CS 545 ASSIGNMENT 3

Paresh Bhambhani

October 18, 2015

Contents

1 SVM with no bias term

The original soft margin primal as discussed in class is:

$$\min_{\mathbf{w},b}\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to: } y_i(\mathbf{w}^\intercal\mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ \ i = 1,\ldots,n\,. \tag{1}$$

The dual for this primal is:

$$\max_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\intercal\mathbf{x}_j \tag{2}$$

$$\text{subject to: } \alpha_i \geq 0,\ \ \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{3}$$

The constraint on the dual is that $\sum_{i=1}^{n}\alpha_i y_i = 0$, the optimizer must change at least two $alpha$ values just as SMO solver.

Now we take a look at the soft margin primal without a bias term present. Hence we have $f(x) = \mathbf{w}^\intercal x$. The primal now becomes:

$$\min_{\mathbf{w}}\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to: } y_i(\mathbf{w}^\intercal\mathbf{x}_i) \geq 1 - \xi_i,\ \xi_i \geq 0,\ \ i = 1,\ldots,n\,. \tag{4}$$

We get the Lagrangian as:

$$\Lambda(\mathbf{w}, \alpha, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - \xi_i - y_i\mathbf{w}^\intercal x_i) - \sum_{i=1}^{n}\beta_i\xi_i \tag{5}$$

To get the saddle point, we calculate the partial derivatives of the Lagrangian and equate them to zero.

$$\begin{align}
\frac{\partial\Lambda}{\partial\mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \tag{6}\\
\Rightarrow \mathbf{w} &= \sum_{i=1}^{n}\alpha_i y_i x_i \tag{7}\\
\frac{\partial\Lambda}{\partial\xi} &= C - \alpha_i - \beta_i = 0 \tag{8}
\end{align}$$

The KKT conditions are:

$$\begin{align}
\mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i x_i &= 0 \tag{9}\\
C - \alpha_i - \beta_i &= 0 \tag{10}\\
1 - \xi_i - y_i\mathbf{w}^\intercal x_i &\leq 0 \tag{11}\\
\alpha &\geq 0 \tag{12}\\
\alpha_i(1 - \xi_i - y_i\mathbf{w}^\intercal x_i) &= 0 \tag{13}
\end{align}$$

To calculate the dual, replace w from the Saddle point conditions into the Lagrangian.

$$\begin{align}
W(\alpha) = \frac{1}{2}\left(\sum_{i=1}^{n}\alpha_i y_i x_i\right)^\intercal\left(\sum_{i=1}^{n}\alpha_i y_i x_i\right) + C\sum_{i=0}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i \tag{14}\\
- \sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n}\alpha_i y_i\left(\sum_{j=1}^{n}\alpha_j y_j x_j\right)^\intercal x_i - \sum_{i=1}^{n}\beta_i\xi_i \tag{15}\\
= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\intercal x_j \tag{16}
\end{align}$$

The resulting dual is:

$$\begin{align}
\max_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\intercal x_j \tag{17}\\
\text{subject to: } \alpha_i \geq 0,\ \ i = 1, \ldots, n\,. \tag{18}
\end{align}$$

This is almost same as the dual we obtained for the SVM with the bias. The only difference being in the constraint condition. The restriction $\sum_{i=1}^{n}\alpha_i y_i = 0$ is absent. Since this condition is not imposed, the optimizer can vary individual values of α without adjusting at least another one to compensate for it. Hence the SMO can work with changing only a single α.

2 Soft-margin SVM for separable data

The answer to the first question is False. Since our data set is linearly separable, there is a hyperplane that definitively divides the data into positive and negative points. The C is also given to be positive which

implies there are few samples that are within the margins. Our optimal solution will be maximizing the margins while accounting for the non-zero $C\sum_{i=1}^{n}\xi_i$ entries, which implies that they are not all zero.
The answer to the second question is No, it is not always better to use the hard margin SVM when we have linearly separable data. There can sometimes be a few outliers within the margin. In this case if we use a hard margin SVM then the support vectors will be pushed too close to the margin. As explained in class, ”we don’t necessarily want to give too much weight to what is happening close to the Margin”. Hence its advisable to give the examples some slack and by ignoring the points too close to the margin, we can get a larger margin and hence a better classifier.

3 Using SVMs

The accuracy plot of the SVM against the soft-margin parameter of the SVM and the free parameter of the kernel function for the SVM with Polynomial kernel and Gaussian kernel is as shown in Figure 1 and Figure 2 respectively.

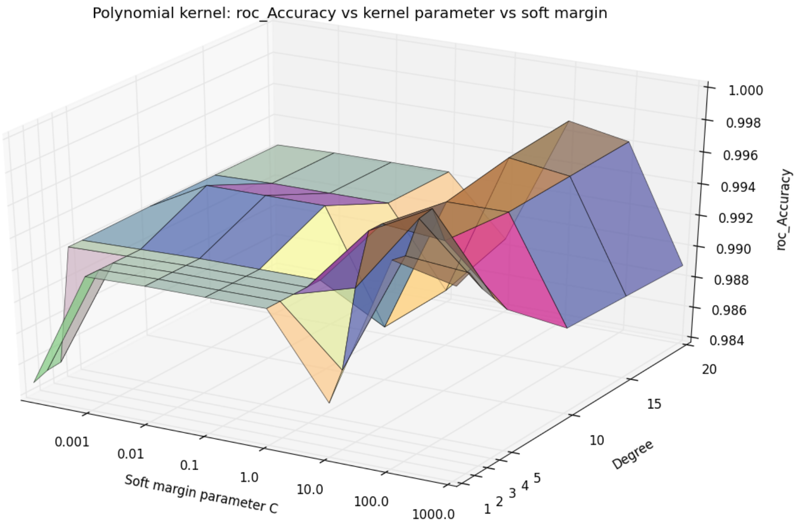

Figure 1: Polynomial Kernel: Accuracy vs Soft margin parameter vs Kernel parameter

The range of values for the soft margin, polynomial kernel and gaussian kernel parameters are taken as, C=[0.0001,0.001,0.01,0.1,1,10,100,1000], p=[1,2,3,4,5,10,15,20] and γ=[0.0001,0.001,0.01,0.1,1,10,100,1000] respectively. As mentioned in the problem, the values should be distributed on an exponential scale, the polynomial kernel parameter being an exception to this.

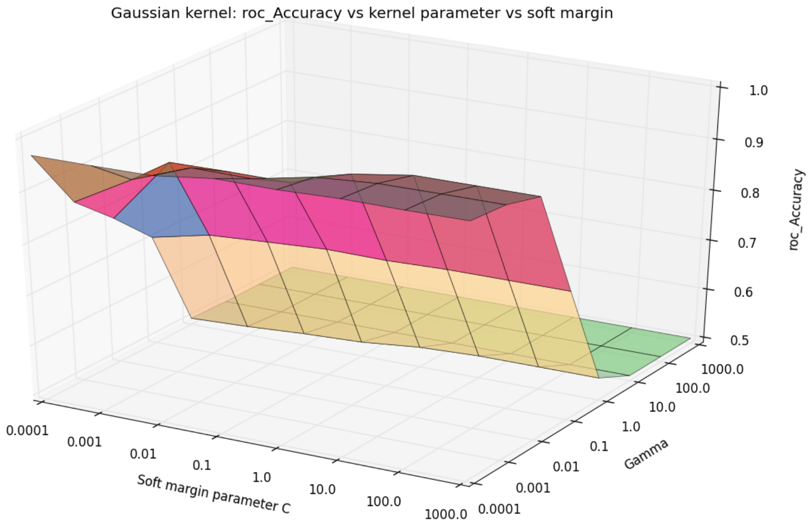

Figure 2: Gaussian Kernel: Accuracy vs Soft margin parameter vs Kernel parameter

The cross section plots of the above 3-D plots for all the values of soft margin parameter (considered above) and all the values of the kernel parameter (considered above) is shown in Figure 3. Instead of showing the cross section for just one soft margin parameter or one kernel parameter, showing it for all the values of the soft margin parameter and kernel parameters used gives a better insight into the behavior of all the values.

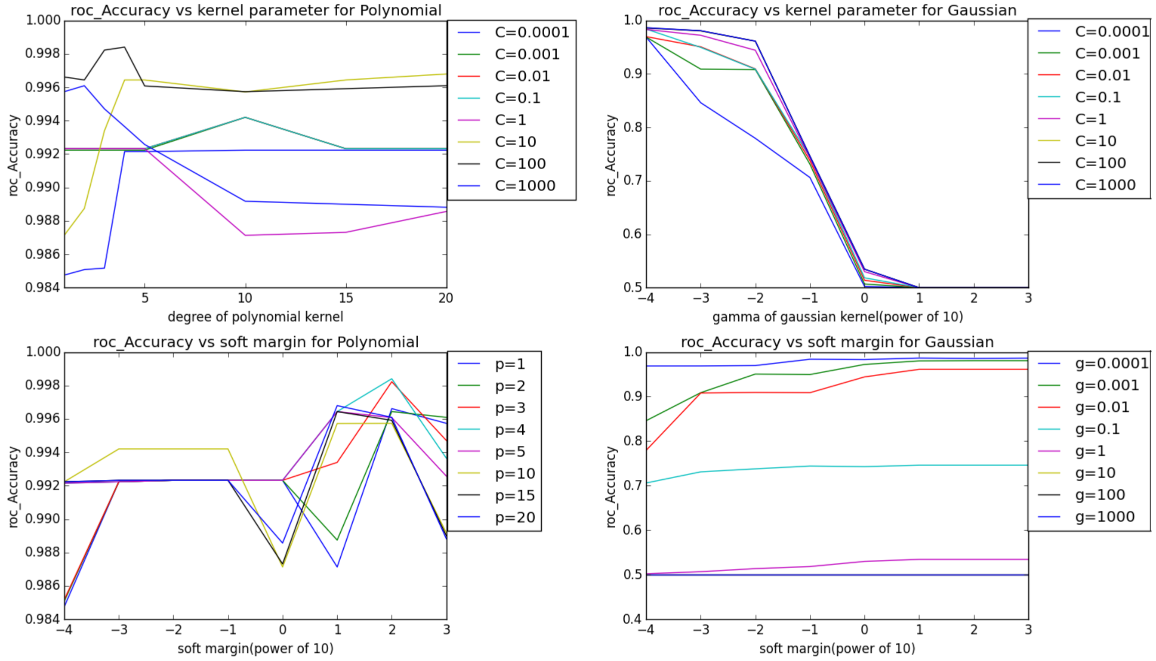

Figure 3: Cross-Sectional Plots

For the accuracy plot of the Poly kernel SVM, the degree values (Kernel parameter) was incremented one at a time and later 5 at a time to see the performance for higher degree polynomials. The accuracy numbers are high despite any variations in the soft margin parameter or the kernel parameter. However there is a slight dip in the accuracy seen when ”C” is 1 and the polynomial degree is high (10,15,20) and again when ”C” is 1000, although the accuracy numbers are still very high. Overall the SVM with the polynomial kernel gives very good ROC accuracy for this data. This behavior can be observed in both the 3-D plot and the 2-D cross section plots. From the plot for accuracy vs the Kernel parameter for the Polynomial Kernel it can also be observed that the SVM with kernel parameter C = 100 gives the best overall accuracy for varying polynomial degrees.

For the accuracy plot of the Gaussian kernel SVM it is seen that as we increase the value of the kernel parameter the accuracy falls and becomes constant at 0.5. This is seen in the 3-D plot as well as the cross sectional plot of accuracy vs the Gaussian kernel parameter. Larger the gamma, tighter the fit of the margin to the dataset would be. Hence for very large gamma values, the classifier accuracy drops to 50%. The plot of accuracy vs the kernel parameter γ shows that we get high accuracy results when the value of soft margin parameter is high (10,100,1000).

Next, we will compare the accuracy of an SVM with a Gaussian kernel on the raw data with accuracy obtained when the data is normalized to be unit vectors.

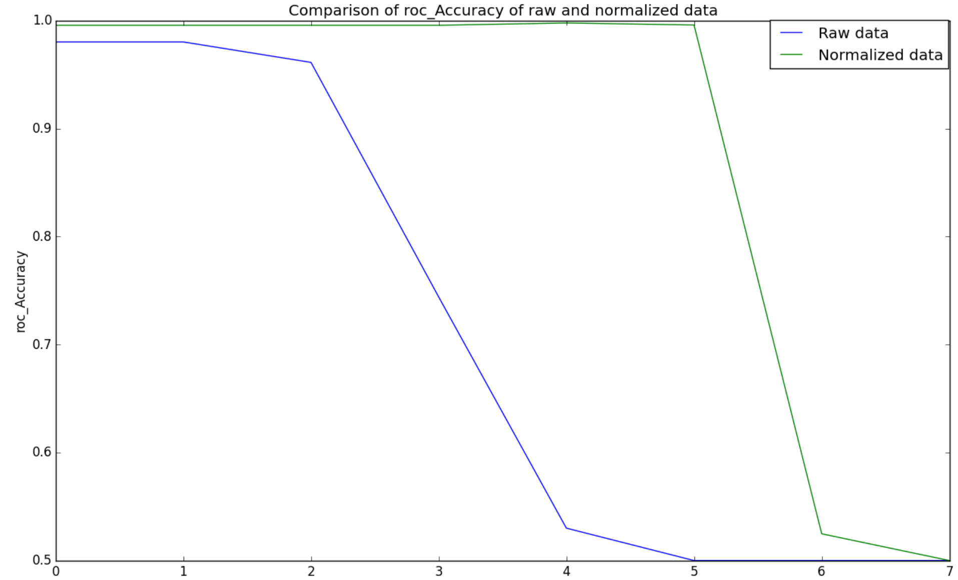

Figure 4: Accuracy variations of Raw and Normalized data

Figure 4 shows the variation of accuracy for both, Raw data and normalized data. For generating this comparison, the optimal parameters were chosen using a Grid search. To get an insight into the behavior of the change in accuracy, the parameters on which the grid search is performed, are varied. We first begin with all the values of the soft margin parameter, ”C” and kernel parameter ”gamma”, which are used in the above section of the assignment. We then iteratively remove the smallest values from both the parameters set and perform the Grid search on the remaining and then calculate the accuracy from 5 fold cross validation. This is done for the ”best parameters” to change to see how Raw data and Normalized data perform. The first term is removed iteratively because lower the gamma is, better is the performance. So if we were to remove the last terms, the ”best parameters” values would not have changed.

The resulting graph is shown in Figure 4. The Y axis shows the accuracy and X axis is analogous to how many values from the Kernel and soft margin SVM parameters set are removed for the Grid search. We can see from the plot that the Normalized data always has better accuracy than raw data. Both show a drop as we remove the best performing parameters from the grid search "ParamGrid" but still the normalized data retains its performance as better than the raw data untill they both drop down to 50% accuracy.

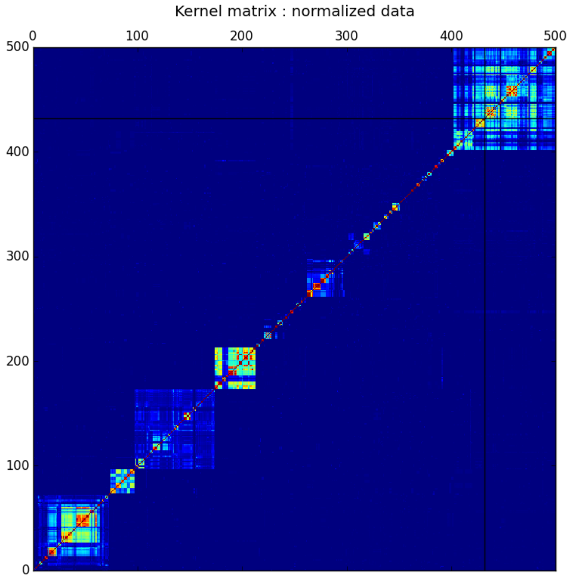

Figure 5: Kernel Matrix for Normalized data

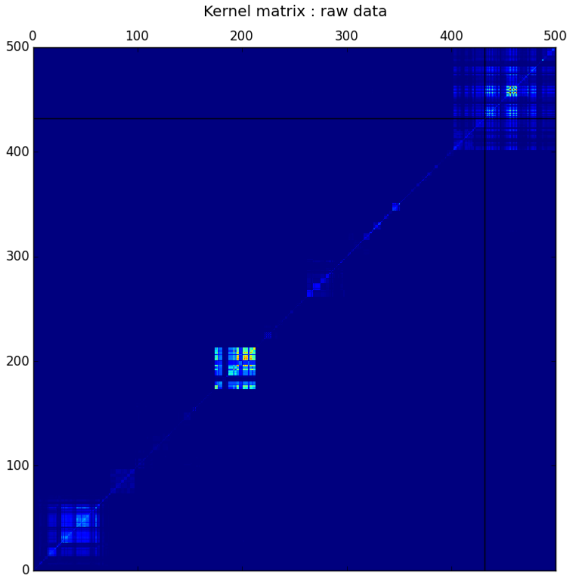

Figure 6: Kernel Matrix for Raw data

Now we will visualize the kernel matrix associated with the dataset. For this we will make use of the PyML module because of its ease in performing the required functions. Figure 5 and Figure 6 show the Kernel matrix for the normalized data and the raw data respectively.
The high amount of the blue area in both, shows the sparsity of the data. Red color corresponds to data being highly similar and as the color ″Heat″ decreases the similarity decreases with blue being the least. Most prominent difference between the two maps of raw and normalized data is the increase in the similarity of examples in the normalized data as against the raw data. One can notice more block of similarities along the diagonal in the normalized data. Since this is a sparse dataset, each example may have different number of features with a non zero value. Since different features will have different range of values, the comparison of two examples with not all same features will not be entirely correct. But once the data is normalized (example wise and not feature wise) now it makes more sense to see the similarity between the examples. Hence the reason for seeing more blocks of similarity in the normalized data than raw data and appearance of the red line along the diagonal in the normalized data map. Another point to be noted is that the small similarity of the examples in raw data is now augmented in the normalized data and they are shown by hotter colors. This implies that after normalization the similarity or the correspondence of the already similar examples, increases.