МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ

Методические указания к лабораторным работам

(первый семестр)

Составитель: С. М. Наместников

Ульяновск

2022

УДК 621.394.343 (076)
ББК 32.88 я7
П33

Рецензент: Deep Learning Engineer компании Huawei, канд. техн. наук, Смирнов П.В.

Одобрено секцией методических пособий научно-методического совета Университета

Методы машинного обучения: методические указания к лабораторным работам (первый семестр) /сост. С. М. Наместников. – Ульяновск : УлГТУ, 2022. – 19 с.

Методические указания по курсу «Методы машинного обучения» для студентов направления 11.04.02 Инфокоммуникационные технологии и системы связи, профиль подготовки " Искусственный интеллект и анализ больших данных в обработке изображений " разработаны в соответствии с программой курса «Методы машинного обучения». Лабораторные работы посвящены исследованию и разработки основных методов машинного обучения с использованием языка Python.

Сборник подготовлен на кафедре «Телекоммуникации».

УДК 621.394.343 (076)
ББК 32.88 я7

СОДЕРЖАНИЕ

Лабораторная работа №1

Расчет коэффициентов разделяющей линии и вычисление отступа (margin) для объектов разных классов

Лабораторная работа №2

Обучение линейного алгоритма бинарной классификации образов с помощью градиентного алгоритма

Лабораторная работа №3

Исследование работы L2-регуляризатора в задачах регрессии

Лабораторная работа №4

Реализация наивного байесовского классификатора

Лабораторная работа №5

Реализация алгоритма метода опорных векторов для задачи бинарной классификации

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Лабораторная работа №1

Расчет коэффициентов разделяющей линии и вычисление отступа (margin) для объектов разных классов

Цель работы: научиться вычислять коэффициенты разделяющей линии и величину отступа (margin) при бинарной классификации объектов.

Теоретический материал

Теория для выполнения лабораторной работы доступна на следующих страницах сайта:

https://proproprogs.ru/ml

в разделах:

- Постановка задачи машинного обучения
- Линейная модель. Понятие переобучения
- Способы оценивания степени переобучения моделей
- Уравнение гиперплоскости в задачах бинарной классификации
- Решение простой задачи бинарной классификации

А также в соответствующих видеоматериалах, размещенных на странице сайта:

http://tk.ulstu.ru/video.php?id=3

Задания на лабораторную работу (по вариантам)

1. Используя рисунок своего варианта, необходимо вычислить коэффициенты

$$\omega = \left[\omega_0, \omega_1, \omega_2\right]^T$$

разделяющей линии, которая определяется выражением:

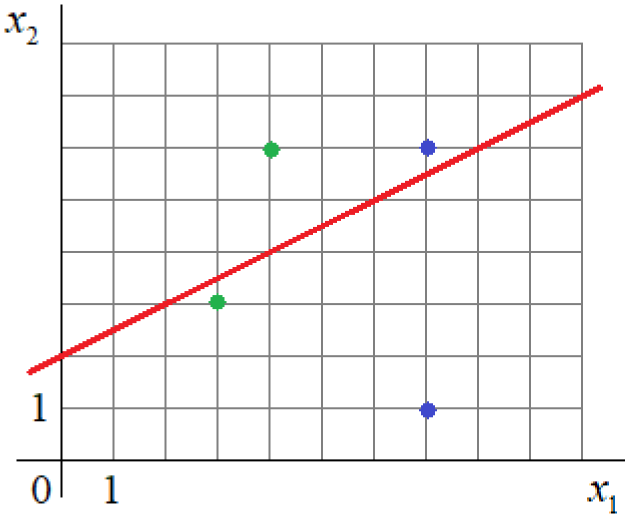
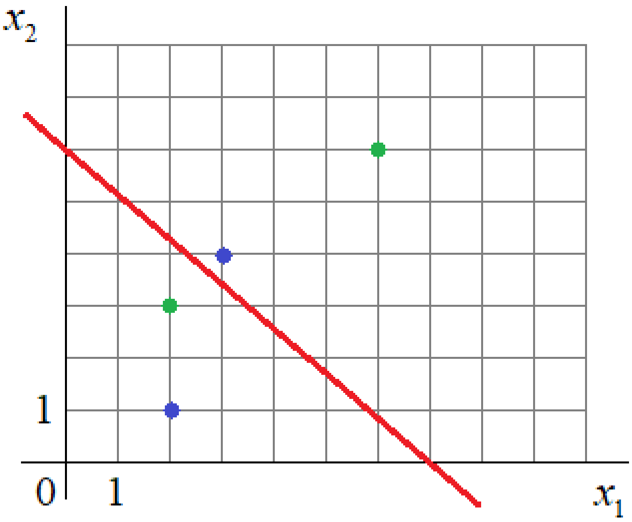
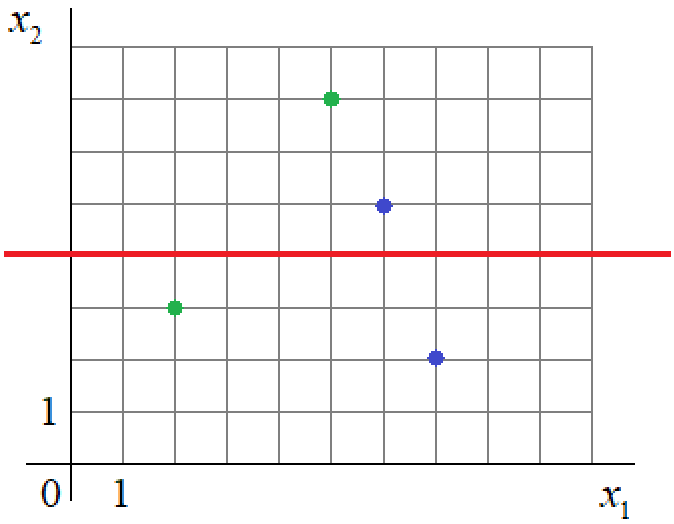
$$\omega_1 \cdot x_1 + \omega_2 \cdot x_2 + \omega_0 = 0$$

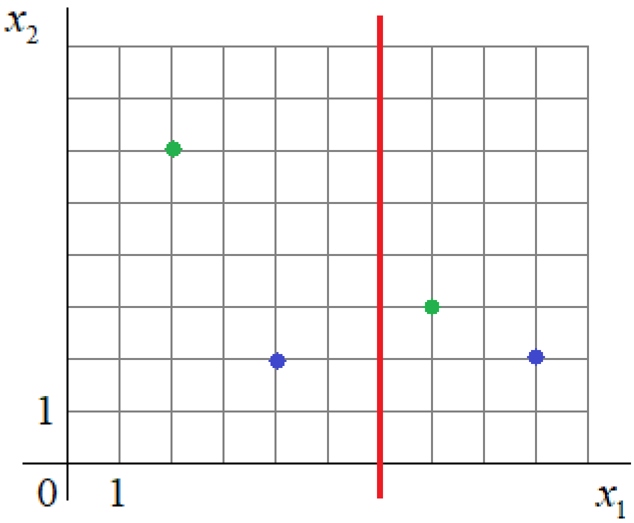
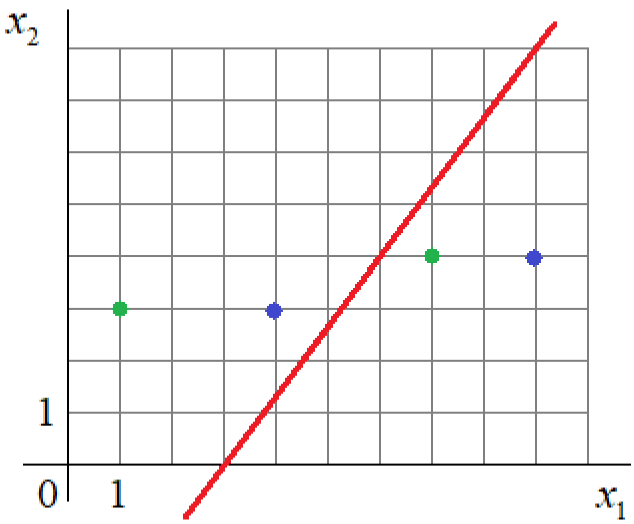
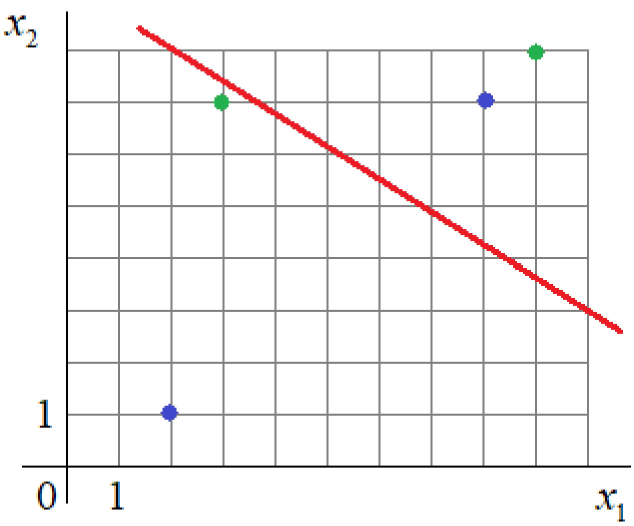
2. Вычислить отступы (margin) для зеленых точек (с меткой класса +1) и синих точек (с меткой класса -1). Напомню, что отступ вычисляется по формуле:

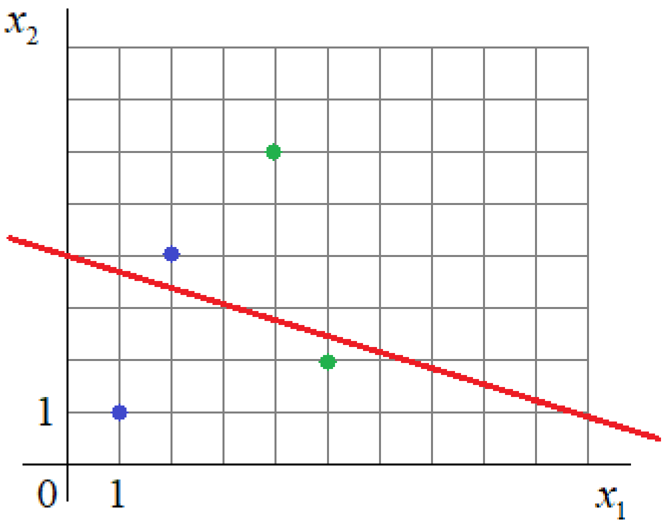
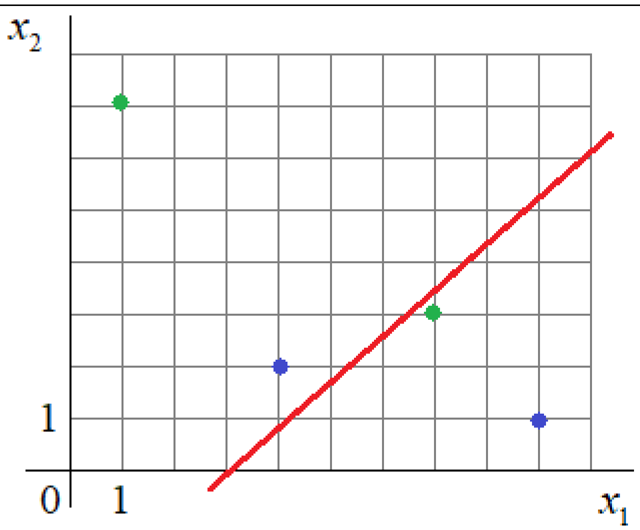
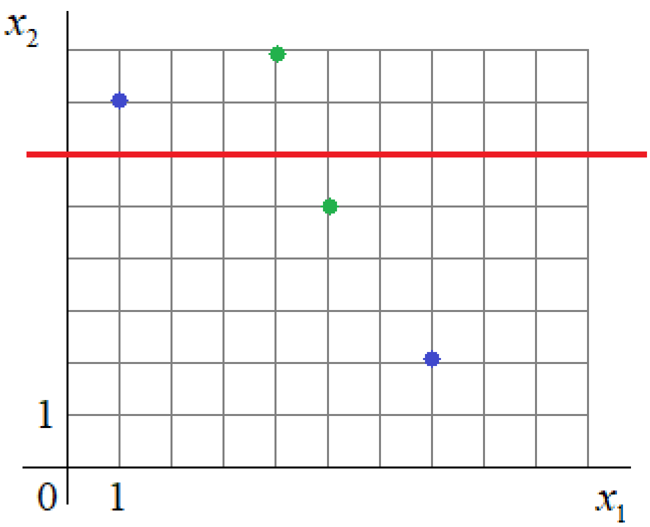
$$M_i = y_i \cdot \langle \omega, x_i \rangle, \quad i = 1, 2, 3, 4,$$

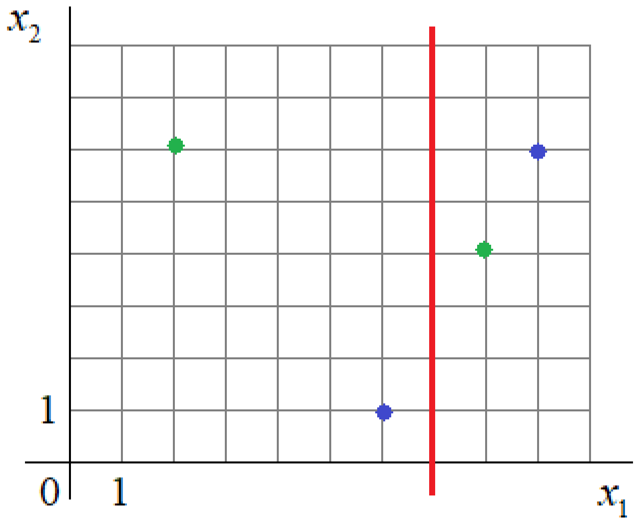
где $y_i \in \{-1; +1\}$ - метка класса образа (точки) x_i; $\langle \omega, x_i \rangle$ - скалярное произведение векторов.

Вектор ω должен быть подобран так, чтобы для наиболее удаленных точек от разделяющей линии отступ был положительным, а для ближних – отрицательным.

Вариант	Графики
1	
2	
3	

4	
5	
6	

7	
8	
9	

10	

Содержание отчета

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Расчеты для весов разделяющей линии.
3. Расчеты для отступов.
4. Выводы по полученным результатам.

Лабораторная работа №2

Обучение линейного алгоритма бинарной классификации образов с помощью градиентного алгоритма

Цель работы: научиться реализовывать алгоритм градиентного спуска для задачи обучения линейной модели бинарной классификации образов.

Теоретический материал

Теория для выполнения лабораторной работы доступна на странице сайта:

https://proproprogs.ru/ml

в разделах:

- Линейная модель. Понятие переобучения
- Способы оценивания степени переобучения моделей
- Уравнение гиперплоскости в задачах бинарной классификации
- Решение простой задачи бинарной классификации
- Функции потерь в задачах линейной бинарной классификации
- Стохастический градиентный спуск SGD и алгоритм SAG
- Пример использования SGD при бинарной классификации образов

а также в соответствующих видеоматериалах, размещенных на странице сайта:

tk.ulstu.ru/video.php?id=3

Задания на лабораторную работу (по вариантам)

В файле iris_data.py даны обучающие выборки (по вариантам) для обучения линейного алгоритма бинарной классификации образов:

http://tk.ulstu.ru/files/iris_data.py

Модель линейного алгоритма должна иметь вид:

$$a(x) = sign(\langle \omega, x \rangle),$$

где $\omega = [\omega_0, \omega_1, \omega_2]^T$ - весовые коэффициенты модели (определяют ориентацию разделяющей линии); $x = [1, x_1, x_2]^T$ - образ обучающей выборки; $sign(v) = \begin{cases} -1, & v < 0 \\ +1, & v > 0 \end{cases}$ - знаковая функция.

То есть, метки классов принимают значения $Y = \{-1; +1\}$.

Ваша задача выполнить обучение линейной модели $a(x)$ (найти значения вектора весовых коэффициентов ω) с помощью градиентного алгоритма (программы, написанной на языке Python), который должен минимизировать величину эмпирического риска:

$$Q\left(X^l\right) = \sum_{i=1}^{l} \left[y_i \neq a\left(x_i\right) \right] \to \min_{\omega}$$

где $[\bullet]$ - нотация Айверсона (квадратные скобки возвращают 1, если условие в скобках истинно, и 0 – в противном случае). То есть, эмпирический риск показывает число неверных классификаций.

Так как градиентный алгоритм может минимизировать только гладкие, дифференцируемые функции, то величину $Q\left(X^l\right)$ следует сверху ограничить именно таким функционалом:

$$Q\left(X^l\right) \le \tilde{Q}\left(X^l\right) = \sum_{i=1}^{l} L\left(a\left(x_i\right), y_i\right) \to \min_{\omega}$$

где $L\left(a\left(x_i\right), y_i\right) = L\left(M_i\right)$ - выбранная функция потерь (здесь $M_i = y_i \cdot \langle \omega, x_i \rangle$ - отступ).

Функция потерь (также, как и набор обучающих данных) определяется вариантом.

Вариант	Функция потерь для реализации градиентного алгоритма	Производная функции потерь
1	$L(M) = \log_2\left(1 + e^{-M}\right)$ - логарифмическая	$\frac{\partial L(M)}{\partial \omega} = -\frac{e^{-M} \cdot x^T \cdot y}{\left(1 + e^{-M}\right) \cdot \ln 2}$
2	$Q(M) = (1 - M)^2$ - квадратичная	$\frac{\partial Q(M)}{\partial \omega} = -2 \cdot \left(1 - \omega^T \cdot x \cdot y\right) \cdot x^T \cdot y$
3	$S(M) = 2 \cdot \left(1 + e^{M}\right)^{-1}$ - сигмоидная	$\frac{\partial S(M)}{\partial \omega} = -\frac{2 \cdot e^{M} \cdot x^T \cdot y}{\left(1 + e^{M}\right)^2}$
4	$E(M) = e^{-M}$ - экспоненциальная	$\frac{\partial E(M)}{\partial \omega} = -e^{-M} \cdot x^T \cdot y$
5	$L(M) = \log_2\left(1 + e^{-M}\right)$ - логарифмическая	$\frac{\partial L(M)}{\partial \omega} = -\frac{e^{-M} \cdot x^T \cdot y}{\left(1 + e^{-M}\right) \cdot \ln 2}$
6	$Q(M) = (1 - M)^2$ - квадратичная	$\frac{\partial Q(M)}{\partial \omega} = -2 \cdot \left(1 - \omega^T \cdot x \cdot y\right) \cdot x^T \cdot y$
7	$S(M) = 2 \cdot \left(1 + e^{M}\right)^{-1}$ - сигмоидная	$\frac{\partial S(M)}{\partial \omega} = -\frac{2 \cdot e^{M} \cdot x^T \cdot y}{\left(1 + e^{M}\right)^2}$

8	$E(M) = e^{-M}$ - экспоненциальная	$\frac{\partial E(M)}{\partial \omega} = -e^{-M} \cdot x^T \cdot y$
9	$L(M) = \log_2\left(1 + e^{-M}\right)$ - логарифмическая	$\frac{\partial L(M)}{\partial \omega} = -\frac{e^{-M} \cdot x^T \cdot y}{\left(1 + e^{-M}\right) \cdot \ln 2}$
10	$Q(M) = \left(1 - M\right)^2$ - квадратичная	$\frac{\partial Q(M)}{\partial \omega} = -2 \cdot \left(1 - \omega^T \cdot x \cdot y\right) \cdot x^T \cdot y$

В качестве начальных значений весовых коэффициентов можно взять следующие:

$$\omega_0 = 0;\ \omega_1 = 0;\ \omega_2 = 1$$

Шаг в градиентном алгоритме для коэффициента ω_0 целесообразно выбрать побольше, а для коэффициентов ω_1, ω_2 - поменьше.

Содержание отчета

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Математические выкладки, необходимые для реализации алгоритма обучения.
3. Текст программы обучения линейной модели с использованием градиентного алгоритма на языке Python.
4. Результаты работы программы в виде графика множества точек обучающей выборки (каждый класс точек должен быть представлен разными маркерами и цветами) и полученной разделяющей линии.
5. Выводы по полученным результатам.

Лабораторная работа №3

Исследование работы L2-регуляризатора в задачах регрессии

Цель работы: изучить особенности работы L2-регуляризатора на примере задачи аппроксимации функции линейной моделью.

Теоретический материал

Теория для выполнения лабораторной работы доступна на странице сайта:

https://proproprogs.ru/ml

в разделах:

- Функции потерь в задачах линейной бинарной классификации
- L2-регуляризатор. Математическое обоснование и пример работы
- L1-регуляризатор. Отличия между L1- и L2-регуляризаторами
- Вероятностный взгляд на L1 и L2-регуляризаторы

а также в соответствующих видеоматериалах, размещенных на странице сайта:

tk.ulstu.ru/video.php?id=3

Задания на лабораторную работу (по вариантам)

1. Вам необходимо аппроксимировать (описать) функцию своего варианта с помощью линейной модели:

$$a(x) = \omega_0 + \sum_{i=1}^{13} \omega_i x^i ,$$

то есть, полиномом 13-й степени. Здесь $\{\omega_i\}$ - весовые коэффициенты, которые требуется найти с помощью градиентного алгоритма по обучающему набору данных.

2. Обучающую выборку следует составить из всех четных индексов сгенерированных значений функции:

$$X^l : \left\{ \left(x_{2i}, y = f\left(x_{2i}\right) \right)_{i=0}^{l\ //\ 2} \right\}$$

То есть, сначала формируется первое значение x_0 с целевым значением $y_0 = f\left(x_0\right)$, затем, второе: $\left(x_2, y_2 = f\left(x_2\right)\right)$ и так пока не дойдем до конца диапазона.

3. После этого, вычислите значения коэффициентов вектора ω для квадратической функции потерь (в задачах регрессии, обычно, используют именно такую функцию потерь), которые минимизируют эмпирический риск:

$$Q\left(X^l\right)=\frac{1}{2}\sum_{i=1}^{l}\left(y_i-a\left(x_i\right)\right)^2\to\min_{\omega}$$

Коэффициенты вычисляются по формуле:

$$\omega_*=\left(X^T\cdot X\right)^{-1}\cdot X^T\cdot Y$$

где X - входные векторы обучающей выборки; Y - вектор (или матрица) целевых значений обучающей выборки:

$$X=\begin{bmatrix}x_1\\x_2\\\dots\\x_l\end{bmatrix}=\begin{bmatrix}x_{11}&x_{12}&\dots&x_{1n}\\x_{21}&x_{22}&\dots&x_{2n}\\\dots&\dots&\dots&\dots\\x_{l1}&x_{l2}&\dots&x_{ln}\end{bmatrix}\qquad Y=\begin{bmatrix}y_1\\y_2\\\dots\\y_l\end{bmatrix}$$

4. Вычислите прогнозы функции с помощью полученной модели $a\left(x\right)$ для всего диапазона значений. (В отсчетах, не участвующих в выборке, значения модели должны сильно расходиться с целевыми.)

5. Вычислите коэффициенты вектора ω с L2 регуляризатором по формуле:

$$\omega_*=\left(X^T\cdot X+\lambda\cdot I\right)^{-1}\cdot X^T\cdot Y$$

где $\lambda>0$ - коэффициент регуляризации; $I_{n\times n}$ - единичная матрица.

6. Для новой модели $a\left(x\right)$ повторите вычисление прогнозов функции для всего диапазона значений.

P.S. Все программы реализовать на языке Python с использованием пакетов NumPy и Matplotlib.

Вариант	Функция для исследования L1 и L2-регуляризаторов.
1	$y\left(x\right)=\dfrac{1}{10+x^3},\quad x\in[0;10;0,1]$
2	$y\left(x\right)=-x^4+100x^2+x,\quad x\in[0;10;0,1]$
3	$y\left(x\right)=x^3-10x^2+x,\quad x\in[0;10;0,1]$
4	$y\left(x\right)=0,1x^5-100x^3+700x^2,\quad x\in[0;10;0,1]$
5	$y\left(x\right)=-0,1x^5+5x^4-700x^2,\quad x\in[0;10;0,1]$
6	$y\left(x\right)=\dfrac{1}{10+x^2},\quad x\in[0;10;0,1]$

7	$y(x)=x^4-10x^3+20x^2-100x, \quad x\in[0;10;0,1]$
8	$y(x)=-0,01x^6-2x^5+200x^3, \quad x\in[0;10;0,1]$
9	$y(x)=-0,01x^6+4x^4-200x^2, \quad x\in[0;10;0,1]$
10	$y(x)=x^3-5x^2-100x+200, \quad x\in[0;10;0,1]$

Содержание отчета

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Математические выкладки для реализации алгоритмов.
3. Тексты программ с результатами их работы.
4. Выводы по полученным результатам.

Лабораторная работа №4

Реализация наивного байесовского классификатора

Цель работы: научиться строить наивный байесовский классификатор и с его помощью выполнять бинарную классификацию образов.

Теоретический материал

Теория для выполнения лабораторной работы доступна на странице сайта:

https://proproprogs.ru/ml

в разделах:

- Логистическая регрессия. Вероятностный взгляд на машинное обучение
- Вероятностный взгляд на L1 и L2-регуляризаторы
- Формула Байеса при решении конкретных задач
- Байесовский вывод. Наивная байесовская классификация
- Гауссовский байесовский классификатор
- Линейный дискриминант Фишера

а также в соответствующих видеоматериалах, размещенных на странице сайта:

tk.ulstu.ru/video.php?id=3

Задания на лабораторную работу (по вариантам)

1. Необходимо построить (реализовать на языке Python) наивный байесовский классификатор на основе, следующих данных обучающей выборки (для своего варианта):

http://tk.ulstu.ru/files/iris_data.py

Полагать, что признаки независимы и распределены по гауссовскому закону (нормальной плотности распределения вероятностей).

2. Для данной обучающей выборки подсчитать число и процент неверных классификаций.

3. Отобразить обучающую выборку в виде графика точек на плоскости (объекты разных классов должны быть иметь разные маркеры и цвет).

Содержание отчета

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Все расчеты, связанные с построением наивного байесовского классификатора.
3. Программа, реализующая наивный байесовский классификатор.
4. Графики и результаты работы программы.
5. Выводы по полученным результатам.

Лабораторная работа №5

Реализация алгоритма метода опорных векторов для задачи бинарной классификации

Цель работы: реализовать метод опорных векторов (SVM) для задачи бинарной классификации.

Теоретический материал

Теория для выполнения лабораторной работы доступна на странице сайта:

https://proproprogs.ru/ml

в разделах:

- Введение в метод опорных векторов (SVM)
- Реализация метода опорных векторов (SVM)
- Метод опорных векторов (SVM) с нелинейными ядрами

а также в соответствующих видеоматериалах, размещенных на странице сайта:

tk.ulstu.ru/video.php?id=3

Задания на лабораторную работу (по вариантам)

1. Необходимо построить (реализовать на языке Python с применением пакета Scikit-Learn) линейный вариант метода опорных векторов, для следующих данных обучающей выборки (для своего варианта):

http://tk.ulstu.ru/files/iris_data.py

2. Для данной обучающей выборки подсчитать число и процент неверных классификаций.

3. Отобразить обучающую выборку в виде графика точек на плоскости (объекты разных классов должны быть иметь разные маркеры и цвет), а также полученную (в результате обучения) разделяющую линию.

Содержание отчета

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Программа, реализующая метод опорных векторов.

3. Графики и результаты работы программы.
4. Выводы по полученным результатам.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. — СПб.: Питер, 2018. — 480 с.
2. Рашид, Тарик. Создаем нейронную сеть.: Пер. с англ. — СПб.: ООО «Альфа-книга», 2017. — 272 с.: ил.
3. Хайкин, Саймон. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.: ил.
4. Васильев К.К., Оптимальная обработка сигналов в дискретном времени: Учебн. пособие. – М.: Радиотехника, 2016. – 288 с.: ил.
5. Christopher M. Bishop, Neural Networks for Pattern Recognition. – Clarendon Press Oxford, 1995 - 498 с.
6. Harrison Kinsley, Neural Networks from Scratch in Python - 666 с.