Problem Set 04
Data Mining and Machine Learning – Spring 2016
Due date: 2016-03-04 13:00

All assignments must be uploaded to the assignments tab in ClassesV2 (notice that this is **not** the dropbox) by the date and time specified. Make sure that you follow the instructions exactly as described. This is a large class with a limited number of TA's and we will be grading parts of the assignment using an automated grading engine; you will lose points for things such as not naming files correctly. You may discuss problem sets with others, but must write up your own solutions. This means that you should have no need to look at other's final written solutions.

You need to turn in all of your solutions as a zip compressed file, named `netid_pset04.zip`, with your actual netid filled in in all lower case letters. This archive should contain just one file:

- `pset04.py` or `pset04.R`

The contents of this file is described below.

SVM Implementation

The task for this assignment is to write two implementations to calculate a linear support vector machine given a set of training data. In both cases you do not need to support any kernels, nor do you need to include a bias term (i.e., we are using a no-intercept model).

The first implementation uses the dual form of the support vector machine equations. Specifically, you need to solve the optimization problem:

$$\max_{\alpha} \quad \left\{ \sum_i \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_i \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'} \right\}$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq C.$$

Which can be re-written as:

$$\max_{\alpha} \quad \left\{ 1_n^t \alpha_i - \alpha^t K \alpha \right\}$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq C.$$

For a suitable matrix K. The first implementation should feed this equation into a general purpose library intended for solving constrained optimization problems. (Hint: this should be fairly easy once you find a suitable library; suggestions are given in the sample code).

The second task requires you to implement the basic Pegasos algorithm from the following paper:

> Shalev-Shwartz, Shai, Yoram Singer, and Nathan Srebro. "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM." In: *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

You can download a copy of the paper here:

`http://www.ee.oulu.fi/research/imag/courses/Vedaldi/ShalevSiSr07.pdf`

Specifically, you need to implement the algorithm in Figure 1 of the paper.

The function names and formats of these two required functions are described in the starter code and test scripts on the course website:

`http://www.stat.yale.edu/~tba3/stat665/psets/pset04/pset04_start.py`
`http://www.stat.yale.edu/~tba3/stat665/psets/pset04/pset04_start.R`
`http://www.stat.yale.edu/~tba3/stat665/psets/pset04/testScript04.R`
`http://www.stat.yale.edu/~tba3/stat665/psets/pset04/testScript04.py`

The test scripts serve to make sure that the **format** of your submission is correct; it is up to you to build tests to see if the actual results make sense. For example, compare your results to standard implementations of support vector machines. Of particular note is the fact that you must return the vector β, not the weights α.