Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Прикладная математика»**  
  
  
  
  
  
  
КУРСОВАЯ РАБОТА

Решение задачи классификации

методом опорных векторов

по дисциплине «Методы оптимизации»

Выполнил студент группы 33601/2

Лесик Демьян

30.11.16

Содержание

[Введение 3](#_Toc468230257)

[Постановка задачи 3](#_Toc468230258)

[Формализация задачи 4](#_Toc468230259)

[Описание алгоритма на примере задачи 8](#_Toc468230260)

[**Описание задачи** 8](#_Toc468230261)

[**Метод множителей Лагранжа** 9](#_Toc468230262)

[**Описание алгоритма** 9](#_Toc468230263)

[**Вывод** 10](#_Toc468230264)

[Случай линейно-неразделимой выборки 12](#_Toc468230265)

[Список литературы 13](#_Toc468230266)

# 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.

Случай с линейно разделимой выборкой - случай, когда существует гиперплоскость, проведя которую можно разграничить множество объектов на изначально известные классы. Соответственно, линейно-неразделимая выборка, когда такой гиперплоскости не существует.

В данной работе я рассматриваю случай с линейно - разделимой выборкой.

# 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

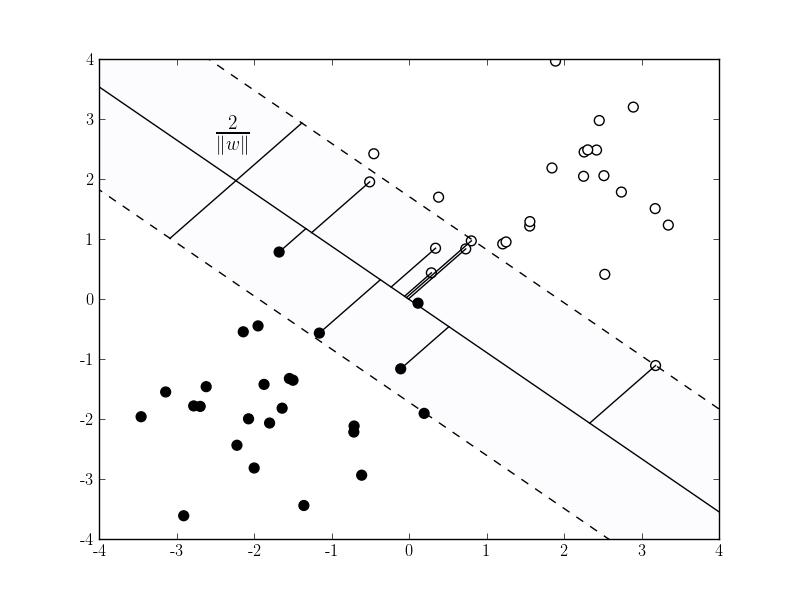
Умножая, если нужно, функцию на некоторое положительное число, система неравенств равносильна системе

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 (\*\*) и (\*\*\*) 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 такая 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 с диагональю: ((0,0,0);( 255, 255, 255)).

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

В моей программе в качестве обучающей выборки я взял картинку размерами 320\*240 пикселей. Итого 76800 пикселей. Понятно, из соображений архитектуры персонального компьютера, что если подставить все 76800 пикселей в решение системы, то не хватит ни памяти, ни точности, ни производительности. Поэтому сначала я произвожу три геометрические оптимизации, которые отсекают большинство точек, вектора которых точно не являются опорными.  
Надо понимать, что изначально множество точек в пространстве RGB не является линейно-разделимым, поэтому перед геометрическими оптимизациями необходимо сделать данное множество линейно-разделимым( всё же я рассматриваю модель).  
После разделение и оптимизаций осталось 7 точек, подозрительных на то, что их вектора опорные. А при дальнейшем решении выяснилось, что действительно опорных всего 5, что и следует ожидать в пространстве, исходя из геометрических соображений. Итак имеется обучающая выборка, состоящая из 7 точек.

### **Метод множителей Лагранжа**

Вернемся к функции:

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

Функция Лагранжа - функция , определяемая выражением , где - множители Лагранжа. Функция Лагранжа используется при решении задач на условный экстремум.  
Порядок действий таков:

1. Cоставляется функция Лагранжа в виде линейной комбинации функции и ограничений ;
2. Находятся частные производные функции Лагранжа, , ;
3. Cоставляется система из *(n + m)* уравнений, .
4. Определяются переменные и множители Лагранжа .

### **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 нулевые 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 и смещение b по формулам   
    и , где - первый найденный корень( он не обязательно 1 в исходной матрице!)

На этом математическое описание алгоритма заканчивается, т.к. я нашёл искомую гиперплоскость. Далее в программе происходит применение найденной гиперплоскости:

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.

### **Вывод**

Так как для каждого типа фотографий( пейзажи, портреты, натюрморты и пр.) нужна своя обучающая выборка, а так же, для каждой выборки нужно настраивать параметры геометрических оптимизаций, то практически данная задача не применима. Однако она наглядно показывает работу метода опорных векторов и выдаёт результаты практически идентичные, сравнимые с переводом цветную картинку в чёрно- белую с помощью инструмента Adobe Photoshop "Порог"(если совпадает тип фотографий).

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

Приведу несколько результатов:

Обучающая выборка:



Результат.  
 Cлева цветная картинка, справа полученная по заданной гиперплоскости:  




# Случай линейно-неразделимой выборки

Есть два способа решение данной задачи, если выборка является линейно-неразделимой.

1. Если не существует гиперплоскости, которая способна разделить обучающую выборку строго на 2 класса, то метод Soft margin выберет гиперплоскость, которая разделит обучающую выборку настолько чисто (с минимальной ошибкой), насколько это возможно, в то же время, максимизируя расстояние до ближайшей точки на обучающей выборке.  
   Метод вводит переменную , которая измеряет оценку ошибки классификации .  
   Целевая функция возрастает за счет функции штрафа, которая возрастает, если .  
   Оптимизация теперь требует нахождения компромисса между большой разницей и штрафом за ошибку.  
   Если штрафная функция является линейной, то задача оптимизации становится:  
   Ключевым преимуществом линейной штрафной функции является то, что переменные исчезают в двойственной задаче, с постоянной , появляющейся только как дополнительное ограничение на множители Лагранжа.
2. Все элементы обучающей выборки вкладываются в пространство более высокой размерности с помощью специального отображения . При этом отображение выбирается так, чтобы в новом пространстве выборка была линейно разделима.  
   Классифицирующая функция принимает вид  
    Выражение  
    называется *ядром* классификатора.

# Список литературы

1) http://www.machinelearning.ru

2) http://statistica.ru/branches-maths/metod-opornykh-vektorov-supported-vector-machine-svm/

3) https://habrahabr.ru/post/105220/

4) http://math.semestr.ru/math/lagrange.php