Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Прикладная математика»**  
  
  
  
  
  
  
КУРСОВАЯ РАБОТА

Метод опорных векторов

по дисциплине «Методы оптимизации»

Выполнил студент группы 33601/2

Лесик Демьян

30.11.16

Содержание

[Введение 3](#_Toc468210070)

[Постановка задачи 3](#_Toc468210071)

[Формализация задачи 4](#_Toc468210072)

[Описание алгоритма на примере задачи 8](#_Toc468210073)

[**Описание задачи** 8](#_Toc468210074)

[**Метод множителей Лагранжа** 8](#_Toc468210075)

[**Описание алгоритма** 9](#_Toc468210076)

[Случай неразделимой выборки и другие применения метода 11](#_Toc468210077)

[Список литературы 12](#_Toc468210078)

# Введение

Одной из основных задач метода опорных векторов(Supported Vector Machine) является задача бинарной классификации: определение к какому из двух изначально известных классов относится классифицируемый объект.

Если имеются образцы каждого класса - объекты, про которые заранее известно к какому классу они принадлежат, то такая задача называется *обучение с учителем ,* а известные данные называются *обучающей выборкой.*

Будем рассматривать SVM метод, относящийся к обучению с учителем.

# Постановка задачи

Дано **X** — пространство объектов (из ),***Y***— бинарные классы  
(***Y*** = {-1,1}). Так же дана обучающая выборка: . Требуется построить функцию (классификатор), сопоставляющий класс ***Y*** произвольному объекту ***x.***

# Формализация задачи

Пусть имеется множество прецедентов , где — обучающая выборка, а— множество меток двух   
классов и .

Требуется по обучающей выборке построить линейную решающую функции, т.е. такую линейную функцию , которая удовлетворяла бы условию

, для всех , и для всех .

Без ограничения общности можно считать, что метки классов равны  
 (1;-1). Тогда поставленную выше задачу можно переформулировать следующим образом. Будем искать, которая бы удовлетворяла условию

для всех

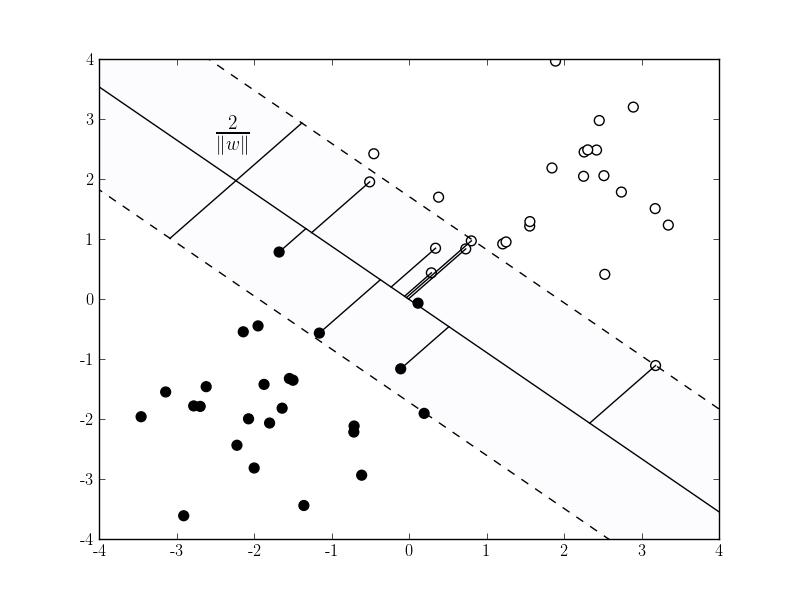
Умножая, если нужно, функцию на некоторое положительное число, система неравенств равносильна системе

для всех .

Этим действием выполняется нормировка.

Так как — линейная функция, то последняя система неравенств примет вид

(\*)

где — вектор весовых коэффициентов, — некоторое число.  
Тогда разделяющей два класса гиперплоскостью будет . Расстояние между граничными гиперплоскостями и равно .  
 На самих граничных плоскостях может находиться некоторое число обучающих векторов. Эти векторы называются *опорными*. 

Цель данного метода заключается в том, чтобы построить такую гиперплоскость, чтобы расстояние от неё до опорных векторов было максимально. Это значит, что расстояние между разделяющими гиперплоскостями должно быть как можно большим, что, в свою очередь значит, что должна быть как можно меньше. Таким образом, ставится задача нахождения минимума квадратичного функционала в выпуклом многограннике, задаваемым системой неравенств (\*). В выпуклом множестве квадратичный всегда имеет единственный минимум (если это множество не пусто). Из теоремы Куна — Таккера следует, что решение этой оптимизационной задачи равносильно поиску седловой точки лагранжиана:

в ортанте по множителям Лагранжа ,  
при условии, что

Последнее условие равносильно тому, что

или

Из необходимых условий существования седловой точки   
(полагая ) имеем:

Откуда следует, что вектор следует искать в виде

причем

В сумму с ненулевыми коэффициентами входят только те векторы, для которых .  
Такие векторы называют опорными, так как это именно те векторы, через которые будут проходить граничные гиперплоскости, разделяющие классы. Для найденного весового вектора смещение b можно вычислить как для любого опорного   
вектора .

Найдем значения множителей Лагранжа, как критических точек лагранжиана. Для этого подставим (4) и (5) в лагранжиан, получим

Таким образом, задача сводится к нахождению критических точек функции

Так как эта функция представляет собой разность линейной и квадратичной функций, причем квадратичная функция отрицательно определена, то требуется найти наибольшее значение функции при условии в области .

# Описание алгоритма на примере задачи

### **Описание задачи**

Для демонстрации данного метода, я выбрал следующую задачу: имеется две картинки( обучающая выборка). Одна цветная, другая та же, но переведенная в черно-белую( каждый пиксель или чёрный или белый).

Каждый пиксель имеет 3 компоненты(red, blue, green), я соответственно разбираю пиксель на эти компоненты. Разобрав так всё цветную картинку, получается массив объектов, каждая ячейка которой содержит 3 компоненты(пространство . Так как значение каждой компоненты меняется от 0 до 255, то областью действия является куб(0, 255).

Далее с помощью черно-белой картинки, каждой ячейки массива я присваиваю значение 1 или -1, в зависимости от цвета. Получается готовая выборка .

### **Метод множителей Лагранжа**

Вернемся к функции:

Есть много различных решений данной системы. Я выбрал правило множителей Лагранжа, как так этот метод является прямым( а не аппроксимационным) и наиболее понятен для программирования.

Функция Лагранжа - функция L(X,λ), определенная выражением L(X,λ) = F(X) + ∑λiφi(x), где λi - множители Лагранжа. **Функция Лагранжа**и спользуется при решении задач на условный экстремум.

составляется функция Лагранжа L(X) в виде линейной комбинации функции F(X) и ограничений gi(x);

1. находятся частные производные функции Лагранжа, ∂L/∂xi, ∂L/∂λi;
2. составляется система из *(n + m)* уравнений, ∂L/∂xi = 0.
3. определяются переменные xi и множители Лагранжа λi.

### **Описание алгоритма**

1. Создаю матрицу(possibleMatrixEq) частных производных (она симметричная)
2. Так как в эту систему могут входить не только опорные вектора, то данная матрица может быть вырождена, поэтому нахожу и удаляю "0" строки и столбцы. Так как действие происходит в , конечная матрица(matrixEq) 5-6 ранга.
3. Решаю данную систему методом Гаусса(MatrixOperations.toSolveSystem()), получаю массив корней. Нахожу в нем отрицательные корни ( сколько положительных корней, столько в системе опорных векторов)
4. Перебираю с помощью рекурсивной функции(LagrangSolution.findMaxF) все возможные комбинации строчек(в которых корни положительные) и запоминаю ту, где максимальна. Эта комбинация корней и будет решением данной задачи.( Те , которые получились отрицательными, зануляю, т.к.
5. Нахожу гиперплоскость W и смещение b по формулам и , где - первый найденный корень( он не обязательной 1 в исходной матрице!)

На этом математическое описание алгоритма заканчивается, т.к. я нашёл искомую гиперплоскость. Однако в программе далее происходит следующее:

На вход подаётся какая- то цветная картинка. В цикле она разбирается на пиксели, пиксели, в свою очередь, на RGB составляющие. Эти составляющее подставляются в функцию которая считает расстояние от плоскости до точки (HyperPlane. pointPosition(Point p, Plane plane)).Если расстояние больше нуля, то в новый массив пикселей записывается чёрный цвет, если меньше нуля - белый. Далее массив чёрно-белых пикселей записывается в новую картинку, которая и получается на выходе.

Полный код программы, а так же результат её работы можно посмотреть по ссылке: https://github.com/LesikDee/SVM

# Случай неразделимой выборки и другие применения метода

# Список литературы