Support Vector Machine

UCLA Math 156

Spring 2016

1 Overview

"In today's machine learning applications, support vector machines are considered a must try – it offers one of the most robust and accurate methods among all well-known algorithms." [1]

The support vector machine is a significant building block in the machine learning field. It solves a fundamental problem very well: separate two classes of data as far as possible. In this lab you will write code which calculates a separating hyperplane in a robust and reasonably efficient way using a formulation of the soft SVM as a quadratic program, easily solved using the built-in MATLAB routine `quadprog`. You are then ready to apply the method to two data collections: one for which we would like to determine if an image is of a human face or not, and a second data set in which documents are taken from two online newsgroups about space and cryptography and we would like to be able to determine which group a document belongs to.

2 Provided Resources

- `imgrid.m` - This helper function takes a matrix of observations (first argument) and the dimensions of each observation (second argument) and displays a grid (with size given by the third argument) of observations. May be useful for viewing data.
- `cbcl.mat` - Data set with small images of human faces in one class and random images, not of faces, in the other class. Variables are formatted as in `mnist.mat`.
- `news.mat` - Data set containing the word histograms for documents from the 20 newsgroups corpus, specifically space and encryption newsgroups, with both `X` and `L` as in the other data sets. In addition, `dict` is contained which indicates what word each row of `X` corresponds to. For example, if the second element of `dict` is the word "cheese" and the fourth document contained the word cheese ten times, $\texttt{X}_{2,4} = 10$.
- `softsvm.m` - Function to be completed which learns a linear classifier using the support vector machine with slack variables.

3 Guide

1. (5) Implement the soft support vector machine. To do this you will be using the MATLAB function `quadprog` which solves a common problem known as a quadratic program.

   We start from the soft SVM problem stated in class/homework,

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + \gamma \sum_n \xi_n \qquad \text{s.t.} \qquad \begin{cases} t_n(w^T\phi(x_n) + b) \geq 1 - \xi_n \\ \xi_n \geq 0 \end{cases}, \quad \forall n.$$

[1] "Top 10 algorithms in data mining" by X. Wu et al. DOI 10.1007/s10115-007-0114-2

Assuming that the basis functions ϕ are just identity (i.e. no non-linear transform), we can rewrite this as a quadratic program in Matlab-convenient notation as follows (check!):

$$\min_{\xi,w,b} \frac{1}{2} \begin{pmatrix}\xi^T & w^T & b\end{pmatrix} \begin{pmatrix}0 & 0 & 0\\ 0 & I_D & 0\\ 0 & 0 & 0\end{pmatrix} \begin{pmatrix}\xi\\ w\\ b\end{pmatrix} + \begin{pmatrix}\gamma^T & 0 & 0\end{pmatrix} \begin{pmatrix}\xi\\ w\\ b\end{pmatrix}$$

$$\begin{pmatrix}-I_N & -TX & -t\end{pmatrix} \begin{pmatrix}\xi\\ w\\ b\end{pmatrix} \leq -1$$

$$\begin{pmatrix}0\\ -\infty\\ -\infty\end{pmatrix} \leq \begin{pmatrix}\xi\\ w\\ b\end{pmatrix}.$$

The variables in the problem are as follows (here D is the dimension of the data space, and N is the number of data points):

- ξ - A length N vector with slack variables ξ_n, one for each observation.
- w - A length D vector with the normal for the separating hyper-plane.
- b - A scalar indicating the separating hyper-plane offset coefficient.
- I_D, I_N - An identity matrix of dimension D or N, respectively.
- γ - A column vector with the slack penalty parameter repeated N times.
- T - An N-by-N diagonal matrix with either 1 or -1 on the diagonal, depending on the class assignments for the data points. The diagonal is equal to the vector t.
- t - A length N column vector with class labels t_n (input to our function). Equal to the diagonal of T.
- X - An N-by-D matrix with one data point x_n in each row (input to our function, transposed).

Here, the block matrix

$$\begin{pmatrix}0 & 0 & 0\\ 0 & I_D & 0\\ 0 & 0 & 0\end{pmatrix}$$

is a $(N+D+1)$-by-$(N+D+1)$ matrix with a diagonal given by N zeros, D ones, and a single final 0. Additionally, the vector

$$\begin{pmatrix}\xi\\ w\\ b\end{pmatrix}$$

is a column vector of length $N+D+1$ containing all the parameters to be learned stacked end-to-end. The solution to our problem, to be returned, is w and b which define the separating hyper-plane $\{\langle w, x\rangle + b = 0\}$.

Note: in older versions of MATLAB (if a warning shows up), you will need:

```
opts = optimset('Algorithm','interior-point-convex');
xi_w_b = quadprog(H,f,A,b,[],[],lb,[],[],opts);
```

In newer versions this behavior is default, and you can directly call

```
xi_w_b = quadprog(H,f,A,b,[],[],lb);
```

Your task is essentially to populate the parameters `H, f, A, b, lb` appropriately.

2. (4) Load the CBCL dataset (check for dimensions of `X`, labels in `L`) and apply the soft SVM classifier with a penalty $\gamma = 0.005$. Generate and turn in a visualization of w, as found by the SVM function, using the command `imagesc(reshape(w, dims))` (here `dims` comes from the original data file). Explain what you see.

3. (4) Generate a plot of `X'*w + b` against `L`, the correct labels. What do the extremes (minimum/maximum) of this plot represent? Were any data points classified incorrectly, and how can you tell?

4. (4) Turn in two images corresponding to the extreme points of this plot, and two more images corresponding to example support vectors from each class.

5. Load the 20 Newsgroups data set (check for dimensions of `X`, labels in `L`) and apply the soft SVM with $\gamma = 0.005$.

6. (4) By examination of the vector w, which words are the most important for separating the two classes of documents? Which words are most distinctly space-related? What about cryptography-related? Give at least five important words for each case.

7. (4) Is the 20 Newsgroups data linearly separable? How do you know?