STAT 6910-003 – SLDM II – Homework #4

Due: 5:00 PM 11/16/18

1. **Convex Losses (10 pts).** We say that a loss is convex if for each fixed y, $L(y, t)$ is a convex function of t.

   (a) (5 pts) Show that the logistic loss is convex.

   (b) (5 pts) Show that if L is a convex loss, then

   $$\hat{R}(\boldsymbol{w}, b) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \boldsymbol{w}^T\boldsymbol{x}_i + b)$$

   is a convex function of $\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}$.

2. **Conceptual questions (8 pts).**

   (a) (4 pts) Linear classifiers. Discuss the differences between LDA, Logistic Regression, separating hyperplanes, and the Optimal Soft-Margin Hyperplane. In what situations would you use each of the classifiers? *Hint:* You may find the discussion in sections 4.5.2 and 12.2 in the ESL book or sections 9.1, 9.2, and 9.5 in the ISL book helpful.

   (b) (4 pts) Describe how you would apply the SVM to the multiclass case.

3. **Kernels (22 pts).**

   (a) (4 pts) To what feature map Φ does the kernel

   $$k(\boldsymbol{u}, \boldsymbol{v}) = (\langle u, v\rangle + 1)^3$$

   correspond? Assume the inputs have an arbitrary dimension d and the inner product is the dot product.

   (b) (18 pts) Let k_1, k_2 be symmetric, positive-definite kernels over $\mathbb{R}^D \times \mathbb{R}^D$, let $a \in \mathbb{R}^+$ be a positive real number, let $f : \mathbb{R}^D \to \mathbb{R}$ be a real-valued function, and let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial with positive coefficients. For each of the functions k below, state whether it is necessarily a positive-definite kernel. If you think it is, prove it. If you think it is not, give a counterexample.

      i. $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$
      ii. $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) - k_2(\mathbf{x}, \mathbf{z})$
      iii. $k(\mathbf{x}, \mathbf{z}) = ak_1(\mathbf{x}, \mathbf{z})$
      iv. $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z})k_2(\mathbf{x}, \mathbf{z})$
      v. $k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$
      vi. $k(\mathbf{x}, \mathbf{z}) = p\left(k_1(\mathbf{x}, \mathbf{z})\right)$

4. **Alternative OSM hyperplane (10 pts).** An alternative way to extend the max-margin hyperplane to nonseparable data is to solve the following quadratic program (another name for an optimization problem with a quadratic objective function):

   $$\begin{array}{cc} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i^2 \\ s.t. & y_i(\boldsymbol{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \ \forall i \\ & \xi_i \geq 0 \ \forall i \end{array}$$

   The only difference with respect to the OSM hyperplane is that we are now squaring the slack variables. This assigns a stronger penalty to data points that violate the margin.

(a) (4 pts) Which loss is associated with the above quadratic program? In order words, show that learning a hyperplane by the above optimization problem is equivalent to ERM with a certain loss.

(b) (4 pts) Argue that the second set of constraints can be dropped without changing the solution.

(c) (2 pts) Identify an advantage and a disadvantage of this loss compared to the hinge loss.

5. **Subgradient methods for the optimal soft margin hyperplane (25 pts).**
In this problem you will implement the subgradient and stochastic subgradient methods for minimizing the convex but nondifferentiable function

$$J(\boldsymbol{w}, b) = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, \boldsymbol{w}^T \mathbf{x}_i + b\right) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$$

where $L(y,t) = \max 0, 1 - yt$ is the hinge loss. As we saw in class, this corresponds to the optimal soft margin hyperplane.

(a) (5 pts) Determine $J_i(\mathbf{w}, b)$ such that

$$J(\mathbf{w}, b) = \sum_{i=1}^{n} J_i(\mathbf{w}, b).$$

Determine a subgradient $\mathbf{u}_i$ of each J_i with respect to the variable $\boldsymbol{\theta} = [b\, \mathbf{w}^T]^T$. A subgradient of J is then $\sum_i \mathbf{u}_i$.
Note: Recall that if $f(\mathbf{z}) = g(h(\mathbf{z}))$ where $g : \mathbb{R} \to \mathbb{R}$ and $h : \mathbb{R}^d \to \mathbb{R}$, and both g and h are differentiable, then the chain rule gives

$$\nabla f(\mathbf{z}) = \nabla h(\mathbf{z}) \cdot g'(h(\mathbf{z})).$$

If g is convex and h is differentiable, then the same formula gives a subgradient of f at $\mathbf{z}$ where $g'(h(\mathbf{z}))$ is replaced by a subgradient of g at $h(\mathbf{z})$.

(b) (6 pts) Download the file `nuclear.mat` from Canvas. The variables x and y contain training data for a binary classification problem. The variables correspond to the total energy and tail energy of waveforms produced by a nuclear particle detector. The classes correspond to neutrons and gamma rays. Neutrons have a slightly larger tail energy for the same total energy relative to gamma rays, which allows the two particle types to be distinguished. This is a somewhat large data set ($n = 20,000$), and subgradient methods are appropriate given their scalability.
Implement the subgradient method for minimizing J and apply it to the nuclear data. Submit two figures: One showing the data and the learned line, the other showing J as a function of iteration number. Also report the estimated hyperplane parameters and the minimum achieved value of the objective function.
Comments:

- Use $\lambda = 0.001$. Since this is a linear problem in a low dimension, we don't need much regularization.
- Use a step-size of $\alpha_j = 100/j$, where j is the iteration number.
- To compute the subgradient of J, write a subroutine to find the subgradient of J_i and then sum those results.
- Since the objective will not be monotone decreasing, determining a good stopping rule can be tricky. For this problem, just look at the graph of the objective function and "eyeball it" to decide when the algorithm has converged.
- Debugging goes faster if you start out with just a subsample of the data.

(c) (6 pts) Now implement the stochastic subgradient method with a minibatch size of $m = 1$. Be sure to cycle through all data points before starting a new loop through the data. Report/hand in the same items as for part (b).
More comments:

- Use the same λ, stopping strategy, and α_j as in part (b). Here j indexes the number of times you have cycled (randomly) through the data; i.e., it is the number of epochs you have trained for.
- Your plot of J versus iteration number will have roughly n times as many points as in part (b) since you have n updates for every one update of the full subgradient method.

(d) (3 pts) Comment on the empirical rate of convergence of the stochastic subgradient method relative to teh subgradient method. Explain your finding.

(e) (5 pts) Submit your code.

6. **Classification (15 pts).**

(a) (2 pts) Download a dataset for classification from the UCI Machine Learning Repository (you can Google it). Report the number of classes and the number of features in your chosen dataset.

(b) (3 pts) Apply logistic regression to this dataset. Calculate the test error based on k-fold cross-validation (you may select k) and report the training and test error. You may use any built-in methods in your language of choice.

(c) (8 pts) Apply the SVM to this dataset using 2 different kernels: linear and Gaussian. Describe the tuning parameters in each case and set these parameters by cross-validation. Report your final test error, training error, and tuning parameters after cross-validation.

(d) (2 pts) Which of the three methods you applied (logistic regression, SVM with linear kernel, SVM with Gaussian kernel) worked best? Do your results make sense?
Comments:

- The linear kernel for SVM is equivalent to applying the optimal soft-margin hyperplane.
- You will probably want to choose a dataset that does not have any categorical features.
- Depending on which code you use, you may want to choose data with a binary classification problem.
- The SVM can take a while to train so you may want to consider sample size when choosing a dataset.

7. How long did this assignment take you? (5 pts)

8. Type up homework solutions (5 pts)