Practical 2: Logistic Regression

In this practical, we'll use Logistic Regression classifier on two different datasets.

Logistic Regression

For logistic regression, you should use the implementation in sklearn. Adding the following line will import the LR model.

```
from sklearn.linear_model import LogisticRegression
```

Read the information provided on the following links to understand some details about how the logistic regression model is implemented in scikit-learn.

- http://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression

Handin 1: In the lectures, we only formulated the Loss function for logistic regression and added a regularization term

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log h(\mathbf{x}_i) + (1-y_i)\log(1-h(\mathbf{x}_i)\right) + \lambda\sum_{j=1}^{D} w_j^2$$

As per this formulation used in the lectures, if you wanted to add a regularization term, and set **λ=0.1** what value of ***C*** would you set in the sklearn implementation?

Iris Dataset

You will now take a look the performance LR on two different datasets. The first is the iris dataset. You can obtain this as follows:

```
from sklearn.datasets import load_iris
iris = load_iris()
X, y = iris['data'], iris['target']
```

There are three classes denoted by 0, 1, 2, which stand for setosa, versicolour and virginica, three varieties of iris. There are four features, all real-valued, measurements of sepal length and width, and petal length and width.

Congressional Voting Records Dataset

The second dataset you will use is voting records in the US House of Representatives in 1984. The goal is to predict whether the representative is a Republican or Democrat. The original dataset is available here:

http://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

For the purpose of this practical, we have put the data in numpy array format, as well as deleted those records that had missing entries. This dataset is available on the slack and can be loaded as follows:

```
import pickle as cp
import numpy as np
X, y = cp.load(open('voting.pickle', 'rb'))
```

A vote of yes is encoded as 1, no as 0. A Republican is 0 and a Democrat 1.

Experiments

For both datasets you'll compare the classification error of tLR trained on increasingly large training datasets. Because the datasets are so small, you should do this multiple times and average the classification error. One run should look as follows:

- Shuffle the data, put 20% aside for testing.

  ```
  N, D = X.shape
  Ntrain = int(0.8 * N)

  shuffler = np.random.permutation(N)

  Xtrain = X[shuffler[:Ntrain]]
  ytrain = y[shuffler[:Ntrain]]

  Xtest = X[shuffler[Ntrain:]]
  ytest = y[shuffler[Ntrain:]]
  ```

- Train 10 classifiers, where the **k**-th classifier is trained using 10k% of the training data. For each classifier store the classification error on the test set.

You may want to repeat this with at least 200 random permutations (possibly as large as 1000) to average out the test error across the runs. In the end, you'll get average test errors as a function of the size of the training data. Plot these curves on the two datasets.

Handin 2: Include the plots of the two curves in your report.