**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

**Факультет программной инженерии и компьютерной техники**

Отчет по лабораторной работе №3

по дисциплине «Интеллектуальная обработка экспериментальных данных»

***Тема* «Метод опорных векторов»**

Выполнила:

студентка гр. № P33212

Ян Цзяфэн

Санкт-Петербург

2021

# **Метод опорных векторов**

## Введение

Метод опорных векторов (SVM, support vector machine) — это модель бинарной классификации. Его базовая модель — это линейный классификатор с наибольшим зазором, определенным в собственной пространстве. Максимальный зазор отличает его от перцептрона. SVM также включает kernel trick, что делает его, по существу, нелинейным классификатором. Стратегия обучения SVM состоит в том, чтобы максимизировать зазор, который можно формализовать как задачу решения выпуклого квадратичного программирования (convex quadratic programming), что также эквивалентно задаче минимизации регуляризованной кусочно-линейной функции потерь. Алгоритм обучения SVM — это оптимальный алгоритм для решения выпуклого квадратичного программирования.

Метод опорных векторов включает построение моделей от простых до сложных: линейно разделяемый метода опорных векторов (linear SVM in linearly separable case), линейного метода опорных векторов (linear SVM) и нелинейного метода опорных векторов (non-linear SVM). Простая модель является основой сложной модели, а также частным случаем сложной модели. Когда обучающие данные являются линейно разделяемыми, линейный классификатор изучается путем максимизации жесткого зазора (hard margin maximization), то есть линейно разделяемый метод опорных векторов, также известная как метод опорных векторов с жёстким зазором (hard margin SVM). Когда обучающие данные являются приблизительно линейно разделяемыми, путем максимизации мягкого зазора (soft margin maximization) также изучается линейный классификатор, то есть линейный метод опорных векторов, также известная как метод опорных векторов с мягким зазором (soft margin SVM). Когда обучающие данные линейно неразделимы, нелинейный метод опорных векторов обучается с использованием kernel trick и максимизации мягкого зазора.

## Линейно разделяемый метод опорных векторов и максимизация жесткого зазора

### Линейно разделяемый метод опорных векторов

#### Линейно разделимый набор данных

Давайте сначала разберемся, что такое линейно разделимое.

В двухмерном пространстве точки двух классов полностью разделены прямой линией, называемой линейно разделимой.

Строгое математическое определение:

Данный набор данных:

,

где если существует гиперплоскость S

*,*

которая может разделять положительные точки экземпляра и отрицательные точки экземпляра набора данных точно по обе стороны от гиперплоскости (т. е. для всех экземпляров i с yi = + 1, wxi + b> 0, а для всех экземпляров i с yi = - 1, wxi + b < 0), то набор данных T называется линейно разделяемым набором данных; в противном случае набор данных T называется быть линейно неразлучными.

#### Линейно разделяемый метод опорных векторов

Учитывая линейно разделяемый обучающий набор данных, гиперплоскость разделения, полученная путем максимизации зазора или эквивалентного решения соответствующей задачи выпуклого квадратичного программирования, имеет вид

*,*

и соответствующая функция принятия решения о классификации

называются линейно разделяемым методом опорных векторов.

### Функциональный зазор и геометрический зазор

Вообще говоря, расстояние от точки до разделяющей гиперплоскости может указывать на степень уверенности в предсказании классификации.

#### Функциональный зазор

В случае, когда определяется гиперплоскость wx + b = 0, |wx + b| может относительно представлять расстояние точки x от гиперплоскости, а соответственность знака wx + b со знаком отметки класса y может указывать на правильность классификации. Следовательно, y(wx + b) может использоваться для выражения правильности и достоверности классификации.

Определим функциональный зазор гиперплоскости (w, b) относительно точки выборки (xi, yi) равен

.

Определим функциональный зазор гиперплоскости (w, b) относительно обучающего набора данных равен

#### Геометрический зазор

При выборе разделяющей гиперплоскости недостаточно только функционального зазора. Поскольку до тех пор, пока w и b изменяются пропорционально, гиперплоскость не изменяется, но функциональный зазор изменяется пропорционально. Следовательно, мы можем нормализовать вектор нормали w разделяющей гиперплоскости (‖w‖=1), так что зазор определяется, и тогда функциональный зазор становится геометрическим интервалом.

Обычно, когда точка выборки (xi,yi) правильно классифицируется гиперплоскостью (w,b), расстояние между точкой xi и гиперплоскостью (w,b) равно

.

Определим геометрический зазор гиперплоскости (w, b) относительно обучающего набора данных равен

.

Функциональный зазор и геометрический зазор имеют следующие отношения:

### Максимизация зазора

Для линейно разделяемого набора обучающих данных существует бесконечное количество линейно разделяемых разделяющих гиперплоскостей (эквивалентных персептрону), но разделяющая гиперплоскость с наибольшим геометрическим зазором уникальна. Нахождение гиперплоскости с наибольшим геометрическим зазором для набора обучающих данных означает классификацию обучающих данных с достаточной степенью уверенности.

Нахождение гиперплоскости разделения с наибольшим геометрическим зазором может быть выражено в виде следующей задачи оптимизации с ограничениями:

Учитывая отношения между геометрическим зазором и функциональным зазором, эту задачу можно переписать как

Значение функционального зазора не влияет на решение задачи оптимизации. Можно взять = 1.

И поскольку максимизация эквивалентна максимизации , она эквивалентна минимизации эту задачу можно переписать как

Найдём оптимальное решение w \*, b \*.

В результате гиперплоскость разделения:

И функция принятия о классификации:

#### Опорный вектор

В случае линейной разделимости экземпляр точки выборки, ближайшей к гиперплоскости разделения среди точек выборки набора обучающих данных, является опорным вектором. То есть выполняются следующие условия:

.

Для положительной точки экземпляра y = + 1 опорный вектор находится в гиперплоскости

.

Для отрицательной точки экземпляра y = - 1 опорный вектор находится в гиперплоскости

.

Расстояние между и называется зазором.

### Dual

Dual algorithm линейно разделяемого метода опорных векторов заключается в получении оптимального решения исходной задачи путем решения dual problem. Сначала построим функцию Лагранжа (Lagrange function). Для каждого неравенства ограничения введем множители Лагранжа (Lagrange multiplier) определим функцию Лагранжа:

где - вектор множителей Лагранжа

Согласно лагранжевой двойственности, двойственная задача исходной задачи является минимаксной:

Эту задачу можно переписать как

Можно получить оптимальное решение .

Вычислить

И выбрать положительный компонент при , вычислить

Можно получить гиперплоскость разделения

и соответствующую функцию принятия решения о классификации

## Линейный метод опорных векторов и максимизация мягкого зазора

### Линейный метод опорных векторов

На самом деле полностью линейно разделяемых данных практически не существует. Для решения этой проблемы введём набор дополнительных переменных , характеризующих величину ошибки на объектах .

Задача обучения линейного метода опорных векторных линейной неразделимости превращается в следующую задачу выпуклого квадратичного программирования:

где C > 0 - параметр настройки метода, который позволяет регулировать отношение между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки.

### Dual

Функция Лагранжа исходной задачи оптимизации имеет вид

где

Двойственная задача исходной задачи:

Значения w\* и b\* в исходной задаче можно получить следующим образом:

Гиперплоскость разделения:

Функция принятия решения о классификации:

## Нелинейный метод опорных векторов и ядра

Для решения задач линейной классификации очень эффективным методом является метод опорных векторов линейной классификации, но иногда задача классификации бывает нелинейной, тогда мы можем использовать нелинейный метод опорных векторов. Главной особенностью нелинейного метода опорных векторов является использование kernel trick.

Для задачи нелинейной классификации во входном пространстве она может быть преобразована в задачу линейной классификации в собственном пространстве с определенной размерностью посредством нелинейного преобразования, а нелинейный метод опорных векторов изучается в многомерном собственном пространстве. Поскольку в двойственной задаче обучения линейного метода опорных векторов целевая функция и функция принятия о классификации включают только внутренний продукт между экземплярами, поэтому нет необходимости явно указывать нелинейное преобразование, но используется функция ядра для замены внутреннего продукта. Функция ядра представляет собой внутренний продукт между двумя экземплярами после нелинейного преобразования. В частности, K(x,z) — это функция или положительно определенное ядро, что означает, что существует отображение из входного пространства в собственное пространство , для x, z в любом входном пространстве существует

В двойственной задаче обучения линейного метода опорных векторов используем функцию ядра K(x,z) для замены внутреннего продукта, и можем получить нелинейный метод опорных векторов:

Таким образом, мы можем получить алгоритм нелинейного метода опорных векторов следующим образом:

Вход: набор обучающих данных где, ;

Выход: гиперплоскость разделения и функция принятия о классификации.

1. Выбрать соответствующую функцию ядра K(x,z) и параметр C > 0, построить и решить задачу выпуклого квадратичного программирования

можно получить оптимальное решение

1. Выбрать положительный компонент при , и вычислить
2. Функция принятия о классификации:

Часто используемая функция ядра - функция ядра Гаусса

Соответствующий SVM является классификатором гауссовской радиальной базисной функции. В этом случае функцией принятия о классификации является

## Реализация метода

Задача: Распознавание лиц с использованием метода опорных векторов на основе набора данных LFW.

Решение на Python:

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.datasets import fetch\_lfw\_people

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import classification\_report

from sklearn.svm import SVC

from sklearn.decomposition import PCA

# Загрузка данных

lfw\_people = fetch\_lfw\_people(min\_faces\_per\_person=70, resize=0.4)

# Формат данных изображений

n\_samples, h, w = lfw\_people.images.shape

# Формат данных

lfw\_people.data.shape

# Классы

lfw\_people.target

target\_names = lfw\_people.target\_names

target\_names

n\_classes = lfw\_people.target\_names.shape[0]

# Разделим набор данных на набор данных для обучения и для тестирования.

x\_train, x\_test, y\_train, y\_test = train\_test\_split(lfw\_people.data, lfw\_people.target)

# Построим модель с помощью SVC и обучим модель

model = SVC(kernel='rbf', class\_weight='balanced')

model.fit(x\_train, y\_train)

# Прогнозирование с помощью набора тестовых данных

predictions = model.predict(x\_test)

print(classification\_report(y\_test,predictions,target\_names=lfw\_people.target\_names))

Изображение выглядит как текст

Автоматически созданное описание

# Используем PCA, чтобы уменьшить размерность набора данных

n\_components = 100

pca = PCA(n\_components=n\_components,whiten=True).fit(lfw\_people.data)

x\_train\_pca = pca.transform(x\_train)

x\_test\_pca = pca.transform(x\_test)

# Построим модель и обучим модель

model\_pca = SVC(kernel='rbf', class\_weight='balanced')

model\_pca.fit(x\_train\_pca,y\_train)

# Прогнозирование с помощью набора тестовых данных

predictions\_pca = model\_pca.predict(x\_test\_pca)

print(classification\_report(y\_test, predictions\_pca, target\_names=target\_names))

Изображение выглядит как текст

Автоматически созданное описание