Machine Learning
Exercise 6

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

June 13, 2014

This exercise is meant for TWO WEEKS. June 19 is again a holiday. We'll discuss these on June 26.

1 SVMs

a) Draw a small dataset $\{(x_i, y_i)\}$, $x_i \in \mathbb{R}^2$ with two different classes $y_i \in \{0, 1\}$ such that a 1-nearest neighbor (1-NN) classifier has a lower leave-one-out cross validation error than a linear SVM classifier.

b) Draw a small dataset $\{(x_i, y_i)\}$, $x_i \in \mathbb{R}^2$ with two different classes $y_i \in \{0, 1\}$ such that 1-NN classifier has a higher leave-one-out cross validation error than a linear SVM classifier.

c) Proof that the constrained optimization problem (where $y_i \in \{-1, +1\}, C > 0$)

$$\min_{\beta,\xi} \|\beta\|^2 + C \sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad y_i f(x_i) \geq 1 - \xi_i \;, \quad \xi_i \geq 0$$

can be rewritten as the unconstrained optimization problem

$$\min_{\beta} \|\beta\|^2 + C \sum_{i=1}^{n} \max\{0,\ 1 - y_i f(x_i)\} \;.$$

(Note that the $\max$ term is the hinge loss. Compare slide 04:80.)

2 Breadth of ML with scikit-learn or Weka

We're going to explore some of the "breadth of ML" with existing ML toolkits.

scikit-learn `http://scikit-learn.org/` is a ML toolkit in python. It is very powerful, implements a large number of algorithms, has a nice interface, and a good documentation.

Weka `http://www.cs.waikato.ac.nz/ml/weka/` is another widely used ML toolkit. It is written in Java and offers a GUI where you can design you ML pipeline.

Here are the instuctions for sklearn, but you can also do the same with Weka[1].

a) Install sk-learn. There is an install guide `http://scikit-learn.org/stable/install.html` and Ubuntu packages exist.

b) Read the tutorial: `http://scikit-learn.org/stable/tutorial/basic/tutorial.html`

c) Download the "MNIST dataset" via sklearn. `http://scikit-learn.org/stable/datasets/index.html#downloading-datasets-from-the-mldata-org-repository`

d) Try to find the best classifier for the data and compare the performance of different classifiers. Try to cover some of the "breadth of ML". Definitely include the following classifiers: LogisticRegression, kNN, SVM, Ensemble methods (DecissionTrees, GradientBoosting, etc.).

d) Try out the same after transforming the data (PCA, KernelPCA, Non-Negative matrix factorization, Locally Linear Embedding, etc.).

[1] However, in the previous year most students did not enjoy Weka.

- http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition
- http://scikit-learn.org/stable/modules/classes.html#module-sklearn.manifold

e) Many classifiers have parameters such as the regularization parameters or kernel parameters. Use GridSearch to find good parameters. http://scikit-learn.org/stable/modules/classes.html#module-sklearn.grid_search

Here is a little example of loading data, splitting the data into a test and train dataset, training a SVM, and predicting with the SVM.

```
sklearn import svm
from sklearn import datasets
from sklearn.cross_validation import train_test_split

iris = datasets.load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, b_test = train_test_split(X, y, test_size=0.33)
clf = svm.SVC()
clf.fit(X_train, y_train)
clf.predict(X_test)
```