Machine Learning

Homework 8 solution - Shyam Arumugaswamy

1.

Generally, for a linearly separable dataset, we use hard margin SVM and when the dataset is not linearly separable due to some noise, we use soft margin SVM by introducing slack variables to SVM objective function to allow the error in the misclassification In this problem, when we have linearly separable dataset, all the training samples may not be correctly classified by soft margin SVM as noise may be present which may cause misclassification. Soft margin SVM has same margin maximizing objective like hard margin SVM, but with additional constraint that each lagrange multiplier associated with support vector is bounded by C. Soft margin SVM could choose decision boundary that has non-zero training error even if data is linearly separable and is less likely to overfit. As value of C decreases, it causes the classifier to sacrifice linear separability in order to gain stability, in a way that influence of any single data point is bounded by C.

2.

The slack penalty C in soft margin SVM sets the relative importance of maximizing the margin and minimizing the amount of slack. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly and will act like hard-margin SVM when C --> ∞. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, we will get misclassified classes, even if our training data is linearly separable. If C values changes within a reasonable range, the optimal hyperplane will just randomly shift by a small amount within the margin(the gap formed by the support vectors).Intuitively, suppose the margin on training data is small, and/or there is no test data points within the margin too, the shifting of the optimal hyperplane within the margin will not affect classification error of the test set. If we set C=0, then SVM will ignore the errors and data entirely, and just try to minimise the sum of squares of the weights(w). Hence C should be greater than 0.

3.

$K(x,y) = (x^T y + c)^d$

$x^T y$ is inner product in n-dimensional Euclidean space and so $x^T y$ is a mercer kernel

constant c >= 0 is trivially positive semi-definite,

So by rule 1 (i.e. sum of two kernels is a kernel), we can say that $(x^T y + c)$ is also a Kernel.

Since $(x^T y + c)^d$ is equivalent to $(x^T y + c)$ multiplied d times, from rule 3 (i.e. product of two kernels is a kernel), we can conclude that $(x^T y + c)^d$ is a valid kernel.

4.

We cannot apply the mentioned or any other infinite feature transformation to the data as there will be problems due to finite space and time on computing machines. But if we consider lazy evaluation, we can apply in different terms.

5.

$K(x,y) = \sum_{i=1}^{\infty} \phi_{\infty,i}(x)\, \phi_{\infty,i}(y)$

Using Taylor's series,

$$K(x,y) = e^{\frac{-x^2-y^2}{2\sigma^2}} \sum_{i=1}^{\infty} \frac{(\frac{xy}{\sigma^2})^k}{k!}$$

$$= e^{\frac{-x^2-y^2}{2\sigma^2}}\; e^{\frac{xy}{\sigma}}$$

$$= e^{-\frac{(x-y)^2}{2\sigma^2}}$$

In an infinite-dimensional feature space, overfitting is a problem when using linear classifier without maximum margin property. Expected loss increases with dimensionality of feature space. For linear maximum margin classifiers, learning theory shows that generalisation performance does not depend on dimensionality of feature space. Overfitting might still be a problem as this does not take into account noise in the input. As maximum margin property limits the acceptable hypothesis for training set, it can be considered as a regularizer.

6.

If the limit $\sigma \rightarrow 0$, kernel function is

$$K(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

For any finite set of points, Kernel matrix is identity matrix K = I

All training samples are classified correctly if

$y_i(\omega^T \phi(x_i) + b) > 0$ for all i

By replacing $\phi(x_i)^T\ \phi(x_j)$ by Kernel $K(x_i, x_j)$ and

substituting dual representation $\omega = \sum_j \alpha_j \phi(x_j)$ we get

$y_i (\sum_j \alpha_j K(x_i, x_j) + b) > 0$

using $K(x,y)= 1$ for $x=y$ condition for i,j

$y_i (\alpha_i + b) > 0$

$y_i^2 > 0$ when we put $y_i = \alpha_i$ and $b = 0$

and this is fulfilled for all training samples.

Hence if a variance is chosen small enough, all finite set of points can be linearly separated using a gaussian Kernel.

7.

The distance to training sample is $|\phi(x) - \phi(x^{(Si)})|_2$

Replacing this by squared distance:

$$|\phi(x) - \phi(x^{(Si)})|_2^2 = (\phi(x) - \phi(x^{(Si)}))^T \phi(x) - \phi(x^{(Si)})$$
$$= \phi(x)^T \phi(x) - 2\, \phi(x)^T \phi(x^{(Si)}) + \phi(x^{(Si)})^T \phi(x^{(Si)})$$

As the first term is a constant when searching for k training samples, we drop this term and find the k training samples that minimize

$\phi(x^{(Si)})^T \phi(x^{(Si)}) - 2\, \phi(x)^T \phi(x^{(Si)}) = K(x^{(Si)}, x^{(Si)}) - 2\, K(x, x^{(Si)})$