MATH/AMATH 516 Course Project May 22, Spring 2009

For the course project, you have two options:

Option 1. Write a report on a topic in numerical optimization. It should be related to the course material and have "sufficient" mathematical content at a graduate level. It can involve modeling and analysis of a real-world problem or a study of a class of nonlinear optimization problems and/or algorithms. A length of 6-10 pages seems reasonable (excluding figures and references). The report will be evaluated based on clarity, content, and understanding. You might check with me on whether a topic is suitable before proceeding further with it.

Option 2. Write a computer program in a standard programming language (C, C++, Fortran, Basic, Matlab, etc.) to solve the support vector machine (SVM) problem for two-class data classification. In this problem, we are given n data points $z_i \in \Re^p$ and their classification $a_i \in \{-1, 1\}$, $i = 1, \ldots, n$. The SVM classifier is obtained by solving the following optimization problem:

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2} x^T Q x - e^T x \\ \text{s.t.} \quad & a^T x = 0, \quad 0 \le x \le Ce, \end{aligned}$$

where $a = (a_1, \ldots, a_n)$, $C > 0$, $e = (1, \ldots, 1)$, and $Q \in \Re^{n \times n}$ is a symmetric matrix with entries of the form

$$Q_{ij} = a_i a_j K(z_i, z_j).$$

Popular choices of $K : \Re^p \times \Re^p \to \Re$ are the linear kernel $K(z_i, z_j) = z_i^T z_j$ (for which $Q = Z^T Z$, with $Z = [a_1 z_1 \ \cdots \ a_n z_n]$) and the radial basis function (rbf) kernel $K(z_i, z_j) = \exp(-\gamma \|z_i - z_j\|^2)$ where γ is a constant. Typically p ("number of features") is not large ($4 \le p \le 300$), but n is large ($n \ge 5000$), and Q is fully dense.

The data for your program would consist n, p and a_i, the nonzeros of z_i, and their indices, $i = 1, \ldots, n$. Test data will be posted on the class webpage. Your program should allow the user to choose C and the kernel. (Take $\gamma = 1/p$, as is typical.) The program should be documented and it will be evaluated based on readability and efficiency. (It should run correctly of course.) Configure your code so that it can run on Unix/Linux (since that's the operating system I will use to run it). You can implement any method that seems suitable (conditional gradient, gradient projection, interior-point method,..) Note that $x = 0$ is a feasible point. Here, f and ∇f are both expensive when n is large. You might benchmark your program against a public-domain program such as LIBSVM and SVM^{light} (found by google). (Free dinner if your code can beat LIBSVM or SVM^{light} in cpu time!)

Caution: If you choose Option 2, it's a good idea to start on it early, especially if you are not an experienced programmer.

The project should be turned in by midnight **Wednsday, June 10**.