Курс по машинному обучению

Царьков Олег

Описание тикета OTSML-1

Введение. SVM (Support vector machine) – схема опорных векторов. Это один из первых алгоритмов машинного обучения, который оснывается на наивном представлении человечества о том, что мы видим предметы, ориентируясь по их границам.

Рассмотрим простейший случай этого алгоритма – на плоскости расположены точки. Точки будем обозначать x_i (это вектора, у каждой точки две координаты, но обозначать все это будем одной буквой). Введем для каждой точки число c_i, которое равно -1 или 1. Оно будет означать, относится ли точка к первому предмету или ко второму.

Предположим, что можно провести прямую линию так, что первый предмет окажется с одной стороны от нее, а второй предмет окажется с другой стороны от нее. Алгоритм направлен на то, чтобы отыскать эту прямую. Надо сказать, что он весьма неестественный. Он итеративный. Это означает, что он совершает постоянно один и тот же набор операций, который приближает неправильную разделяющую прямую (которая пересекает какой-нибудь из предметов, или вообще где-то в стороне находится) к правильной разделяющей прямой. И если вдруг у него это не получается, он работает до бесконечности. Например, это может не получиться, когда предметы вовсе неразделяемы прямой. Алгоритм настолько слабый и нечеткий, что даже если сначала подобрать плохую прямую, то он ее никогда не приблизит к хорошей. Но, мы научимся подбирать правильную изначальную прямую, так, чтобы алгоритм сходился к ответу.

Лагранжиан. Во-первых, требуется базовый навык нахождения минимума у функции. Это делается с помощью приравнивания производной к нулю.

Например,

$$a \cdot x^2 + b \cdot x + c \to min.$$

Посчитаем производную по x и приравняем ее нулю:

$$2a \cdot x + b = 0 \Longrightarrow x = -b/2a,$$

что совпадает с координатой вершины параболы, как это и должно было быть.

Однако, одной переменной в решении этой задачи мало. Когда есть фукнция от нескольких переменных, то для определения ее максимумов и минимумов нужно приравнять нулю производные по всем переменным в отдельности. Например,

$$x^2 + y^2 - 2y + 1 \to min.$$

Производная по x:

$$2x = 0,$$

по y:

$$2y - 2 = 0,$$

откуда получаем, что минимум в точке $x = 0, y = 1$.

Другой пример: функция

$$x^2 - 2xy + y^2 \to min.$$

Производная по x:

$$2x - 2y = 0,$$

по y:

$$-2x + 2y = 0,$$

откуда следует, что $x = y$. Действительно, во всех таких точках достигается минимум.

Но это еще не конец.

Оказывается, еще нужно уметь считать минимум при каких-то условиях. Например, минимум

$$\begin{cases} x^2 + y^2 \to min, \\ x \geqslant 1, \\ y \geqslant 2. \end{cases}$$

Тут уже не скажешь, что минимум при $x = 0, y = 0$, потому что требуется найти минимум среди точек $x \geqslant 1, y \geqslant 2$.

Для этого тоже есть четкий метод, который называется теоремой Куна-Таккера: если нужно решить задачу

$$\begin{cases} f(x_1, \dots, x_n) \to min, \\ g_i(x_1, \dots, x_n) \geqslant 0, \end{cases}$$

то нужно ввести дополнительные переменные λ_i для каждого i, и решать следующую задачу:

$$\begin{cases} f(x_1, \dots, x_n) - \lambda_i g_i(x_1, \dots, x_n) \to min_{x_1, \dots, x_n} max_{\lambda_1, \dots, \lambda_k}, \\ \lambda_i g_i(x_1, \dots, x_n) = 0, \\ \lambda_i \geqslant 0, \end{cases}$$

Ничего особо хитрого нет, при условиях $\lambda_i g_i(x_1, \dots, x_n) = 0$ минимизируемое выржаение как было $f(x_1, \dots, x_n)$, так и осталось. А вот условия $g_i(x_1, \dots, x_n) \geqslant 0$ исчезли. Остались лишь незначительные условия $\lambda_i \geqslant 0$ на независимые переменные λ_i.

Рассмотрим пример:

$$\begin{cases} x^2 + y^2 \to min, \\ x - 1 \geqslant 0, \\ y - 2 \geqslant 0. \end{cases}$$

Согласно вышеизложенному методу, напишем

$$\begin{cases} x^2 + y^2 - \lambda_1(x-1) - \lambda_2(y-2) \to min_{x,y} max_{\lambda_1, \lambda_2}, \\ \lambda_1(x-1) = 0, \\ \lambda_2(y-2) = 0, \\ \lambda_1 \geqslant 0, \lambda_2 \geqslant 0. \end{cases}$$

Вначале нужно найти минимум выражения $x^2 + y^2 - \lambda_1(x-1) - \lambda_2(y-2)$ по x, y. Приравняем для этого производные к нулю.

Производная по x:

$$2x - \lambda_1 = 0 \Longrightarrow x = \lambda_1/2.$$

Производная по y:

$$2y - \lambda_2 = 0 \Longrightarrow y = \lambda_2/2.$$

Теперь, подставим это в исходное выражение:

$$\frac{\lambda_1^2}{4} - \lambda_1\left(\frac{\lambda_1}{2} - 1\right) + \frac{\lambda_2^2}{4} - \lambda_2\left(\frac{\lambda_2}{2} - 2\right) = -\frac{\lambda_1^2}{4} + \lambda_1 - \frac{\lambda_2^2}{4} - 2\lambda_2.$$

Оставшееся выражение надо максимизировать по переменным λ_1, λ_2. Посчитаем его производные.

Производная по λ_1 :

$$-\frac{\lambda_1}{2}+1=0 \Longrightarrow \lambda_1=2.$$

Производная по λ_2:

$$-\frac{\lambda_2}{2}+2=0 \Longrightarrow \lambda_2=4.$$

Условия $\lambda_i \geqslant 0$ соблюдены. Подставляем полученные выражения и получаем:

$$x=\lambda_1/2=1,$$
$$y=\lambda_2/2=2,$$

условия

$$\lambda_1(x-1)=0, \lambda_2(y-2)=0$$

оказались соблюдены автоматически.

Получаем, что в задаче

$$\begin{cases} x^2+y^2 \to min, \\ x-1 \geqslant 0, \\ y-2 \geqslant 0. \end{cases}$$

ответ

$$x=1,$$
$$y=2,$$

что итак понятно.

Разберем еще один пример:

$$\begin{cases} x^2+y^2 \to min, \\ x+1 \geqslant 0, \\ y-2 \geqslant 0. \end{cases}$$

Эта задача эквивалентна такой:

$$\begin{cases} x^2+y^2-\lambda_1(x+1)-\lambda_2(y-2) \to min_{x,y} max_{\lambda_1,\lambda_2}, \\ \lambda_1(x+1)=0, \\ \lambda_2(y-2)=0, \\ \lambda_1 \geqslant 0, \lambda_2 \geqslant 0. \end{cases}$$

Вначале нужно найти минимум выражения $x^2+y^2-\lambda_1(x+1)-\lambda_2(y-2)$ по x, y. Приравняем для этого производные к нулю.

Производная по x:

$$2x-\lambda_1=0 \Longrightarrow x=\lambda_1/2.$$

Производная по y:

$$2y - \lambda_2 = 0 \Longrightarrow y = \lambda_2/2.$$

Теперь, подставим это в исходное выражение:

$$\frac{\lambda_1^2}{4} - \lambda_1\left(\frac{\lambda_1}{2}+1\right) + \frac{\lambda_2^2}{4} - \lambda_2\left(\frac{\lambda_2}{2}-2\right) = -\frac{\lambda_1^2}{4} - \lambda_1 - \frac{\lambda_2^2}{4} - 2\lambda_2.$$

Оставшееся выражение надо максимизировать по переменным λ_1, λ_2. Посчитаем его производные.

Производная по λ_1 :

$$-\frac{\lambda_1}{2} - 1 = 0 \Longrightarrow \lambda_1 = -2.$$

Производная по λ_2:

$$-\frac{\lambda_2}{2} + 2 = 0 \Longrightarrow \lambda_2 = 4.$$

Вот в этом примере-то мы и видим проблему. Получается, что условие $\lambda_1 \geq 0$ не верно. Нужно было найти максимум при положительных значениях λ_1. В данном случае, он будет в точке $\lambda_1 = 0$, потому что при $\lambda_1 \geqslant -2$ функция убывает, что видно из того, что ее производная меньше нуля. Значит, ее максимум при $\lambda_1 \geqslant 0$ в точке 0.

Получаем

$$x = \lambda_1/2 = 0,$$
$$y = \lambda_2/2 = 2,$$

условия

$$\lambda_1(x+1) = 0, \lambda_2(y-2) = 0$$

оказались соблюдены автоматически.

Получаем, что ответ здесь – $x = 0, y = 2$.

Вывод – когда мы решаем задачу на условный минимум, нам не помогает искать нули производной, потому что они могут не удовлетворять накладываемым условиям. Но наш метод позволил нам вместо сложных условий иметь неравенства $\lambda_i \geqslant 0$ для независимых переменных λ_i, и все равно можно сказать, как ведет себя функция при изменении λ_i, посчитав ее производную. Либо производная обнулится в точке, где $\lambda_i > 0$ (и тогда все хорошо), либо в точке, где $\lambda_i < 0$, и тогда λ_i “провалится” в 0.

Системы линейных уравнений. Система линейных уравнений — это система уравнений

$$\begin{cases} a_{11} \cdot x_1 + \ldots + a_{1n} x_n = b_1, \\ \vdots \\ a_{m1} \cdot x_1 + \ldots + a_{mn} x_n = b_m. \end{cases}$$

с переменными $x_1, \ldots, x_n$, а все числа $a_{i,j}$ и b_i – заданные заранее числа.

Основная идея при решении такой системы заключается в том, что если вычесть из одного уравнения другое, домноженное на какой-то коэффициент, то множество решений системы никак не изменится. Такие операции вычетания будем называть преобразованиями. Идея состоит в том, чтобы такими преобразованиями сделать как можно больше коэффициентов a_{ij} и b_i равными нулю. Тогда систему будет легче решать.

Будем действовать по следующему алгоритму: выпишем матрицу из коэффициентов

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} & b_m \end{pmatrix}.$$

Пусть первые $c_1 - 1$ столбцов состоят из одних нулей, и только в c_1 столбце есть ненулевой элемент $a_{i,c_1} \neq 0$. Поделим i-ую строку на a_{i,c_1} и поменяем ее с первой (уравнения можно делить на числа и менять местами).

Получим матрицу (заново переобозначим ее элементы через a_{ij}, b_j)

$$\begin{pmatrix} 0 & \ldots & 0 & 1 & \ldots & a_{1n} & b_1 \\ 0 & \ldots & 0 & a_{2,c_1} & \ldots & a_{2n} & b_2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & a_{m,c_1} & \ldots & a_{mn} & b_m \end{pmatrix}.$$

Далее, из каждого i-ого уравнения, начиная со второго, вычтем первое, домноженное на a_{i,c_1}. Тогда получим матрицу

$$\begin{pmatrix} 0 & \ldots & 0 & 1 & \ldots & a_{1n} & b_1 \\ 0 & \ldots & 0 & 0 & \ldots & a_{2n} & b_2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & a_{mn} & b_m \end{pmatrix}$$

с уже другими числами a_{ij}, b_j (не хочется придумывать новые обозначения. Главное – где остались нули, а где еще непонятно, нули или не нули).

Далее забудем про первую строку. Не будем делать с ней никаких операций, отложим ее в сторону. Будем иметь дело с матрицей

$$\begin{pmatrix} 0 & \dots & 0 & 0 & \dots & a_{2n} & b_2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & a_{mn} & b_m \end{pmatrix}.$$

Далее, как и раньше, найдем первый столбец, который не состоит целиком из нулей и обозначим его за c_2. Пусть $a_{i,c_2} \neq 0$. Поделим i-ое уравнение на это число и поменяем его местами со вторым. Получим

$$\begin{pmatrix} 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & a_{2n} & b_2 \\ 0 & \dots & 0 & 0 & \dots & 0 & a_{3,c_2} & \dots & a_{3n} & b_3 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & a_{m,c_2} & \dots & a_{mn} & b_m \end{pmatrix}.$$

Начиная с третьего уравнения, вычтем из всех второе, чтобы обнулить коэффициенты в стоблце:

$$\begin{pmatrix} 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & a_{2n} & b_2 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & a_{3n} & b_3 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & a_{mn} & b_m \end{pmatrix}.$$

Опять же, тут уже другие a_{ij} и b_j.

Если вернуть первую строчку, то получится

$$\begin{pmatrix} 0 & \dots & 0 & 1 & \dots & a_{1,c_2-1} & a_{1,c_2} & \dots & a_{1n} & b_1 \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & a_{2n} & b_2 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & a_{3n} & b_3 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & a_{mn} & b_m \end{pmatrix}.$$

Так можно продолжить этот алгоритм, убрав первые две строки, найдя ненулевой столбец начиная с третьей строки и так далее.

В итоге все уравнения будут иметь такой вид:

$$x_{c_k} + a_{k,c_k+1}x_{c_k+1} + \ldots + a_{k,c_k+2}x_{c_k+2} + \ldots + a_{k,n}x_n = b_k.$$

Причем, по построению видно, что $c_1 < c_2 < c_3 < \ldots < c_m$. Но, начиная с некоторого момента, мы могли не отыскать ненулевой столбец, то есть начиная с некоторой строки имееем нулевые строки. Нулевые строки – это тождественные уравнения $0 = 0$, их следует отбросить.

Могла быть строка, в которой $c_k = n + 1$, то есть все $a_{k,j} = 0, j \leqslant n = c_k - 1$, а коэффициент $b_k \neq 0$, то есть получается уравнение

$$0 = b_k,$$

которое не имеет ни одного решения. Если такое уравнение попалось, значит вся сиситема не имеет решений.

Осталось лишь разобрать случай, когда $c_1 < c_2 < c_3 < \ldots < c_m \leqslant n$. Система принимает вид

$$\begin{cases} x_{c_1} + \ldots + a_{1n}x_n = b_1, \\ \vdots \\ x_{c_m} + \ldots + a_{mn}x_n = b_m. \end{cases}$$

Такую систему уравнений решать совсем просто: значения переменных $x_{c_m+1}, \ldots, x_n$ задаем произвольно, затем выражаем

$$x_{c_m} = -a_{m,c_m+1}x_{c_m+1} - \ldots - a_{mn}x_n + b_m,$$

затем опять задаем значения переменных $x_{c_{m-1}+1}, \ldots, x_{c_m-1}$ произвольным образом, и выражаем

$$x_{c_{m-1}} = -a_{m-1,c_{m-1}+1}x_{c_{m-1}+1} - \ldots - a_{m-1,n}x_n + b_{m-1},$$

и так далее.

Таким образом и решается линейная система уравнений.

Теоретические основы SVM. Пусть x – точка на плоскости с двумя координатами. Пусть w – какой-то фиксированный вектор размерности 2. Пусть w_0 – число.

Тогда уравнение

$$< w, x > + w_0 = 0$$

задает прямую, где $< w, x >$ – скалярное произведение векторов. Действительно, если расписать скалярное произведение в координатах, получим

$$w_1 x_1 + w_2 x_2 + w_0 = 0,$$

что и правда является уравнением прямой на плоскости (x_1, x_2).

Неравенство

$$< w, x > + w_0 > 0$$

задает одну полуплоскость этой прямой,

$$< w, x > + w_0 < 0$$

задает другую.

Пусть полуплоскость, которая > 0, содержит в себе предмет, точки которого мы помечали $c_i = 1$, а другая полуплоскость содержит предмет, точки которого мы помечали $c_i = -1$.

Тогда условие того, что точки x_i правильно расположены относительно прямой, такое:

$$< w, x_i > + w_0 > 0 \text{ при } c_i = 1,$$
$$< w, x_i > + w_0 < 0 \text{ при } c_i = -1.$$

Это условие очень удобно переписывается в виде

$$c_i(< w, x_i > + w_0) > 0.$$

Причем это число чем меньше, тем ближе точка x_i расположена к нашей разделяющей прямой. Нетрудно понять, что в случае оптимального расположения этой прямой, будут точки x_+ и x_-, у которых $c_+ = 1, c_- = -1$ такие, что они равноудалены от прямой и находятся на ней на минимальном и одинаковом расстоянии. Иными словами, точки x_+ и x_- – ближайшие точки обоих предметов к разделяющей прямой – находятся от нее на одинаковом расстоянии.

Поделим координаты вектора w и число w_0 на одно и то же число (от этого прямая не меняется) так, чтобы

$$c_+(< w, x_+ > + w_0) = 1 = c_-(< w, x_- >, w_0).$$

Тогда для всех остальных точек x_i верно, что

$$c_i(< w, x_i > + w_0) \geqslant 1,$$

причем точки, для которых эта величина равна 1, являются самыми близкими к данной прямой и называются опорными векторами.

В области

$$-1 < < w, x > +w_0 < 1$$

нет точек нашей выборки. Эта область представляет собой полосу, заключенную между двумя параллельными прямыми

$$< w, x > +w_0 + 1 = 0$$

и

$$< w, x > +w_0 - 1 = 0.$$

Она называется разделяющей полосой. Отсюда становится понятно название опорных векторов, потому что они лежат на границе этой полосы и она как бы опирается на них, не трогая остальные точки. Выкидывание остальных точек из выборки не влияет никак на положение этой полосы.

Несложный подсчет ширины этой полосы показывает, что она равна

$$\frac{2}{||w||},$$

где через $||w||$ обозначается длина вектора w.

Если мы хотим построить более устойчивый классификатор, то мы хотим чтобы прямая как можно больше отстояла от точек первого предмета и от точек второго предмета. Ясно, что это выполняется, когда ширина полосы больше. То есть, в поисках прямой, нужно пробовать решать задачу

$$||w|| \to min.$$

Чем меньше длина этого вектора, тем шире полоса, тем дальше находятся точки первого и второго предмета от прямой и тем четче эта прямая их разделяет.

Вот мы и пришли к задаче Лагранжа

$$\begin{cases} ||w||^2 \to min, \\ c_i(< w, x_i > +w_0) \geqslant 1. \end{cases}$$

Как мы уже говорили ранее, она эквивалентна такой задаче

$$\begin{cases} \frac{1}{2}||w||^2 - \sum\limits_{i=1}^{n} \left(\lambda_i \left(c_i \left(< w, x_i > +w_0\right) - 1\right)\right) \to min_{w,w_0} max_{\lambda_i}, \\ \lambda_i \geqslant 0, \\ \lambda_i \left(c_i \left(< w, x_i > +w_0\right) - 1\right) = 0. \end{cases}$$

Пусть $w = (w_1, w_2)$ – координаты вектора w, $(x_{i,1}, x_{i,2})$ – координаты вектора x. Напомним, что мы ищем вектор w и число w_0, а вектора x_i и их разметка по классам c_i известна.

Перепишем выражение в координатах

$$\frac{1}{2}w_1^2 + \frac{1}{2}w_2^2 - \sum_{i=1}^{n} \left(\lambda_i c_i w_1 x_{i,1} + \lambda_i c_i w_2 x_{i,2} + \lambda_i c_i w_0 - \lambda_i\right).$$

Ищем его минимум, рассматривая производные по w_1, w_2, w_0.

По w_1:

$$w_1 - \sum_{i=1}^{n} \left(\lambda_i c_i x_{i,1}\right) = 0.$$

По w_2:

$$w_2 - \sum_{i=1}^{n} \left(\lambda_i c_i x_{i,2}\right) = 0.$$

По w_0:

$$\sum_{i=1}^{n} \left(\lambda_i c_i\right) = 0.$$

Теперь нужно в исходном выражении избавиться от w_1, w_2, w_0 и дальше максимизировать его по λ_i. Заметим, что по первым двум соотношениям, мы можем подставить

$$w_1 = \sum_{i=1}^{n} \left(\lambda_i c_i x_{i,1}\right),$$
$$w_2 = \sum_{i=1}^{n} \left(\lambda_i c_i x_{i,2}\right),$$

а про w_0 думать не надо, потому что коэффициент при нем $\sum_{i=1}^{n} \left(\lambda_i c_i\right)$ по третьему соотношению оказался равен нулю.

Имеем в результате:

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j x_{i,1}, x_{j,1}\right) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j x_{i,2}, x_{j,2}\right) -$$
$$- \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i c_i x_{i,1}\lambda_j c_j x_{j,1}\right) -$$
$$- \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i c_i x_{i,2}\lambda_j c_j x_{j,2}\right) + \sum_{i=1}^{n}\left(\lambda_i\right) =$$
$$= \sum_{i=1}^{n}\left(\lambda_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j x_{i,1}, x_{j,1}\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j x_{i,2}, x_{j,2}\right) =$$
$$= \sum_{i=1}^{n}\left(\lambda_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j < x_i, x_j >\right).$$

В итоге в этом выражении остались только переменные λ_i, по которым его надо максимизировать. Но это все равно что минимизировать это же выражение, домноженное на -1. В итоге, получаем задачу

$$\begin{cases} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j < x_i, x_j >\right) - \sum_{i=1}^{n}\left(\lambda_i\right) \to min, \\ \lambda_i \geqslant 0, \\ \lambda_i\left(c_i\left(< w, x_i > + w_0\right) - 1\right) = 0, \\ \sum_{i=1}^{n}\left(\lambda_i c_i\right) = 0. \end{cases}$$

Финальное решение задачи SVM. Осталось всего-лишь решить задачу

$$\begin{cases} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j < x_i, x_j >\right) - \sum_{i=1}^{n}\left(\lambda_i\right) \to min, \\ \lambda_i \geqslant 0, \\ \lambda_i\left(c_i\left(< w, x_i > + w_0\right) - 1\right) = 0, \\ \sum_{i=1}^{n}\left(\lambda_i c_i\right) = 0. \end{cases}$$

И потом рассчитать

$$w_1 = \sum_{i=1}^{n}\left(\lambda_i c_i x_{i,1}\right),$$

$$w_2 = \sum_{i=1}^{n}\left(\lambda_i c_i x_{i,2}\right),$$

далее взять какое-нибудь i, для которого $\lambda_i \neq 0$, и посчитать w_0 из условия

$$\lambda_i\left(c_i\left(< w, x_i > + w_0\right) - 1\right) = 0.$$

Подсчет w_1, w_2, w_0 и означает построение требуемой прямой.

Итак, решаем задачу

$$\begin{cases} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lambda_i\lambda_j c_i c_j < x_i, x_j >\right) - \sum_{i=1}^{n}\left(\lambda_i\right) \to min, \\ \lambda_i \geqslant 0, \\ \lambda_i\left(c_i\left(< w, x_i > + w_0\right) - 1\right) = 0, \\ \sum_{i=1}^{n}\left(\lambda_i c_i\right) = 0. \end{cases}$$

Сейчас бы мы радостно посчитали производные по λ_i и нашли бы ответ. Но мы помним, что если минимум достигается при $\lambda_i < 0$, то λ_i проваливается в 0 и бесполезно приравнивать нулю производную по λ_i.

Было бы так хорошо, если бы мы знали заранее, для каких индексов i $\lambda_i = 0$, а для каких индексов i $\lambda_i > 0$.

Вот здесь и начнется некая неточность алгоритма, его итеративность. Возьмем произвольным образом подмножество индексов A в множестве от 1 до n, для которых мы будем “думать”, что $\lambda_i > 0$. Для остальных индексов $\lambda_i = 0$, поэтому можно упростить наше выражение:

$$\frac{1}{2}\sum_{i\in A}\sum_{j\in A}\left(\lambda_i\lambda_j c_i c_j < x_i, x_j >\right) - \sum_{i\in A}\left(\lambda_i\right).$$

Пусть $A = \{a_1, \ldots, a_k\}$ – элементы множества A. Выражение принимает следующий вид:

$$\frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\left(\lambda_{a_i}\lambda_{a_j}c_{a_i}c_{a_k} < x_{a_i}, x_{a_j} >\right) - \sum_{i=1}^{k}\left(\lambda_{a_i}\right).$$

Кроме того, ранее имелось условие

$$\sum_{i=1}^{k}\left(\lambda_{a_i}c_{a_i}\right) = 0.$$

Выразим последнюю λ_{a_k} через предыдущие

$$\lambda_{a_k} = -\sum_{i=1}^{k-1}\left(\lambda_{a_i}c_{a_i}c_{a_k}\right)$$

и подставим это в наше выражение:

$$\frac{1}{2}\sum_{i=1}^{k-1}\sum_{j=1}^{k-1}\left(\lambda_{a_i}\lambda_{a_j}c_{a_i}c_{a_j} < x_{a_i}, x_{a_j} >\right) + \frac{1}{2}\lambda_{c_k}^2 < x_{a_k}, x_{a_k} > -$$
$$- \sum_{i=1}^{k-1}\left(\lambda_{a_i}\lambda_{c_k}c_{a_i}c_{a_k} < x_{a_i}, x_{a_k} >\right) - \sum_{i=1}^{k-1}\left(\lambda_{a_i}\right) + \sum_{i=1}^{k-1}\left(\lambda_{a_i}c_{a_i}c_{a_k}\right) =$$
$$= \frac{1}{2}\sum_{i=1}^{k-1}\sum_{j=1}^{k-1}\left(\lambda_{a_i}\lambda_{a_j}c_{a_i}c_{a_j} < x_{a_i}, x_{a_j} >\right) + \frac{1}{2}\sum_{i=1}^{k-1}\sum_{i=1}^{k-1}\left(\lambda_{a_i}c_{a_i}c_{a_k}\lambda_{a_j}c_{a_j}c_{a_k} < x_{a_k}, x_{a_k} >\right) -$$
$$- \sum_{i=1}^{k-1}\sum_{j=1}^{k-1}\left(\lambda_{a_i}\lambda_{a_j}c_{a_j}c_{a_k}c_{a_i}c_{a_k} < x_{a_i}, x_{a_k} >\right) - \sum_{i=1}^{k-1}\left(\lambda_{a_i}\right) + \sum_{i=1}^{k-1}\left(\lambda_{a_i}c_{a_i}c_{a_k}\right) =$$
$$= \frac{1}{2}\sum_{i=1}^{k-1}\sum_{j=1}^{k-1}\left(\lambda_{a_i}\lambda_{a_j}c_{a_i}c_{a_j} < x_{a_i}, x_{a_j} >\right) + \frac{1}{2}\sum_{i=1}^{k-1}\sum_{i=1}^{k-1}\left(\lambda_{a_i}\lambda_{a_j}c_{a_i}c_{a_j} < x_{a_k}, x_{a_k} >\right) -$$
$$- \sum_{i=1}^{k-1}\sum_{j=1}^{k-1}\left(\lambda_{a_i}\lambda_{a_j}c_{a_j}c_{a_i} < x_{a_i}, x_{a_k} >\right) - \sum_{i=1}^{k-1}\left(\lambda_{a_i}\right) + \sum_{i=1}^{k-1}\left(\lambda_{a_i}c_{a_i}c_{a_k}\right) =$$
$$= \frac{1}{2}\sum_{i=1}^{k-1}\sum_{j=1}^{k-1}\left(\lambda_{a_i}\lambda_{a_j}c_{a_i}c_{a_j}(< x_{a_i}, x_{a_j} > - < x_{a_k}, x_{a_j} - x_{a_k} >\right) - \sum_{i=1}^{k-1}\left((1 - c_{a_i}c_{a_k})\lambda_{a_i}\right)$$

Для этих индексов из A мы предполагаем, что узнаем минимум, если просто приравняем производную по λ_i к нулю. Сделаем это:

$$\sum_{j=1}^{k-1}\left(\lambda_{a_j}c_{a_i}c_{a_j} < x_{a_i}-x_{a_k}, x_{a_j}-x_{a_k} >\right) - 1 + c_{a_j}c_{a_k} = 0 \text{ для каджого } i = 1, 2, \ldots k-1.$$

Мы получили систему из $k-1$ уравнений, и неизвестных тоже $k-1$. Решая ее, получаем все $\lambda_{a_j}, j = 1, \ldots, k-1$. Далее, выражаем

$$\lambda_{a_k} = -\sum_{i=1}^{k-1}\left(\lambda_{a_i}c_{a_i}c_{a_k}\right),$$

а остальные λ_j считаем равными нулю, как уже было сказано.

Таким образом, все λ_j посчитаны. Но, это было бы так, если бы множетво индексов A было бы выбрано правильно. Поступим следующим образом: если для какого-то индекса i оказалось, что

$$\lambda_i < 0,$$

то будем считать, что переменная λ_i должна была "провалиться" в 0, и удалим ее из множества A.

Также проверим, не нарушается ли исходное условие

$$c_i(< w, x_i > + w_0) \geqslant 1$$

и если оно нарушается для какого-то i, то наоборот, добавим его в A.

Затем повторим все заново для нового множества индексов A. Если все условия соблюдены – хорошо. Если нет, снова поменяем A требуемым образом и снова запустим эту итерацию.

Очень важно понимать в этом месте, что никаких четких строгих математических объяснений, почему мы добавляем и удаляем элементы в A именно таким, а не каким-то другим образом нет. Как только мы делаем предположение о том, что есть множество A индексов, для которых $\lambda_i > 0$, а для остальных $= 0$, мы уже не можем пользоваться теоремой Куна-Таккера. Теорема просила нас решить задачу, ничего себе не предполагая. А мы взяли и предположили что-то. Неудивительно, что мы могли прийти к неправильному ответу, мы решили как-то подредактировать A и попробовать снова. Никто нам не может гарантировать, что за конечное число шагов мы получим ответ.

Мы можем зациклиться до бесконечности, никогда не дойдя до ответа. Самый простой пример этого – когда точки первого и второго объекта перемешаны, и их нельзя разделить прямой. Тогда сколько не заменяй это множество A, никогда нельзя построить такие λ_i, а по ним w_1, w_2, w_0, что все условия

$$c_i(< w, x_i > + w_0) \geqslant 1$$

были бы выполнены. И пришлось бы все время редактировать множество A до бесконечности.

Все-таки алгоритм хороший. Если точки первого и второго объекта все же разделяются прямой, то можно попробовать так подобрать изначальное множество A, чтобы алгоритм сошелся к ответу.

Множество A по смыслу содержит те индексы, для которых $\lambda_i > 0$, но, поскольку у нас есть условия

$$\lambda_i \left(c_i \left(< w, x_i > +w_0\right) - 1\right) = 0,$$

то для таких индексов верно

$$c_i \left(< w, x_i > +w_0\right) = 1,$$

а это условие, как пояснялось в самом начале, означает, что точки опорные – они находятся ближе всего к разделяющей прямой.

Попробуем с самого начала подобрать A так, чтобы это было правдой. Найдем точку первого предмета x_+ и точку второго предмета x_-, которые находятся ближе всего друг к другу. Их индексы и запишем в множество A.

Дальше, даже если мы плохо угадали, и есть еще опорные точки, наш итеративный алгоритм, описанный в предыдущем параграфе справится, и сам все достроит как надо.

Краткая схема итогового алгоритма.

1) Найти ближайшие две точки – одна помеченная -1, другая помеченная 1, запихать их индексы в множество $A = \{a_1, \ldots, a_k\}$.

2) Решить систему уравнений

$$\sum_{j=1}^{k-1} \left(\lambda_{a_j} c_{a_i} c_{a_j} < x_{a_i} - x_{a_k}, x_{a_j} - x_{a_k} >\right) - 1 + c_{a_j} c_{a_k} = 0 \text{ для каджого } i = 1, 2, \ldots k-1.$$

получив λ_{a_j} для индексов $j = 1, \ldots, k-1$. Находим λ_{a_k} по формуле

$$\lambda_{a_k} = -\sum_{i=1}^{k-1} (\lambda_{a_i} c_{a_i} c_{a_k})$$

Остальные λ_j положить равными нулю.

3) Рассчитать по λ_i числа w_1, w_2, пользуясь формулами

$$w_1 = \sum_{i=1}^{n} (\lambda_i c_i x_{i,1}),$$
$$w_2 = \sum_{i=1}^{n} (\lambda_i c_i x_{i,2}),$$

4) Взять какое-нибудь $i \in A$ и посчитать w_0:

$$w_0 = \frac{1}{c_i} - < w, x_i > .$$

5) Проверить условия:

а) Если $\lambda_i < 0$, удалить индекс i из множества A,

б) Если $c_i(< w, x_i > + w_0) < 1$, положить индекс i в множество A.

6) Если была хоть одна модификация на шаге 5), вернуться к шагу 2).

7) Закончить алгоритм и выдать w_1, w_2, w_0 в качестве ответа.