Machine Learning

Lab 7: SVM

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

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`. (Note: typically, one does not solve this primal problem, but the associated dual, in particular when kernels are being used). 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

- `cbcl.mat` - Data set with small images of human faces in one class and random images, not of faces, in the other class. Note: this is a different `cbcl.mat` than we used for PCA!
- `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. Each data point (column of `X`) is a vector whose components indicate how many times a certain words appears in that document. In addition, `dict` is included, 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, `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. Implement (and include in your report) the soft margin support vector machine as function `softsvm`. To do this you will be using the MATLAB function `quadprog` which solves a common problem known as a quadratic program. The SVM with slack variables in a form identifiable as a quadratic program is

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

$$\text{s.t.} \qquad \begin{pmatrix}-I_N & -LX^{\top} & -\ell\end{pmatrix}\begin{pmatrix}\xi\\ \mathbf{w}\\ b\end{pmatrix}\leq -1, \qquad \begin{pmatrix}0\\ -\infty\\ -\infty\end{pmatrix}\leq\begin{pmatrix}\xi\\ \mathbf{w}\\ b\end{pmatrix}.$$

   Note that some of the zeros stand for entire rows/columns/blocks of zeros, as appropriate.

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, one for each observation.
- $\mathbf{w}$ — 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 — Identity matrix of dimension D or N, respectively.
- γ — A column vector with the slack penalty parameter repeated N times.
- L — 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 ℓ.
- ℓ — A length N column vector with class labels (input to our function). Equal to the diagonal of L.
- X — An D-by-N matrix with one data point in each column.

For example, the 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 matrix

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

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 $\mathbf{w}$ and b which define the separating hyper-plane $\{\mathbf{x} \mid \mathbf{w}^\top \mathbf{x} + b = 0\}$.

Important: some of these variables are quite big (but mostly zeros). For most efficient computation (and it is still going to take tens of seconds!), make sure to use sparse matrices whenever possible: use `speye` and `spdiags` to construct sparse matrices and build from there!

2. Load the CBCL dataset and apply the soft SVM classifier with a penalty $\gamma = 0.005$. Generate and turn in a visualization of $\mathbf{w}$, as found by the SVM function, using the command `imagesc(reshape(w, dims))` (here `dims` comes from the original data file). What does this picture of $\mathbf{w}$ represent? How do you interpret it?

3. Generate a plot of `X'*w + b` (overlay with and compare to the plot of `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. How can we determine that a data point was a support vector?

5. Turn in two images corresponding to the extreme points of this plot (most positive/negative scoring column of X shown as images), and two more images corresponding to example support vectors from each class. Discuss what you observe!

6. Load the 20 Newsgroups data set and apply the soft SVM with $\gamma = 0.005$.

7. By examination of the vector $\mathbf{w}$, 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.

8. Extra: Is the 20 Newsgroups data linearly separable? How do you know/could you find out?