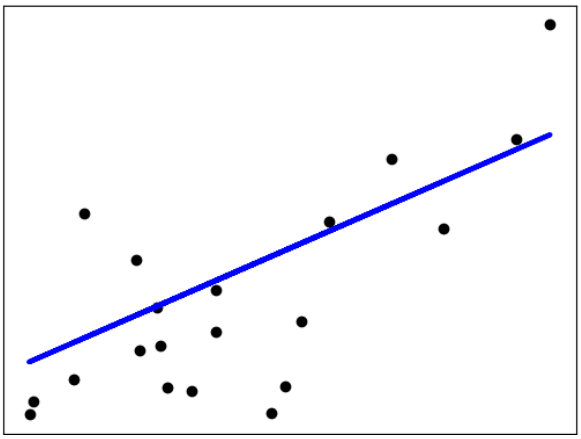
Задание 4. Линейные методы регрессии и классификации.
Метод стохастического градиента
(подробней см. в [1, стр. 51–62], [2], [3, стр. 49–84], [4])
(документация: 1.1. Linear Models — scikit-learn 1.1.3 documentation
1.5. Stochastic Gradient Descent — scikit-learn 1.1.3 documentation
sklearn.linear_model.LinearRegression — scikit-learn 1.1.3 documentation
sklearn.linear_model.Perceptron — scikit-learn 1.1.3 documentation)

I. Линейные методы регрессии и классификации

Пусть есть обучающая выборка $X = (x_i, y_i)_{i=1}^{l}$, где $x_i \in R^n$ – набор из l объектов, у каждого такого объекта x_i есть n признаков $f_j(x_i)$ $(j = 1..n)$, и $y_i \in R$ – известные ответы обучающей выборки.



В линейной модели **восстановления регрессии** решение ищется в виде линейной функции: суммы всех признаков $f_j(x)$ с коэффициентами w_j (взвешенная сумма всех признаков):

$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^{n} f_j(x) w_j.$$

Или с дополнительным коэффициентом w_0:

$$a(x, w) = w_0 + \langle x, w \rangle = w_0 + \sum_{j=1}^{n} f_j(x) w_j,$$

тогда функция $a(0, w) = w_0$, не будет всегда проходить через начало координат.

Замечание. Можно добавлять признаки к задаче, например, кроме признака f_j добавить в матрицу «объектов–признаков» еще признак f_j^2 и тогда будем искать решение в виде полинома второй степени по этому признаку.

Если взять квадратичную функцию потерь для каждой i-й точки:

$$L(a, y) = (a_i - y_i)^2$$

и просуммировать ее, получим функционал качества:

$$Q(w) = \sum_{i=1}^{l} (a(x_i, w) - y_i)^2 \to \min_w$$

такой же как в методе наименьших квадратов. Т.к. функция $Q(w)$ дифференцируема по w, в методе наименьших квадратов минимум этой функции ищется из условия равенства нулю всех производных: $\frac{\partial Q(w)}{\partial w_k} = 0$.

Подставим $a(x, w)$:

$$Q(w) = \sum_{i=1}^{l} \left(\sum_{j=1}^{n} f_j(x_i) w_j - y_i \right)^2$$

и продифференцируем

$$\frac{1}{2} \frac{\partial Q(w)}{\partial w_k} = \sum_{i=1}^{l} \sum_{j=1}^{n} (f_j(x_i) w_j - y_i) f_k(x_i) = 0, \qquad k = 1..n.$$

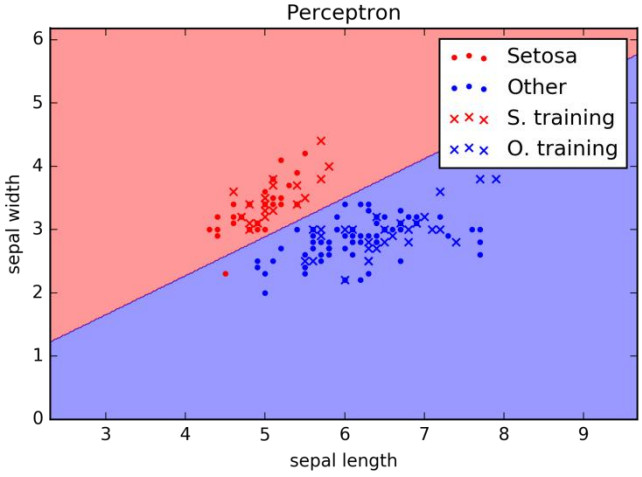
И задача сводится к решению СЛАУ n-го порядка:

$$\sum_{j=1}^{n} w_j \sum_{i=1}^{l} f_j(x_i)\, f_k(x_i) = \sum_{i=1}^{l} y_i f_k(x_i)\,, \qquad k = 1..n.$$

Из которой находятся неизвестные коэффициенты w_j.

Но в задачах машинного обучения метод наименьших квадратов в чистом виде часто не дает хорошего результата, т.к. матрица может быть большого порядка и плохо обусловлена или вообще вырождена из-за линейной зависимости признаков в задаче. Обычно применяют модификации метода или другие методы.

В случае **задачи классификации** постановка задачи похожа. У нас также есть обучающая выборка $X = (x_i, y_i)_{i=1}^{l}$, где $x_i \in R^n$ – набор из l объектов, у каждого такого объекта x_i есть n признаков $f_j(x_i)$ $(j = 1..n)$, но известные ответы обучающей выборки – это один из двух классов $y_i \in \{-1, +1\}$.



В линейной модели **классификации** решение ищется в виде функции знака числа (sign) от суммы всех признаков $f_j(x)$ с коэффициентами w_j:

$$a(x, w) = \text{sign}\,\langle x, w\rangle = \text{sign}\left(\sum_{j=1}^{n} f_j(x) w_j\right).$$

Геометрический смысл состоит в том, что если w направляющий вектор к разделяющей два класса гиперплоскости, то если x лежит по одну сторону с вектором w от гиперплоскости, то скалярное произведение будет больше нуля и точка x попадает в класс +1. Если же x лежит по другую сторону от вектора w от гиперплоскости, то знак скалярного произведения меньше нуля и точка попадает в класс –1.

В этой задаче определить функцию потерь для каждой точки x_i можно так:

$$L(a, y) = [\langle x_i, w\rangle y_i < 0]$$

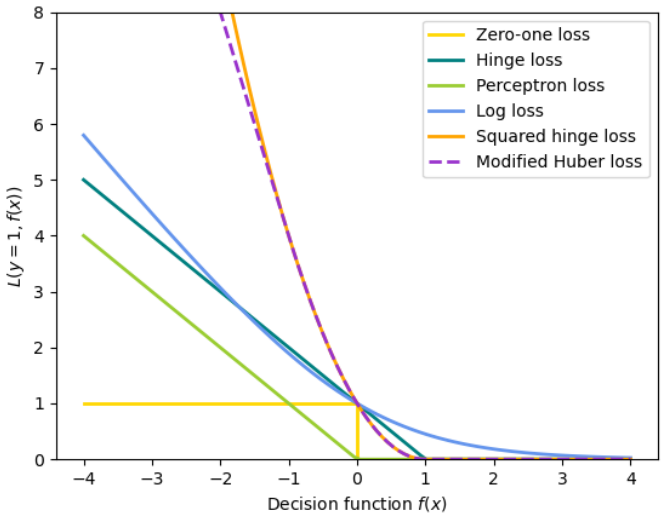
Т.е. если знак предсказания $\langle x_i, w\rangle$ совпадает с известным ответом y_i, то $\langle x_i, w\rangle y_i > 0$ и выражение $\langle x_i, w\rangle y_i < 0$ неверно и тогда $L(a, y) = [\text{False}] = 0$. Если же знак предсказания $\langle x_i, w\rangle$ противоположен знаку известного ответа y_i, то $\langle x_i, w\rangle y_i < 0$ и тогда $L(a, y) = [\text{True}] = 1$.

Такая функция выдает только информацию правильно или неправильно был предсказан ответ и с ее помощью можно посчитать количество неправильных ответов. У нее есть недостаток, т.к. она не говорит о величине ошибки, функционал качества с такой функцией потерь будет кусочно-постоянным, его нельзя продифференцировать и приравнять нулю производные как в методе наименьших квадратов.

Вместо пороговой функции потерь используют ее непрерывные аппроксимации:

$$L(a, y) = [\langle x_i, w\rangle y_i < 0] \leq L(\langle x_i, w\rangle y_i),$$

где $M_i(w) = \langle x_i, w\rangle y_i$ – **отступ** (margin) объекта x_i. При положительном отступе (нет ошибки) пороговая функция потерь давала 0, при отрицательном отступе (ошибка есть) она давала 1 в функционал качества. На рисунке она изображена желтым цветом. Также на рисунке изображено несколько используемых для нее аппроксимаций (функций потерь).



Отступ несет в себе информацию не только о наличии ошибки, по нему можно еще судить и о расстоянии точки от разделяющей гиперплоскости: чем больше отступ по модулю, тем дальше точка находится от гиперплоскости. Для точек находящихся на гиперплоскости отступ равен 0.

Поэтому можно взять в качестве функции потерь непрерывную невозрастающую функцию от отступа и тогда за большие ошибки будет больший штраф. И просуммировав ее для всех точек, получим функционал качества:

$$Q(w) = \sum_{i=1}^{l} L(\langle x_i, w\rangle y_i) \to \min_w$$

И также как и в задаче регрессии свести задачу к поиску минимума непрерывного (или даже гладкого) функционала качества. К нему можно применять различные численные методы поиска минимума.

II. Метод стохастического градиента

Задачу поиска минимума функционала можно решать, например, с помощью **метода градиентного спуска**. В этом методе выбирается начальная точка $w^{(0)}$, в ней считается значение градиента $\nabla Q\left(w^{(0)}\right)$, который показывает направление роста $Q(w)$ в точке $w^{(0)}$. И делается шаг в противоположном градиенту направлении (направлении убывания):

$$w^{(1)} = w^{(0)} - h\nabla Q\left(w^{(0)}\right), \qquad \text{где вектор} \qquad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=1}^{n}.$$

Шаг h, называется темпом обучения, можно выбирать разными способами. И, продолжая этот процесс, на каждом шаге будем получать все меньшие значения функционала качества $Q(w)$, придем к некоторому значению w^*. Такой способ поиска минимума теоретически приводит к поиску локального минимума функ-

ции $Q(w)$. Если, у нее есть несколько локальных минимумов, может быть найдено не оптимальное значение (не глобальный минимум).

Здесь требуется вычислять градиент на каждой итерации метода градиентного спуска, который в задачах машинного обучения может состоять из большого количества слагаемых. Идея ускорения сходимости: не вычислять весь градиент на каждой итерации, а брать по одной случайной точке (x_i, y_i) и сразу обновлять вектор весов w. Называется методом **стохастического градиентного спуска (Stochastic Gradient Descent, SGD)**.

Алгоритм метода стохастического градиента (SGD):

Вход: выборка X^l, темп обучения h, темп забывания λ.

Выход: вектор весов w.

1. Инициализировать w;
2. Инициализировать оценку функционала:

$$Q(w) = \frac{1}{l}\sum_{i=1}^{l} L(\langle x_i, w\rangle y_i);$$

3. **Повторять:**
4. Выбрать объект x_i из X^l случайным образом;
5. Вычислить потерю: $\varepsilon_i = L_i(w) = L(\langle x_i, w\rangle y_i)$;
6. Сделать градиентный шаг: $w = w - h\nabla L_i(w)$;
   (оптимизируем веса w для одной случайной точки x_i);
7. Оценить функционал: $\bar{Q} = (1-\lambda)\bar{Q} + \lambda\varepsilon_i$;
   (примерно оцениваем значение функционала, добавляя в него оценку для последней точки с коэффициентом λ и уменьшая весь функционал на такой же коэффициент («забывая» старые значения); если $\lambda = \frac{1}{l}$, то в формуле $\bar{Q} = \left(1 - \frac{1}{l}\right)\bar{Q} + \frac{1}{l}\varepsilon_i$ как бы убрали одно значение, равное среднему от всех и добавили одно, посчитанное последним);
8. **пока** значение $\bar{Q}$ и/или веса w не сойдутся.

Замечание 1. У этого метода есть оптимизации, например, метод стохастического усредненного градиента, диагональный метод Левенберга–Марквардта и др.

Существует несколько способов выбора начального вектора w так, чтобы метод не сошелся к не самому лучшему локальному минимуму, например, мультистарт: пробовать для нескольких разных начальных w и сравнивать результаты. Существует несколько способов выбора шага h.

Из-за линейной зависимости (или почти линейной зависимости) признаков у задачи может быть не одно оптимальное решение, а целое семейство (прямая или

гиперплоскость) решений (минимумов функционала или точек с почти одинаковыми значениями функционала). Метод становится неустойчивым и может сходиться к любой из этих точек. И на его ответ могут сильно влиять отдельные точки обучающей выборки. Возникает переобучение, когда результаты для тренировочной и тестовой выборках сильно различаются. Признаком такого переобучения являются большие по модулю веса w_j разных знаков. Для борьбы с увеличением весов применяют **регуляризацию**, которая не дает расти w_j, например, так:

$$Q(w) = \frac{1}{l}\sum_{i=1}^{l} L(\langle x_i, w\rangle y_i) + \frac{\tau}{2}\sum_{j=1}^{n} w_j^2 \to \min_w$$

В линейных методах классификации и регрессии, также как и в методе ближайших соседей, **применяют масштабирование** признаков, которое повышает качество предсказаний.

Алгоритм обучения нейрона, предложенный Розенблатом (1957 г., правило обучения **персептрона**, см. [3, стр. 49–84]), является частным случаем метода стохастического градиента с функцией потерь $L(x) = \{-x, \text{при } x < 0; 0, \text{при } x \geq 0\}$ и без регуляризации.

Достоинства метода стохастического градиента: легко реализуется; легко обобщается на нелинейные модели $g(x, w)$ и любые функции потерь $L(w)$; допускает потоковое обучение, когда не все данные загружаются; подходит для больших данных, дает неплохое решение, даже не обработав всю выборку.
Недостатки: возможна расходимость или медленная сходимость; застревание в локальных минимумах; много параметров, которые нужно настраивать; возможно переобучение из-за линейной зависимости признаков.

III. В библиотеке **Scikit-Learn** реализован метод восстановления регрессии линейной модели с использованием метода наименьших квадратов LinearRegression (sklearn.linear_model.LinearRegression — scikit-learn 1.1.3 documentation). Также реализованы методы стохастического градиента для задачи классификации SGDClassifier и регрессии SGDRegressor (1.5. Stochastic Gradient Descent — scikit-learn 1.1.3 documentation).

В методе стохастического градиента для задачи классификации SGDClassifier задаются:

- loss = ‘hinge’ – функция потерь (loss function), по умолчанию задан метод опорных векторов;
- penalty = ‘l2’ – регуляризация (или штраф) суммой квадратов;
- alpha = 0.0001 – коэффициент регуляризации;

- l1_ratio – коэффициент регуляризации (используется, если penalty = ‘elasticnet’);
- learning_rate = ’optimal’ – выбор шага (темпа обучения) в формуле стохастического градиента, по умолчанию (’optimal’) уменьшается по заданной формуле;
- eta0 = 0.0 – начальный шаг (темпа обучения) в формуле стохастического градиента, по умолчанию не используется;
- и др.

Метод персептрон Perceptron (sklearn.linear_model.Perceptron — scikit-learn 1.1.3 documentation) совпадает с методом SGDClassifier с параметрами:

- loss = "perceptron" – функция потерь персептрон;
- penalty = None – нет регуляризации;
- learning_rate = "constant" – шаг (темпа обучения) постоянный;
- eta0 = 1 – величина шага.

Задания

I. Прочитайте пункт “Линейные методы регрессии и классификации” и рекомендованную литературу к нему и ответьте на вопросы:

1) В каком виде ищется решение в задаче восстановления регрессии? На что влияет коэффициент w_0?
2) Какой вид имеет функция потерь в методе наименьших квадратов?
3) Как в методе наименьших квадратов находится минимум?
4) В каком виде ищется решение в задаче классификации? Какой геометрический смысл вектора коэффициентов w?
5) В чем недостаток использования пороговой функции потерь?
6) Что такое отступ? Когда он бывает положительным и отрицательным? Какую еще информацию можно узнать с помощью него?
7) Какие бывают функции потерь?

II. Прочитайте пункт “Метод стохастического градиента” и рекомендованную литературу к нему и ответьте на вопросы:

1) В чем отличия метода стохастического градиента от метода градиентного спуска?
2) Зачем делается регуляризация?
3) Какие параметры можно задавать в методе?

4) Какие преимущества и недостатки метода стохастического градиента?

III. Выполните задание из в файла "statement-linear.pdf ".

Литература

[1] К.В. Воронцов Математические методы обучения по прецедентам (теория обучения машин). 141 с. (Voron-ML-1.pdf)

[2] Машинное обучение (курс лекций, К.В.Воронцов): Линейный классификатор (machinelearning.ru), Метод стохастического градиента (machinelearning.ru) (“ Линейные методы классификации и регрессии: метод стохастического градиента”: https://www.youtube.com/watch?v=thrPR77K-os)

[3] Рашка Себастьян, Мирджалили Вахид Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2, 3-е изд.: Пер. с англ. СПб. : ООО "Диалектика", 2020. 848 с.

[4] Андреас Мюллер, Сара Гвидо Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. 393 с.