6.867: Homework 2

In this assignment we will look at two popular methods for classification: logistic regression and SVMs. We will be following the same process that we did for Homework 1 (and repeated in the last section of this handout).

- This assignment may be done by pairs of students.
- You submit your paper via the Easy Chair site by 11:59 PM on Thursday October 25.
- Each student who was an author or co-author on a submission will be assigned several papers to review.
- Reviews must be entered on the Easy Chair site by 11:59PM on Thursday, November 1.
- All reviews will be made visible shortly thereafter, and students will have a 2–3 day period in which to enter rebuttals of their reviews into EasyChair if they wish.

The following questions are the points that your paper should cover in order to receive full credit. Your presentation should roughly follow the order of these questions so that your reviewers can see what you're doing.

1 Logistic Regression

We have given you three data sets (in the `data` folder); the name of each data set indicates whether it is to be used as a "training" set or as a "validation" set. Note that for consistency with SVMs, we will label all the data with $y^{(i)} \in \{-1, 1\}$. See the discussion in Bishop, section 7.1.2 about how to reformulate the logistic regression model for this case; equations (7.47) and (7.48) define the relevant log likelihood function.

1. Implement logistic regression using one of the minimizers you experimented with in Homework 1 (your own gradient descent or one of the other optimization methods you tried). Include a quadratic regularizer on the weights. Write a prediction function that can be used to "predict" an input point, that is, compute the posterior probability $P(y \mid x, w)$. Use this to report a count of the number of mistakes on the training data. Look in the file `logreg_test` for a simple skeleton that shows how to read files and plot results.

2. Test your implementation on the data sets provided. Report the behavior of your algorithm when $\lambda = 0$ (no regularization) on the training and validation data sets. Report the behavior of your algorithm when λ increases on the training and validation data sets. Report the general trends in what you find; comment specially on performance on non-separable data.

3. Extend your logistic regression implementation to handle second order (polynomial) basis functions, in particular, you should have basis functions for 1, for x_i for $i = 1, \ldots, D$ and for $x_i x_j$ for $i = 1, \ldots, D$, $j = i, \ldots, D$, where D is the dimension of the input data. Report the performance of the generalized regression on the training and data set. Illustrate the effect of regularization and explain what you see.

2 Support vector machine implementation

1. Implement a linear SVM with slack variables. You should implement both the primal and the dual form; compare the results to verify that you have implemented them correctly. See the separate file `optimizers.txt` on how to install and call the optimizers. Write a prediction function that can be used to "predict" an input point, that is, compute the functional margin of the point. Look in the file `svm_test` for a simple skeleton that shows how to read files and plot results.

2. Run your SVM implementation on the training and validation data sets we gave you. Report your results and explain what you find.

3 Support vector machine interpretation

Include your answers to these questions in your paper submission.

1. Consider a very simple two-point dataset where `X = [1 0;-1 0]` (each row is a data point) and `Y = [1; -1]`.
   (a) Why doesn't the solution change if we select any $C > 1$? Explain this in terms of the optimization problem.
   (b) When $C = 1/4$, we obtain a geometric margin of size 2. Construct by hand a possible solution to $\mathbf{w}$, b and the slack variables in this case. Is the solution unique?

2. Run your implementation on the datasets provided with the code. Try $C = 0.01, 0.1, 1, 10, 100$ and show your results, both graphically and by reporting the number of mistakes on the training and validation data sets.
   (a) What happens to the geometric margin $1/||\mathbf{w}||$ as C increases? Will this always happen as we increase C?

(b) What happens to the number of support vectors as C increases?

(c) The value of C will typically change the resulting classifier and therefore also affects the accuracy on test examples. Why would maximizing the geometric margin $1/\|\mathbf{w}\|$ on the training set not be an appropriate criterion for selecting C? Is there an alternative criterion that we could use for this purpose?

3. Suppose we fix C and obtain $\mathbf{w}$, b, and ξ_i as the solution to the quadratic programming problem. The slack is relevant (non-zero) only for support vectors. How does the value of slack ξ_i relate to the distance of a support vector $x^{(i)}$ from the decision boundary?

4. The quadratic programming problem involves both $\mathbf{w}, b$ and the slack variables ξ_i. We can rewrite the optimization problem in terms of $\mathbf{w}, b$ alone. This is done by explicitly solving for the optimal values of the slack variables $\xi_i = \xi_i(\mathbf{w}, b)$ as functions of $\mathbf{w}, b$. The values of these slack variables, as functions of $\mathbf{w}, b$, are "loss-functions" as shown below. What functions $\xi_i(\mathbf{w}, b)$ determine the optimal ξ_i? Are all the margin constraints satisfied with these expressions for the slack variables?

   The resulting minimization problem over $\mathbf{w}, b$ can be formally written as

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i(\mathbf{w}, b)$$

   where the first (regularization) term biases our solution towards zero in the absence of any data and the remaining terms give rise to the loss functions, one loss function per training point, encouraging correct classification. Do we need any additional constraints? Many learning criteria can be understood and compared in the above regularization + loss form.

4 Kernel SVM

1. Extend your dual SVM implementation to use kernels. Hint: this is a relatively small change. Implement the prediction function. Test a second order polynomial kernel and a Gaussian kernel on the training and validation data. You should **not** have to modify your basic dual solution when you change kernels. Show (and explain) your results both graphically and by reporting mistakes for several values of C and the Gaussian kernel variance. Compare to your logistic regression results.

Grading process ("same as it ever was")

You will find a zip file with some useful code and data in the Resources section of the Piazza course page. You can do these assignments in any computational system that you are used to. We

recommend Matlab or the Pylab/Numpy/Scipy/Matplotlib cluster of packages and we'll try to provide help for those two systems. If you use anything else, you're on your own...

You will be turning in a single, readable "paper" (a single PDF file) with your solutions. We will be emulating the process of submitting papers for publication to a conference. We will be using an actual conference review system (Easy Chair) to have these papers peer reviewed (by the other students). This means that your answers have to be readable and understandable to your peers and, where possible, interesting. Note that when explanations are called for, you will need to convince the reviewers that you understand what you're talking about. The course staff will serve as the Program Committee for the conference and make all final decisions. The details of this process are posted on Piazza.

Grading rubric

Your paper must be anonymous (no identifying information should appear in the PDF file). If it is not, it will automatically receive a 20% deduction, and will be graded by a grumpy staff member.

The paper must be no more than 6 pages long in a font no smaller than 10 point. It should include whatever tables, graphs, plots, etc., are necessary to demonstrate your work and conclusions. *It should not include code.*

Each of the four parts of the assignment will be graded on a scale from 0 to 5 (where 0 is failing and 5 is an A) on two aspects:

- **Content:** Did the solution answer the questions posed? Were the answers correct? Were the experiments well-designed or examples well chosen?
- **Clarity:** Were the results written up clearly? Were the plots labeled appropriate and described well? Did the plots support the points in the paper? Did the discussion in the paper illuminate the plots?

As a reviewer, you will be asked to provide a score for each section, and at at least two paragraphs of feedback, per review, explaining things that were done well and things that could have been improved upon.

Your overall score for this assignment will be:

- **80%**: The average of all 8 scores on your assignment given by all three reviewers.
- **20%**: A score for the quality of your reviews. This will be full credit, by default. But we will skim reviews and examine some carefully and may reduce this grade for review commentary that is sloppy or wrong.

The course staff will spot-check submissions and reviews, paying careful attention to cases where there were rebuttals. The staff will act as the program committee and determine a final score. Our overall goals in this process are:

- To motivate you to work seriously on the problems and learn something about the machine learning material in the process
- To engage you in thinking critically and learning from other students' solutions to the problems

We will arrange to give full credit to anyone who submits a serious and careful solution to the problems and who gives evidence of having read carefully the solutions they were assigned and who writes thoughtful reviews of them.