

# CPS 271 - Machine Learning

## Homework Due 11/12/09

### 1 Dual (Kernelized) regression

Equation 6.9 in the textbook gives an expression for the regularized, kernelized, prediction for a new input  $x$ . Prove that if  $K$  is invertible and  $\lambda = 0$ , then the kernelized regression will have perfect performance on the training data. (This is a very easy problem once you understand what  $k$  and  $K$  represent, so this is an exercise in getting you familiar with these concepts.)

### 2 Kernels I

Using the other rules for constructing valid kernels on the top of p. 296 of the textbook, prove that 6.15 is true.

### 3 Kernels II

*Without* using any of the other rules on the top of p. 296, prove the 6.13 is true.

### 4 Implementing Gaussian Process Regression

Implement Gaussian process regression and reproduce figure 6.8 for 3 different values of  $\beta$ . You don't need to reproduce the green line or the shading, but you should use some method of showing how the variance changes with  $\beta$ .

**Note that for the last two problems, we are *not* asking you to graph results with a separate test set.**

### 5 Implementing Support Vector Machines

Implement the primal form of the SVM with misclassification penalties using quadprog. (This is much simpler than the dual form, though it doesn't lend itself to kernelized implementations.) Test it on the dataset from Figure A.7 (available from the book web page) and produce a plot showing the decision boundary as well as the correctly and incorrectly classified points for a few different values of  $C$ . If your implementation runs too slowly on your machine, you may use a subset of the data. (Note that for solving a real problem, you would be much better off with an off-the-shelf SVM implementation, first because such implementations are compatible with the kernel trick, and second because they take advantage of specific structure in the SVM optimization problem that is not present in generic QPs to achieve a much faster implementation.)

## 6 Support Vector Machines with Richer Features

For this problem, repeat the previous problem but with the kernel  $k(x, y) = (x^T y + 1)^2$ . It's OK to "cheat" on this by running your code from the previous problem with the features that correspond to the space implicit with this kernel. If you're feeling more ambitious, you can try implementing the dual form of the SVM, but that is *not* required since it's a bit messy. Rerun your experiments from the previous problem with the same values of  $C$ , and produce a new set of figures. You should expect to see improved classification performance.