SGN-41007 Pattern Recognition and Machine Learning

Exercise Set 7: February 22–February 24, 2017

Exercises consist of both pen&paper and computer assignments. Pen&paper questions are solved at home before exercises, while computer assignments are solved during exercise hours. The computer assignments are marked by **python** and Pen&paper questions by **pen&paper**

1. **pen&paper** *Error rate confidence limits.*

   We train a classifier with a set of training examples, and test the accuracy of the resulting model with a set of $N = 100$ test samples. The classifier misclassifies $K = 5$ of those.

   a) Find the 90% confidence interval of the result. Hint: The classification accuracy can be modeled using binomial distribution, whose confidence intervals are discussed here:

   `https://en.wikipedia.org/wiki/Binomial_distribution#Confidence_intervals`

   b) Another classifier misclassifies only 3 test samples. Is it better than the first one with statistical significance at 90% confidence level?

2. **pen&paper** In Exercise set 5 (question 2a), we derived the formula for the gradient of log-loss.

   a) Compute the gradient for L_2 penalized log-loss.

   b) Study also the gradient for L_1 penalized log-loss. Propose an approximation, whose gradient would be defined for all $\mathbf{w}$.

3. **python** Implement the L_2 penalized log-loss minimizer in Python. You can use the template of Question 3 at Exercise set 5.

4. **python** Apply the recursive feature elimination approach (`sklearn.feature_selection.RFECV`) with logistic regression classifier for the arcene dataset. The data can be downloaded in `*.mat` format from:

   `http://www.cs.tut.fi/courses/SGN-41007/exercises/arcene.zip`

   Use `scipy.io.loadmat` to open the file. Note that your have to ravel `y_train` and `y_test` so that `sklearn` will accept them.

   a) Instantiate an RFECV selector (call it `rfe` from now on). To speed up computation, set `step = 50` in the constructor. Also set `verbose = 1` to see the progress.

   b) Fit the RFECV to `X_train` and `y_train`.

c) Count the number of selected features from `rfe.support_`.

d) Plot the errors for different number of features:
`plt.plot(range(0,10001,50), rfe.grid_scores_)`

e) Compute the accuracy on `X_test` and `y_test`. You can use `rfe` as any other classifier.

5. **python** Apply L_1 penalized Logistic Regression for feature selection with the arcene dataset. Find a good value for parameter C by 10-fold cross-validating the accuracy. Study the sparseness of the solution: how many features were selected?

a) Instantiate a LogisticRegression classifier. Set `penalty = 'l1'` in the constructor.

b) Cross validate the accuracy of a range of `C` values (see earlier exercises).

c) Fit the LogisticRegression to `X_train` and `y_train`.

d) Count the number of selected features from `clf.coef_`, where clf is your logistic regression classifier.

e) Compute the accuracy on `X_test` and `y_test`.