Exercise 3
Statistical Learning Theory (Lab)

Release: May 17, 2019 Submit: May 30, 2019.

This exercise is meant to familiarize you with the complete pipeline of solving a machine learning problem. You need to obtain and pre-process the data, develop, implement and train a machine learning model and evaluate it by splitting the data into a training and validation set.

Notes:

- Ask for help, if needed.
- Implement does **not** mean import!
- Most of the required calculations have already been done in the lecture notes.
- Submit **working** code, e.g. a Jupyter Notebook.

1 Logistic Regression

In statistics, the model given as

$$p(y=1|\theta) = \sigma(x^T\theta), \qquad p(y=0|\theta) = 1 - p(y=1|\theta), \qquad x, \theta \in \mathbb{R}^n \tag{1}$$

with σ being the *logistic sigmoid* function, defined as:

$$\sigma(a) = \frac{1}{1 + exp(-a)}$$

is known as *logistic regression* and can be interpreted as a generalization of the standard linear model. It should be emphasized, that *logistic regression* is a model for classification, rather than regression.

Problem 1. Develop a logistic regression classifier. As stated in [Dom12], each machine learning problem can be understood as a combination of the three components: problem representation, evaluation and optimization.

(a) **Representation:** Implement the logistic classifier from eq. (1) that predicts the classes $\hat{Y} \in \{0,1\}^m$, given input features $X \in \mathbb{R}^{(m,n)}$ and a parameter vector $\theta \in \mathbb{R}^n$.

(b) **Evaluation:** As an evaluation function (a.k.a *loss function*, *error function* or *objective function*), use binary cross entropy (see lecture notes), given as:

$$E(\theta) = -\frac{1}{m}\sum_{i=1}^{m} Y_i * \log(p(Y_i = 1|\theta)) + (1 - Y_i) * \log(1 - p(Y_i|\theta) = 1)$$

Implement a function, that computes the binary cross entropy, given the true labels Y and the predictions $\hat{Y}$. Often it is convenient to have multiple metrics at hand. In classification problems, the accuracy of a prediction is defined as:

$$ACC = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Implement a function that calculates the accuracy given the true labels Y and the predictions $\hat{Y}$.

(c) **Optimization:** Given the loss function $E(\theta)$. Minimizing this function with respect to the parameters θ can not be done in closed form. Therefore we use an iterative approach that starts with an initial estimate for θ and approaches the solution at each iteration step. The most simple approach is to take the gradient $\nabla E(\theta)$ of $E(\theta)$ with respect to θ and walk into direction of the negative gradient. This method is called gradient-descent [Gra]. See alg. (1).

Implement the gradient descent method for the logistic regression problem. As default parameters, you can use $\eta = 0.1$, $\lambda = 1000$, $\mu = 1e-4$.

(d) **Fit the data:** Use your implementation to fit a model to the data given in the file *data.csv*. Each row represents one sample point. The first two columns contain the features and the third column contains the labels. The data is visualized in fig. (1).

(e) **Visualization:** Visualize the data, colorize the true labels (see fig. (1)) and highlight the mispredicted data points.
What is the final value of the loss function and which accuracy did you achieve by training your model on the whole data set?

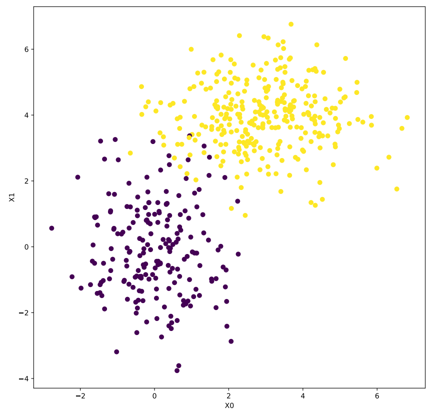

Figure 1: Scatter plot of the data in. Each data point belongs to one of two classes, highlihted in yellow and purple.

Algorithm 1: Gradient Descent

Input: Design matrix X, label vector Y learning rate η,
tolerance criterium μ, maximum number of iterations λ
Output: Model parameters θ
$i \leftarrow 0$;
$\theta \leftarrow (0, \ldots, 0)$;
repeat
  $\theta_{old} \leftarrow \theta$;
  $\theta \leftarrow \theta - \eta * \nabla E(\theta | X, Y)$;
  $i \leftarrow i + 1$;
until $|E(\theta|X,Y) - E(\theta_{old}|X,Y)| < \mu \; or \; i > \lambda$;

2 SPAM Classification by using a Logistic Regression classifier

Problem 2. The *UCI SMS Spam Collection* data set [UCI] is a public collection of 5574 binary labeled SMS messages, that have been collected from mobile spam research. Use you implementation of the logistic classifier to fit a model that classifies messages into *spam* or *ham* messages. For this, download the dataset, split the dataset into a train and validation set. Train your model on the train set and validate it on the validation set. Since your data is in a raw format you need to preprocess it.

Hints

- A common way to vectorize text documents is to use token frequencies. This is called *bag-of-words* representation. You can checkout the *sklearn* text tutorial for getting started [UCI].
- Make sure your functions are implemented correctly.

Have fun!

References

[Dom12] DOMINGOS, Pedro: A Few Useful Things to Know About Machine Learning. In: *Commun. ACM* 55 (2012), Nr. 10, S. 78–87

[Gra] GRADIENT DESCENT: *Gradient descent — Wikipedia, The Free Encyclopedia.* `https://en.wikipedia.org/wiki/Gradient_descent`

[UCI] UCI: *SMS Spam Collection.* `https://archive.ics.uci.edu/ml/datasets/sms+spam+collection`