Project 4

due: 12/15/16 (midnight)

December 1, 2016

Problem 1 (20pt): Kernel Logistic Regression

Develop an algorithm for kernel logistic regression. Let $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$ be the set of training examples, where $\mathbf{x}_i \in \mathcal{R}^d$ and $y_i \in \{-1, +1\}$. In the linear logistic regression model, we write $Pr(y|\mathbf{x})$ as

$$Pr(y|x) = \frac{1}{1 + \exp(-y[\mathbf{w}^T\mathbf{x} + b])}$$

and obtain the optimal linear classifier model $\mathbf{w}$ (recall that we usually implicitly solve for b by adding an additional feature to $\mathbf{x}$ that is all 1s) by maximizing the regularized log-likelihood, i.e.,

$$\mathbf{w}^* = \arg\min \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \ln(1 + \exp(-y_i[\mathbf{w}^T\mathbf{x}_i + b])).$$

The first step towards the kernel logistic regression model is to derive the dual formulation of logistic regression. The dual should end up with a formulation that is dependent on the dot product $\mathbf{x}_i^T\mathbf{x}_j$, which you replace with the kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$. Note that you will end up with both an algorithm for finding the optimal $\mathbf{w}$ and an expression for $Pr(y|\mathbf{x})$. In this homework, you need to submit as a PDF (use a computer editor with equation editor, such as Word or LaTeX):

- Your derivation of the dual formulation for linear logistic regression;
- The kernel version of the dual formulation for logistic regression;
- The kernel version of $Pr(y|\mathbf{x})$ used the obtained dual variables.

Problem 2 (30pt): Apply Kernel Logistic Regression

Apply your regularized kernel logistic regression algorithm to the `heartstatlog` data. Randomly select 80% of the training data to form the training set and use the remaining 20% for validation (do this for a few different random draws and average the results). Set the candidate values of $\lambda = 0.01, 0.05, 0.25, 1, 5, 25, 100$. For each λ report and plot the training error (the error you get when apply your classifier to the training data), validation error (the error you get when you apply your classifier to the cross-validation test data), and test error (the error you get on the testing data). Try this for different kernels. As a sanity check, if you use the linear kernel, i.e, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$, you should get the results you get for linear logistic regression.