Machine learning lab course (Praktikum)

Sommersemester 2014

Fachgebiet Maschinelles Lernen
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Problem set 4: Support Vector Machines

Part 1: Implementation

Assignment 1 (20 points)

Implement the SMO algorithm for SVMs. Refer to the handbook for pseudo code. The signature follows that of kernel ridge regression:

`svm_smo(kernel, kernelparameter, regularization)`

with functions `fit(X, y)` and `predict(X)`. The labels in y will be +1/-1. Your `fit` method should only save the support vectors, not all data points. The class has three subfunctions for the SMO algorithm:

`_compute_box_constraints`, `_update_parameters` and `_compute_updated_b`.

The test script provided will test these functions to make potential debugging easier.
Hint: If you do predictions, you might want to use the `sign` function to get +1/-1 labels.

Assignment 2 (15 points)

Implement a plotting method

`plot_svm_2d(X, y, model)`

that takes as input a $(2 \times n)$ data matrix `X`, +1/-1 labels `y` and a fitted svm model. It should plot the points in `X` as circles in different colors, mark support vectors with a cross *and plot the separating hyperplane.*
Hint: To plot the separating hyperplane evaluate your classifier on each point of a grid that stretches all data points and then plot the contour line.

Assignment 3 (15 points)

Write the SVM dual optimization problem as a quadratic programming (QP) problem in the form of

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^\intercal P x + q^\intercal x \\ \text{s.t.} \quad & Gx \preceq h \\ & Ax = b \end{aligned}$$

Implement this in the class `svm_qp` with the help of `cvxopt.solvers.qp` (see stubs). The class should have the same signature and functions as `svm_smo`. You might have to install the `cvxopt` package via `pip install cvxopt`.

Part 2: Application

Assignment 4 (10+10+5 points)

Use your SVM SMO implementation to train classifiers on the `easy_2d` dataset from the ISIS site.

1. Find the optimal parameters C and σ for a Gaussian kernel and plot the results. Use your cross validation method. If you do not have a running cross validation method, contact us.

2. Train one model for a σ and C that obviously overfit and for a σ and C that obviously underfit the data. Plot the results.

3. For optimal C and σ, plot a receiver operator characteristics (ROC) curve by varying the bias parameter b of your SVM model.

Assignment 5 (5+5 points)

1. Compare the running time of the `svm_qp` implementation and a properly implemented SMO routine from `scikit-learn` (See the already implemented `svm_sklearn` class. Note that you might have to install scikit-learn via `pip install sklearn`). Which algorithm is faster?

2. Besides from an implementation in a low level language like C, how would you optimize your SMO implementation? I.e. where in the algorithm described in the handbook/guide do you see room for improvement?

Assignment 6 (15 points)

In this assignment you will work on the famous UCI Iris dataset[1]. This is a 4-dimensional dataset with 150 instances in 3 classes. Use the `.npy` file from the ISIS site. Which classes are linearly separable from the two other classes and which classes are not? Are they separable with a non-linear classifier? Describe what you tested. Provide the found hyperparameters and classification accuracies.

[1] http://archive.ics.uci.edu/ml/datasets/Iris