис. 21: Линейное разделение

3 Заключение

Итак, в этой лекции мы осветили еще один подход к классификации – метод опорных векторов. Как вы видите, все рассмотренные подходы неидеальны, и экспертное мнение, экспертный подход человека оказываются чрезвычайно важны. Кроме того, не все данные допускают разумное разделение. В то же время, метод опорных векторов – мощнейший инструмент для решения задачи классификации, хотя и выбор ядра, как обычно, задача, которая отдается на откуп исследователю.

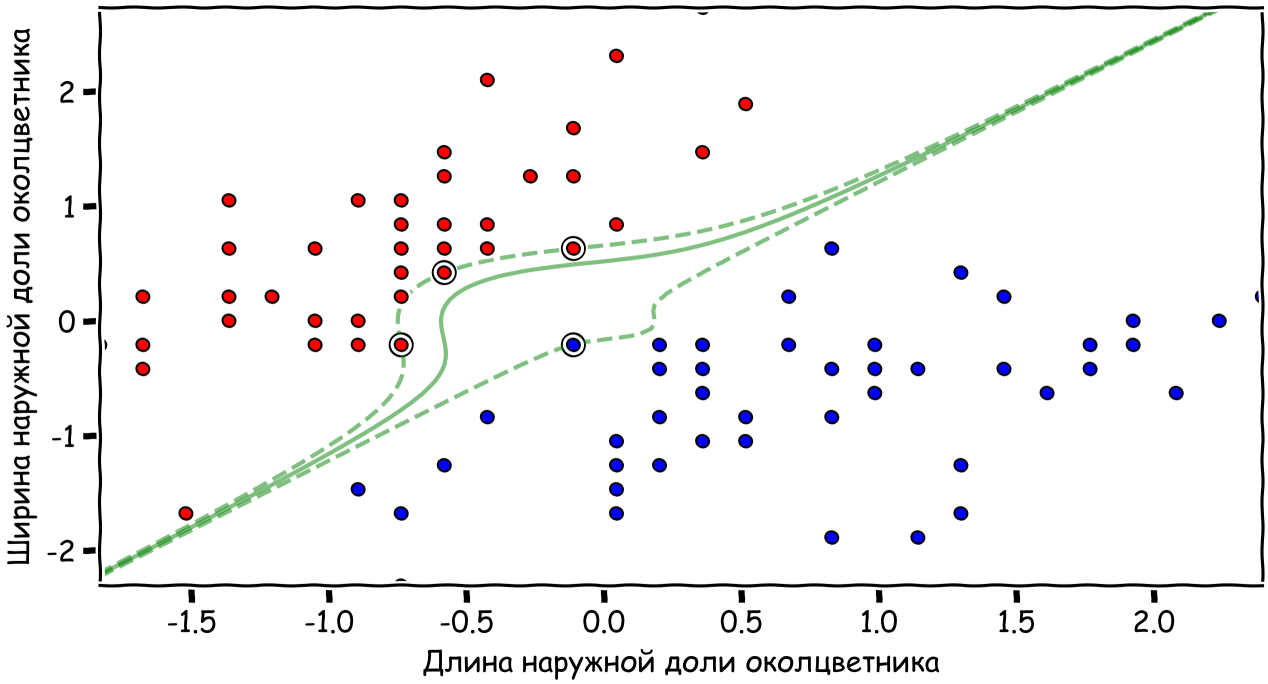


Рис. 22: Разделение полиномиальным ядром третьей степени

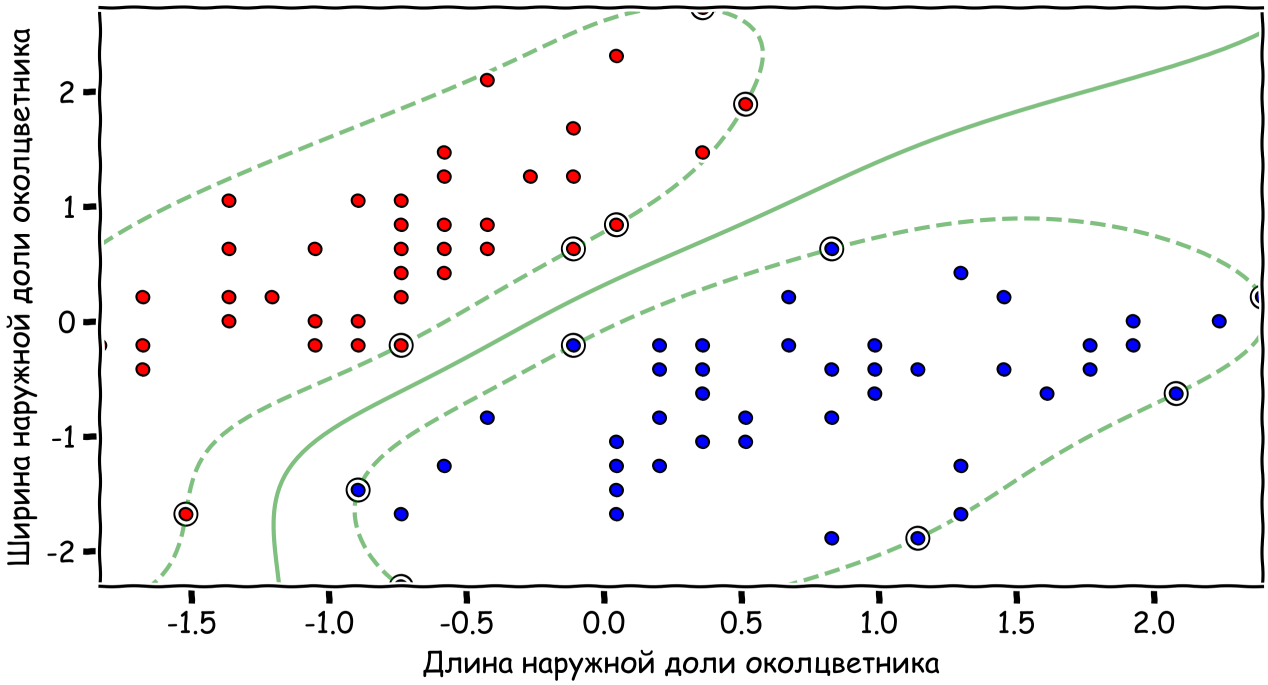


Рис. 23: Разделение ядром RBF

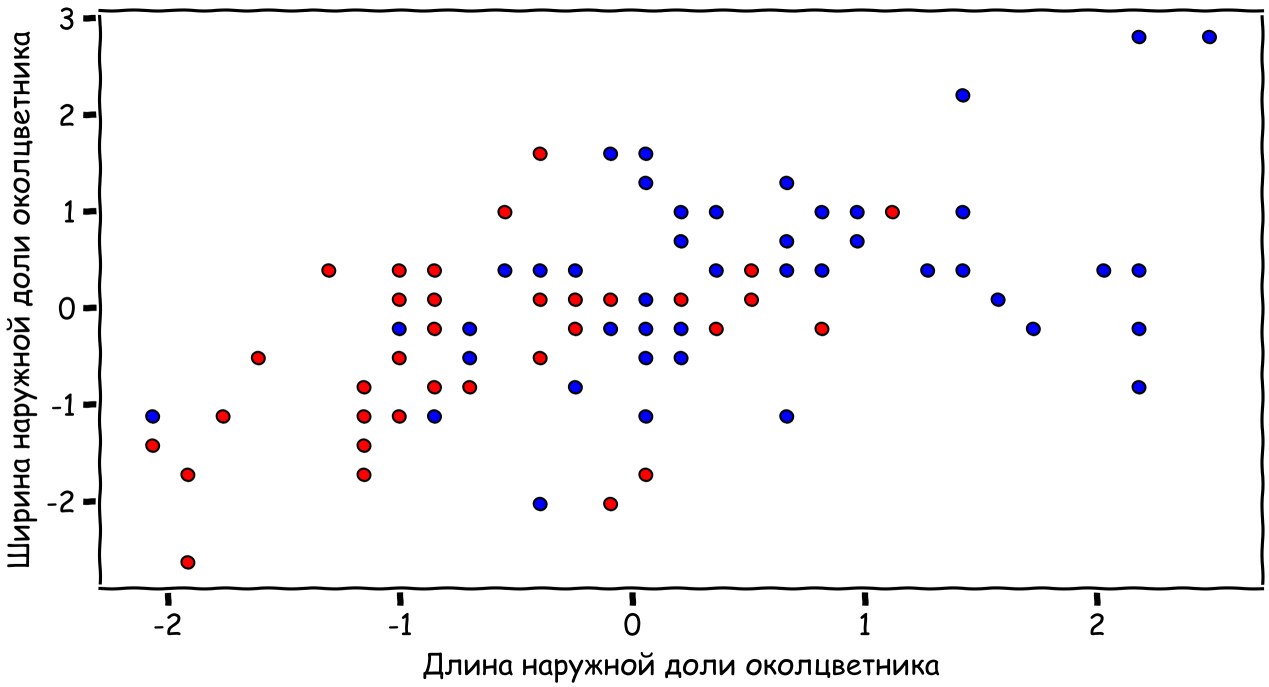


Рис. 24: Первые два атрибута цветков versicolor и virginica

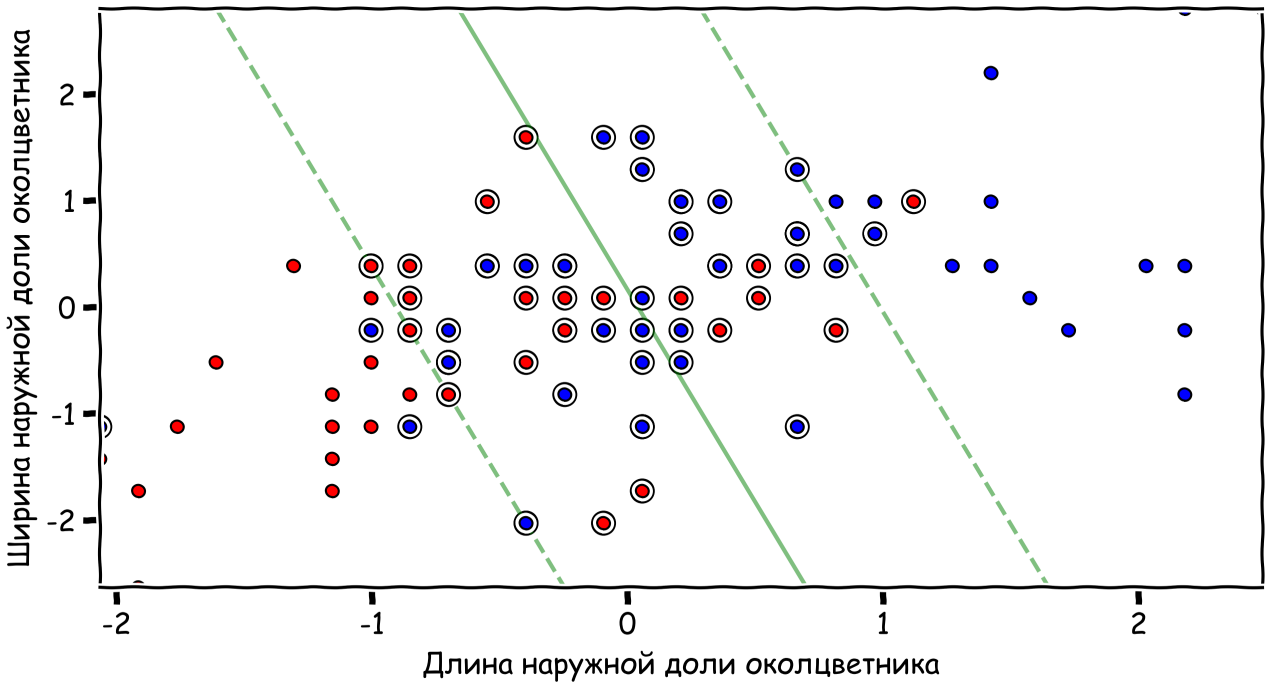


Рис. 25: Разделение гиперплоскостью

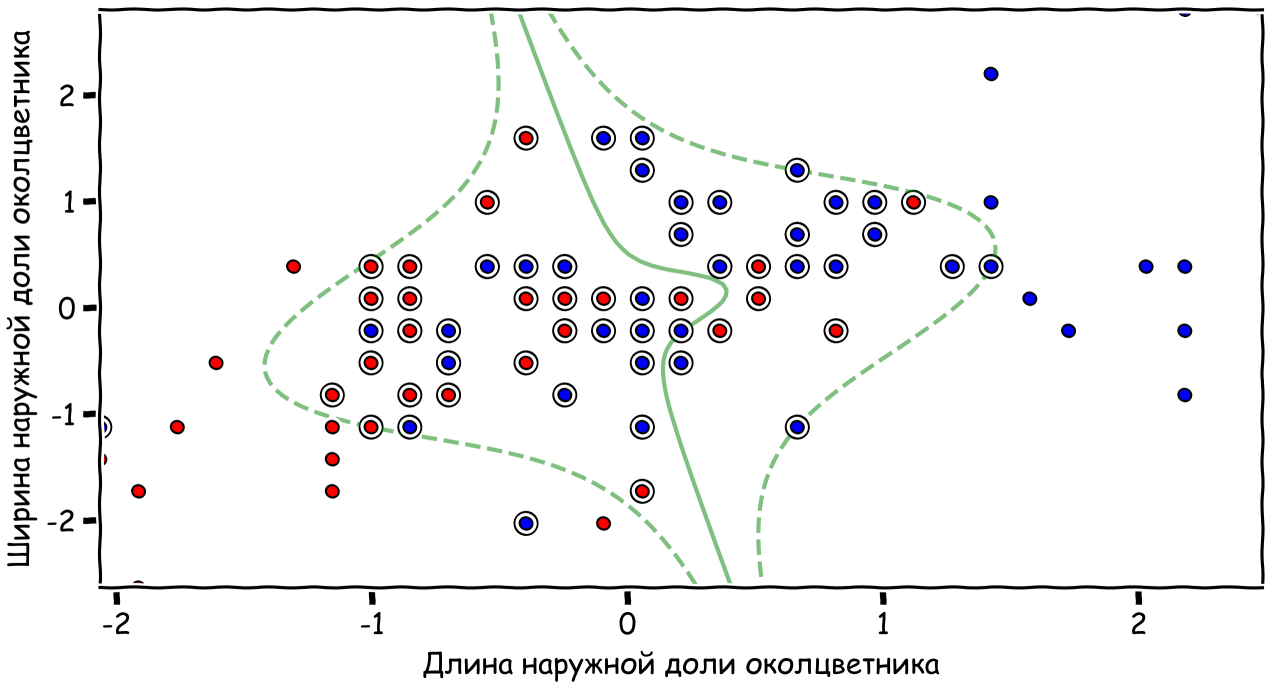


Рис. 26: Разделение полиномиальным ядром третьей степени

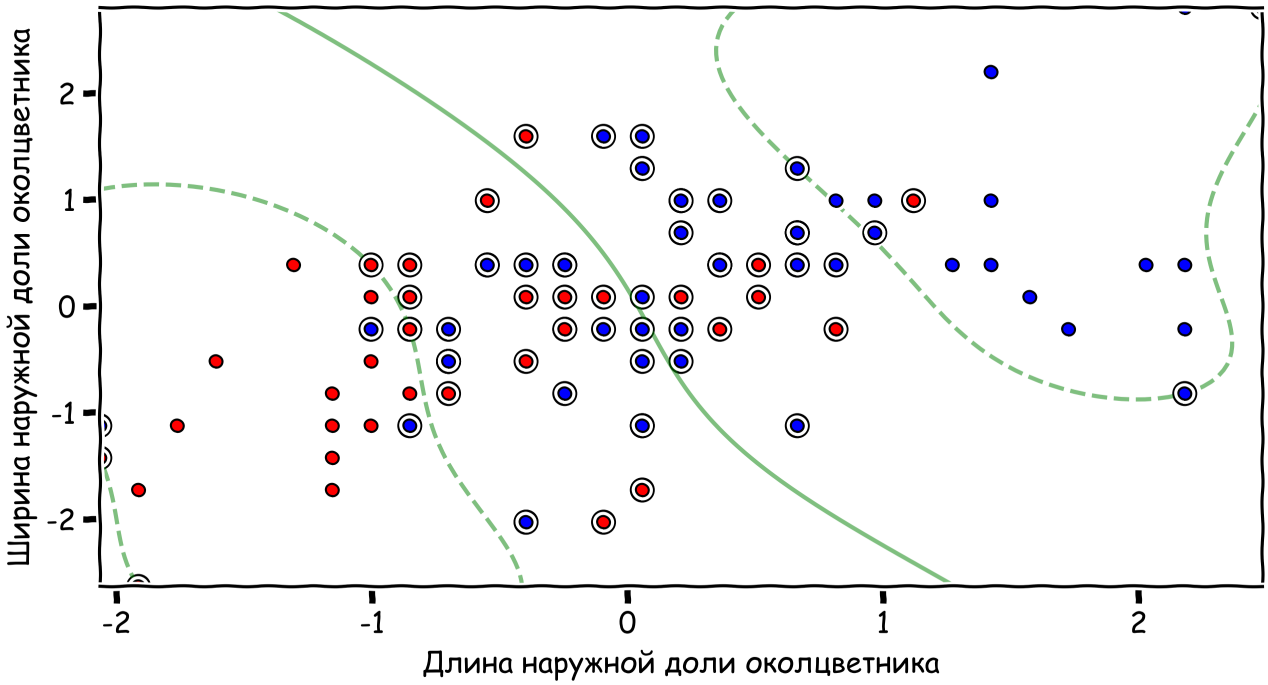


Рис. 27: Разделение ядром RBF