Machine Learning in Economics and Finance 1

Homework 2

July 21, 2020

- This homework assignment covers Lecture 2 – "Linear Regression and Logistic Regression"
- This homework is due 11 PM, Sunday, 26 July, 2020. The Google Form for submission will be sent out later.

Problem 1. *(10 points)*
In the "New York City Taxi Fare Prediction" Competition on Kaggle (https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/data), competitors have to predict the cost (in USD) of a taxi ride in New York City given the following features:

- *pickup_datetime* – timestamp value indicating when the taxi ride started.
- *pickup_longitude* – float for longitude coordinate of where the taxi ride started.
- *pickup_latitude* – float for latitude coordinate of where the taxi ride started.
- *dropoff_longitude* – float for longitude coordinate of where the taxi ride ended.
- *dropoff_latitude* – float for latitude coordinate of where the taxi ride ended.
- *passenger_count* – integer indicating the number of passengers in the taxi ride.

Competitors are given a training dataset (*train.csv*) with input features and target *fare_amount* values. They will then have to predict the *fare_amount* for each row of input features in a test set (*test.csv*). Using Tom M. Mitchell's definition of Machine Learning discussed in Lecture "Introduction to Machine Learning" (for Parts (a) and (b)),
(a) Describe the experience E and the class of tasks T of the algorithms used to solve this problem.
(b) Propose a performance measure P that we can use to rank the competitors' submissions.
(c) Is this problem a supervised learning one or an unsupervised learning one? Justify your answer.
(d) Is this problem a classification or a regression problem? Justify your answer.

Problem 2. *(10 points)*
In a homework at the Machine Learning class, Toan uses Logistic Regression to classify customers of a Consumer Finance Company (CFC) into two categories: Low-risk (Negative) and High-risk (Positive). Comparing the output of his model with the loan performance data of 1000 customers,Toan ends up with the following confusion matrix (For a description, see, for example https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/):

n=1000	Predicted Low-risk	Predicted High-risk
Actual Low-risk	850	50
Actual High-risk	20	80

(a) Calculate True Positive rate, False Positive rate, True Negative rate, and False Negative rate.
(b) Discuss the costs (to the CFC) of a False Positive and a False Negative.
(c) Calculate the Accuracy, Precision, Recall, and F_1 score of this classifier.

For references, see also

- https://en.wikipedia.org/wiki/Precision_and_recall,
- https://en.wikipedia.org/wiki/F1_score.

Problem 3. *(10 points)*
From Lecture 4 – Linear Regression, we fit a line through the input points as follows

$$\begin{aligned} y(\mathbf{x}, \mathbf{w}) &= w_0 + w_1 x_1 + ... + w_D x_D \\ &= \mathbf{w}^T \mathbf{x} \end{aligned}$$

where $\mathbf{w} = (w_0, w_1, ..., w_D)$ and $\mathbf{x} = (1, x_1, ..., x_D)$. It can be shown that the Maximum Likelihood Estimation for $\mathbf{w}$ is the one that minimizes the following loss function:

$$L(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - \mathbf{w}^T x^{(i)} \right)^2 \tag{1}$$

where N is the number of training examples. To prevent overfitting in Linear Regression, we can use a technique called regularization in which we add a penalty term in the loss function to discourage higher values of the coefficients $w_j, j = 1, \ldots, D$. For $\mathbf{l}_2$ regularization (ridge regression), the loss function becomes

$$L_r(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - \mathbf{w}^T x^{(i)} \right)^2 + \lambda \sum_{j=1}^{D} w_j^2 \tag{2}$$

where we use the constant λ to adjust the effect of the regularization term. Calculate the gradient and Hessian of $L_r(\mathbf{w})$ with respect to $\mathbf{w}$, $\nabla_{\mathbf{w}} L(\mathbf{w})$ and $\nabla^2_{\mathbf{w}} L(\mathbf{w})$.

Problem 4 [Mini-project – Linear Regression]. *(30 points)*

In the *scikit-learn*'s diabetes dataset, the 10 features are physiological variables (age, sex, weight, blood pressure) measured on 442 patients. The features have already been mean centered and scaled. The target value is a measure of disease progression after one year. In this mini-project, we will build a Linear Regression model to predict disease progression from the above physiological variables.

```
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

diabetes = datasets.load_diabetes()
print("Features: ", diabetes.feature_names)

# Load the diabetes dataset
diabetes_X, diabetes_y = \
        datasets.load_diabetes(return_X_y=True)
```

(a) Split the dataset to training set and test set using the ratio training set : test set = 8: 2. The utility *model_selection.train_test_split* can be used to split the dataset.
(b) Use scikit-learn's Logistic Regression model *linear_model.LinearRegression()* to train a model using the training set. What are the coefficients (including the intercept) that you get from the model? What is the coefficient of determination (R^2) of the model?
(c) Calculate the Mean Squared Error of the model on the test set.
(d) Use **l_2** regularization (*linear_model.Ridge()*) with parameter alpha in the list $[0.001, 0.01, 0.1, 1, 10, 100, 1000]$. Calculate the Mean Squared Error of the model on the test set for each value of alpha.
(e) Use **l_1** regularization (*linear_model.Lasso()*) with parameter alpha in the list $[0.001, 0.01, 0.1, 1, 10, 100, 1000]$. Calculate the Mean Squared Error of the model on the test set for each value of alpha.

Problem 5 *(10 points)* In Regularized Logistic Regression, the loss function can be written as

$$L(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y^{(i)}log(\sigma(\mathbf{w}^T\mathbf{x}^{(i)})) + (1-y^{(i)})log(1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)}))\right] + \frac{\lambda}{2N}\sum_{j=1}^{D} w_j^2$$

where we use the constant λ to adjust the effect of the regularization term. Calculate the gradient $\nabla_{\mathbf{w}} L(\mathbf{w})$.

Problem 6. [Mini-project – Logistic Regression]. *(30 points)*
In this problem you will logistic regression for a binary classification problem with the cancer dataset available in *scikit-learn* library. The image of a fine needle aspirate (FNA) of a breast mass is used to compute the 30 features in this dataset. For more details, see https://scikit-learn.org/stable/datasets/index.html#breast-cancer-dataset. There are two types of cancer classes: malignant (harmful) and benign (not harmful). The dataset can be loaded as follows

```
# Import scikit-learn dataset library
from sklearn import datasets
# Load dataset
cancer = datasets.load_breast_cancer()
```

The feature names and label names can be printed as follows

```
# print the names of the 13 features
print("Features: ", cancer.feature_names)
# print the label type of cancer('malignant' 'benign')
print("Labels: ", cancer.target_names)
```

The feature data and label data are in *cancer.data* and *cancer.target*.
(a) Split the dataset to training set and test set using the ratio training set : test set = 7 : 3. The utility *model_selection.train_test_split* can be used to split the dataset.
(b) Use scikit-learn's *StandardScaler*, fit and transform each feature into a standard normal distribution. Then use the same parameters of the distributions to transform the test set.
(c) Use scikit-learn's Logistic Regression model *linear_model.LogisticRegression* to train a model on the training set and apply it to the test set.
(d) Let the benign class be the negative class and malignant class be the positive class. Calculate the TP rate, FP rate, FN rate, Precision, Recall, and F1 score on the training set and test set.

References

[1] https://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html
[2] sklearn.datasets.load_iris
https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html