Exercises
Data Mining: Learning from Large Data Sets
FS 2016

LAS Group, Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `http://las.ethz.ch/teaching/dm-f16/`
Email questions to:
Carlos Cotrini, ccarlos@inf.ethz.ch

Series 3, Oct 27th, 2016 (Online Convex Programming)

It is not mandatory to submit solutions and sample solutions will be published after one week. If you choose to submit your solution, please send an e-mail from your `ethz.ch` address with subject `Exercise3` containing a PDF (LaTeX or scan) to `ccarlos@inf.ethz.ch` until Wednesday, Nov 2nd 2016.

Problem 1 (Online SVM):

For this exercise, you will implement an online support vector machine algorithm to classify handwritten digits. The data set you will use is included in the `hw3-sup.zip` file available on the course webpage. It consists of 11500 training examples, and 1900 test examples, taken from the MNIST handwritten digits dataset (3s and 5s). Each row in the files `Xtrain.csv` and `Xtest.csv` contains features that represent pixel intensities. The other two files, `Ytrain.csv` and `Ytest.csv`, contain labels with -1 corresponding to digit 5, and 1 to digit 3.

(a) Implement the online SVM algorithm presented in the lecture, and shown again in Algorithm 1 below.

(b) Find a suitable value for the regularization parameter λ, for example, using cross-validation. You should only use the training set for this task, and not modify the value based on test set results.

(c) By training the SVM on random subsets of the full training set, compute the classification error of the test set as a function of the number of training examples, and create a plot that illustrates this dependence.

Algorithm 1 Online support vector machine

function ocp_svm(X,Y,λ)
$\mathbf{w} \leftarrow 0$
for $t = 1, 2, \ldots$ **do**
    $\eta_t = 1/\sqrt{t}$
    if $y_t \mathbf{w}^T \mathbf{x}_t < 1$ **then**
        $\mathbf{w} \leftarrow \mathbf{w} + \eta_t y_t \mathbf{x}_t$
        $\mathbf{w} \leftarrow \mathbf{w} \ \min\left(1, 1/(\sqrt{\lambda}\|\mathbf{w}\|)\right)$
    end if
end for
return w

Problem 2 (Online Logistic Regression):

Modify the code of the previous problem, and repeat the steps described above, to implement the online logistic regression algorithm discussed in the tutorial session:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \mathbf{w}^T x_i\right)\right) \qquad \text{s.t. } \|\mathbf{w}\|_1 \leq \frac{1}{\sqrt{\lambda}},$$

where $\| \cdot \|_1$ is the ℓ_1-norm. Compare the accuracy and runtime of the two algorithms.

Note: For the projection step onto the ℓ_1-ball, you can use the algorithm described by Duchi et al. [1] in Section 4 of their paper.

Solution 1 & 2:

The full Python code that implements the two online algorithms can be find in `hw3-sol-sup.zip` that is available on the course website. Figure 1 shows the improvement in test accuracy as the number of training instances provided to the two algorithms increases. The lines show mean accuracy across 100 repetitions for different permutations of the training instances; the error bars depict ± 2 standard errors.

A few additional comments:

- While logistic regression seems to be doing slightly better here, remember that its main advantage over SVM is the fact that it provides explicit uncertainty estimates about the classification of each test instance, which can be useful in some applications (e.g., for active learning).
- The main advantage of L_1 regularization is the sparsity of the resulting weight vectors, which can make it easier to interpret the trained model, and might also make prediction more efficient for large feature spaces. This comes at the expense of a larger training time, due to the fact that projecting onto the L_1 ball is less efficient than projecting onto the L_2 ball.

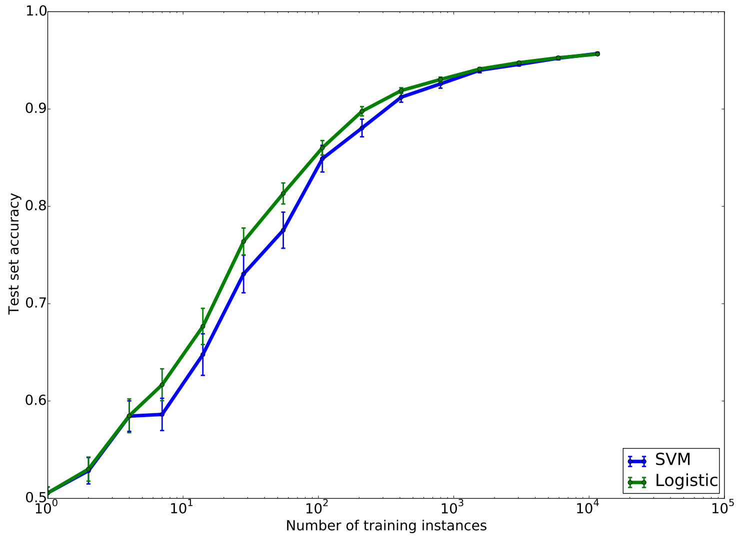

Figure 1: Accuracy vs number of training instances for the two OCP algorithms.

References

[1] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *International Conference on Machine Learning (ICML)*, 2008.