

CS281B/Stat241B Homework Assignment 2 (due Thursday, February 14, 2008)

1. **(Implementing an SVM classifier)** Investigate the performance of SVM classifiers on two classification problems. There are several public domain implementations of SVMs available on the web; see, for example, the list at <http://www.kernel-machines.org/software.html>. Consider two data sets:

- Simulate a two-class classification problem that is easy to visualize. For example, consider a uniform distribution on a subset of \mathbb{R}^2 , with $\Pr(Y = 1|X = x) = 0.05 + 0.9 \times \mathbf{1}[w^T x > 0]$. Use a soft-margin SVM. Investigate the effects of the kernel and the regularization coefficient (for example, the constant C in the standard formulation) on the decision boundary, the number of support vectors obtained, and the misclassification probability.
- Download a public domain data set for a binary classification problem that interests you. Such data sets are available in many places, such as the UCI repository, <http://archive.ics.uci.edu/ml/>. Compare the performance of an SVM classifier with that of a pattern classification technique based on a probabilistic model, such as logistic regression. Split the data and use a hold-out set to estimate the misclassification probability of the classifiers.

2. **(Kernels)**

- Consider the constant function, $k(x, y) = c$ for all x, y . Is k a symmetric positive semidefinite kernel? If not, explain why not. If so, describe its reproducing kernel Hilbert space.
- For two symmetric, positive semidefinite kernels k_1, k_2 defined on the same space, let k be their minimum, $k(u, v) = \min\{k_1(u, v), k_2(u, v)\}$. Is k also a symmetric positive semidefinite kernel?
- Consider a symmetric, positive semidefinite kernel, k defined on \mathcal{X} , and let K denote the kernel matrix corresponding to the set $\{x_1, \dots, x_n\} \subset \mathcal{X}$ (that is, $K_{i,j} = k(x_i, x_j)$). Suppose that K has full rank. For $u \in \mathcal{X}$, define $k_x(u) = (k(u, x_1), \dots, k(u, x_n))'$, and for $u, v \in \mathcal{X}$, define

$$\tilde{k}(u, v) = k(u, v) - k_x(u)' K^{-1} k_x(v).$$

Show that \tilde{k} is also a symmetric, positive semidefinite kernel.

3. **(Constrained optimization)** The ν -support vector regression method involves the following optimization problem.

$$\begin{aligned} & \text{minimize}_{w \in \mathbb{R}^d, \epsilon \in \mathbb{R}} && \frac{1}{2} \|w\|^2 + C \left(\nu \epsilon + \frac{1}{n} \sum_{i=1}^n (|y_i - w'x_i| - \epsilon)_+ \right) \\ & \text{subject to} && \epsilon \geq 0. \end{aligned}$$

where $x_1, \dots, x_n \in \mathbb{R}^d$, $y_1, \dots, y_n \in \mathbb{R}$, and $C, \nu > 0$.

The representer theorem shows that the solution satisfies $w = \sum_{i=1}^n \alpha_i x_i$ for some $\alpha_1, \dots, \alpha_n$.

(a) Show that, if $\epsilon > 0$ at the solution, we have

$$|\{i : |w'x_i - y_i| > \epsilon\}| \leq \nu n \leq |\{i : \alpha_i \neq 0\}|.$$

(b) Show that the optimization is equivalent to the following.

$$\begin{aligned} & \text{minimize}_{\alpha \in \mathbb{R}^n} && \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j x_i' x_j - \sum_i \alpha_i y_i \\ & \text{subject to} && \alpha \in S_{C,\nu,n}, \end{aligned}$$

where, for some radius $r(C, \nu, n)$, $S_{C,\nu,n}$ is a certain subset of the l_1 ball,

$$S_{C,\nu,n} \subset \{\alpha : \|\alpha\|_1 \leq r(C, \nu, n)\},$$

with $\|\alpha\|_1 = \sum_i |\alpha_i|$. How does the parameter ν affect $r(C, \nu, n)$?

4. **(Fast approximate primal SVMs)** The support vector machine is based on the following optimization problem.

$$\text{minimize}_{w \in \mathbb{R}^d} \quad \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n (1 - y_i w' x_i)_+ \quad (1)$$

where $x_1, \dots, x_n \in \mathbb{R}^d$, $y_1, \dots, y_n \in \{\pm 1\}$, and $C > 0$. The usual way to express this as a quadratic program is to introduce n slack variables, $\xi_i \geq 0$, and n constraints. An alternative is to introduce a single slack variable and 2^n constraints.

- (a) Show that the SVM optimization is equivalent to the following QP.

$$\begin{aligned} \text{minimize}_{w \in \mathbb{R}^d, \xi \in \mathbb{R}} \quad & \frac{1}{2} \|w\|^2 + C\xi \\ \text{subject to} \quad & \forall b \in \{0, 1\}^n, \xi \geq \frac{1}{n} \sum_{i=1}^n b_i (1 - y_i w' x_i). \end{aligned} \quad (2)$$

It turns out that we can find approximate solutions to this exponentially large optimization problem in time linear in n . The key insight is that the 2^n constraints are not all equally important.

Consider an algorithm that starts with $B_1 = \{0\} \subset \{0, 1\}^n$, and at iteration t , solves the QP

$$\begin{aligned} \text{minimize}_{w_t \in \mathbb{R}^d, \xi_t \in \mathbb{R}} \quad & \frac{1}{2} \|w_t\|^2 + C\xi_t \\ \text{subject to} \quad & \forall b \in B_t, \xi_t \geq \frac{1}{n} \sum_{i=1}^n b_i (1 - y_i w_t' x_i). \end{aligned} \quad (3)$$

and then sets $B_{t+1} = B_t \cup \{b\}$, where $b \in \{0, 1\}^n$ corresponds to the constraint in (2) that requires the largest value of ξ to make a feasible pair (w_t, ξ) , that is, it adds the b that maximizes

$$J_t(b) := \frac{1}{n} \sum_{i=1}^n b_i (1 - y_i w_t' x_i).$$

The algorithm continues until $J_t(b) - \xi_t \leq \epsilon$, (ϵ is a parameter of the algorithm).

- (b) Show that the solution (w_t, ξ_t) returned by this algorithm has a smaller objective than the solution of (2), and that $(w_t, \xi_t + \epsilon)$ is feasible in (2). Hence show that w_t is an approximate solution to (1).

It turns out (we won't prove this), that, under some mild assumptions on the data, this algorithm always terminates after some number of steps T that depends on ϵ but not on n or d . Since the QP (3) can be solved in time polynomial in the number of variables and constraints, this means that, for fixed ϵ and d , the algorithm runs in time $O(n)$.

- (c) Suppose now that the data is sparse, that is, many of the components of the x_i are zero. Define

$$s = \frac{1}{n} |\{(i, j) : 1 \leq i \leq n, 1 \leq j \leq d, x_{i,j} \neq 0\}|,$$

where $x_i = (x_{i,1}, \dots, x_{i,d})$. Assume that $s \ll d$ (that is, the data is sparse), and that d increases with n such that $sn \geq d$ (because we do not bother to include features in the x_i that are always zero). Describe a version of the algorithm that runs in time $O(sn)$ for fixed ϵ .