Machine Learning Assignment 8

Shivangi Aneja

18-December-2017

Problem 1

Given a linearly separable dataset $\mathcal{D} \Rightarrow$ Soft Margin SVM

The cost function for Soft Margin SVM is given by:
$f_0(w, b, \xi) = \frac{1}{2}w^T w + C \sum_{i=1}^{N} \xi_i$

And the constraints are given by
$y_i(w^T x_i + b) - 1 + \xi_i \geq 0 \; i = 1, ..., N$
$\xi_i \geq 0 \; i = 1, ..., N$

The definition of Soft Margin SVM says that we try to minimize $f_0(w, b, \xi)$ by allowing some errors(points to be misclassified) to formulate a good generalization model. The penalty we impose on misclassified cases (error) depends on value of C.

(1) $\xi_i = 0 \Rightarrow$ These data points are either on the margin or on the correct side of the margin. These data points are classified correctly.
(2) $0 < \xi_i \leq 1 \Rightarrow$ These data points lie inside the margin but on the correct side of the decision boundary.These data points are also classified correctly.
(3) $\xi > 1 \Rightarrow$ These data points lie on the wrong side of the decision boundary. These points are thus misclassified

Thus, to conclude it is not guaranteed that all training samples in $\mathcal{D}$ are classified correctly. Some of them can be misclassified depending on the value of ξ

Problem 2

The cost function for Soft Margin SVM is given by:
$f_0(w, b, \xi) = \frac{1}{2}w^T w + C \sum_{i=1}^{N} \xi_i$

Our goal is to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary.

C controls tradeoff between slack variable penalty and margin.

(1) Large Values Of $C \Rightarrow$ Chooses a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.
$C \to \infty$ means that we impose infinite penalty on misclassified cases and it is thus a Hard Margin SVM
(2) Small Values Of $C \Rightarrow$ Chooses for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.For very tiny values of C, we get misclassified cases, often even if training data is linearly separable
C $=$ 0 means we impose no penalty on the misclassified cases.And in this case it is possible that entire data set is misclassified.

Thus to conclude $C > 0$ to impose at least some penalty on misclassified cases otherwise the model is too general and always misclassifies the data points.

Problem 3
The Kernel is given by:
$k(x, y) = (x^T y + c)^d$ where $c \geq 0$ and $d \in \mathbb{N}^+$

To prove a kernel is valid:

(1) Symmetric : $k(x, y) = k(y, x)$
$x = (x_1, x_2, ..., x_N)$ and $y = (y_1, y_2, ..., y_N)$
$k(x, y) = (x^T y + c)^d = (x_1 y_1 + x_2 y_2 + ...x_N y_N + c)^d......(1)$
$k(y, x) = (y^T x + c)^d = (x_1 y_1 + x_2 y_2 + ...x_N y_N + c)^d......(2)$
From (1) and (2) it is clear that $\boxed{k(x, y) = k(y, x)}$
Thus, it is symmetric

(2) Kernel Matrix $\mathbf{K}$ is positive semi-definite
For some vector $\mathbf{z}$ we have ,
$\boldsymbol{z^T K z} = \sum_i \sum_j z_i K_{ij} z_j = \sum_i \sum_j z_i \boldsymbol{x_i} \boldsymbol{x_j} z_j = \sum_i z_i \boldsymbol{x_i} \sum_j z_j \boldsymbol{x_j} = || \sum_i z_i \boldsymbol{x_i} ||^2 \geq 0$
$\Rightarrow \boldsymbol{z^T K z} \geq 0$
Thus it symmetric

Hence, it is a valid kernel

Problem 4

$$\phi_n(x) = \left\{ e^{-x^2/2\sigma^2}, e^{-x^2/2\sigma^2}\frac{x}{\sigma}, \frac{e^{-x^2/2\sigma^2}\left(\frac{x}{\sigma}\right)^2}{\sqrt{2}}, ..., \frac{e^{-x^2/2\sigma^2}\left(\frac{x}{\sigma}\right)^n}{\sqrt{n!}} \right\}$$

Suppose, $n \to \infty$ we have,

$$\phi_\infty(x) = \left\{ e^{-x^2/2\sigma^2}, e^{-x^2/2\sigma^2}\frac{x}{\sigma}, \frac{e^{-x^2/2\sigma^2}\left(\frac{x}{\sigma}\right)^2}{\sqrt{2}}, ..., \frac{e^{-x^2/2\sigma^2}\left(\frac{x}{\sigma}\right)^i}{\sqrt{i!}}, ... \right\}$$

This transformation maps feature space defined as $\phi_\infty : \mathbb{R} \to \mathbb{R}^\infty$.
We have an infinite dimensional feature space. It is not possible to store it in memory. Thus it is infeasible to compute in this feature space.We do not calculate $\phi_\infty(x)$ directly, instead use kernel trick to solve this kind of problem.

> For a given pair of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
> $k(\boldsymbol{x}, \boldsymbol{y}) = \phi_\infty^T(\boldsymbol{x})\phi_\infty(\boldsymbol{y}) = C(\text{scalar})$

Thus, we compute the dot product in the higher-dimensional space without explicitly transforming the vectors into the higher-dimensional space first and use them

Problem 5

$$\phi_\infty(x) = \left\{ e^{-x^2/2\sigma^2}, e^{-x^2/2\sigma^2}\frac{x}{\sigma}, \frac{e^{-x^2/2\sigma^2}\left(\frac{x}{\sigma}\right)^2}{\sqrt{2}}, ..., \frac{e^{-x^2/2\sigma^2}\left(\frac{x}{\sigma}\right)^i}{\sqrt{i!}}, ... \right\}$$

$$K(x, y) = \sum_{i=0}^{\infty} \phi_{\infty,i}(x)\phi_{\infty,i}(y)$$

Putting in the values of x and y , we get
$K(x,y) = \phi_{\infty,0}(x)\phi_{\infty,0}(y) + \phi_{\infty,1}(x)\phi_{\infty,1}(y) + \phi_{\infty,2}(x)\phi_{\infty,2}(y) + ...$

$$K(x,y) = e^{-x^2-y^2/2\sigma^2} + e^{-x^2-y^2/2\sigma^2}\frac{xy}{\sigma^2} + \frac{e^{-x^2-y^2/2\sigma^2}\left(\frac{xy}{\sigma^2}\right)^2}{2!} + ... + \frac{e^{-x^2-y^2/2\sigma^2}\left(\frac{xy}{\sigma^2}\right)^i}{i!}$$
$$K(x,y) = e^{\frac{-x^2-y^2}{2\sigma^2}}\left\{1 + \frac{xy}{\sigma^2} + \frac{\left(\frac{xy}{\sigma^2}\right)^2}{2!} + ... + \frac{\left(\frac{xy}{\sigma^2}\right)^i}{i!}\right\}$$
$$K(x,y) = e^{\frac{-x^2-y^2}{2\sigma^2}}\sum_{n=0}^{\infty}\frac{1}{n!}\left(\frac{xy}{\sigma^2}\right)^n$$

Using the Taylor Expansion $e^Z = \sum_{n=0}^{\infty}\frac{1}{n!}(Z)^n$, we get
$$K(x,y) = e^{\frac{-x^2-y^2}{2\sigma^2}} e^{\frac{xy}{\sigma^2}} = e^{\frac{-x^2-y^2+2xy}{2\sigma^2}} = e^{\frac{-(x-y)^2}{2\sigma^2}}$$
$$\boxed{K(x,y) = e^{\frac{-(x-y)^2}{2\sigma^2}}}$$
For a linear classifier : Overfitting increases as the dimensions of the feature/input space increases
For SVM : If we use the kernel function, then problem of overfitting can be avoided. This is because we do not explicitly calculate these feature space and kernel takes care of it.

Problem 6
Linear Separability in Feature Space defined by ϕ_∞ depending on choice of σ
$K(x,y) = e^{\frac{-(x-y)^2}{2\sigma^2}}$
Consider the following cases :
(a) $\sigma \to 0$
$$K(x,y) = \left\{\begin{array}{ll} 0 & x \neq y \\ 1 & x = y \end{array}\right\}.$$
$\boldsymbol{K}$ is a diagonal matrix with 1 at diagonal and 0 elsewhere
For points to be correctly classified in SVM,
$y_i(\boldsymbol{w}^T\phi(\boldsymbol{x_i}) + b) > 1 \quad \forall i......(1)$
$\boldsymbol{w} = \sum_{j=1}^{N}\alpha_j y_j \phi(\boldsymbol{x}_j)......(2)$

Using (1) and (2), we have

$y_i(\sum_{j=1}^{N} \alpha_j y_j \phi^T(\boldsymbol{x}_j)\phi(\boldsymbol{x_i}) + b) > 1 \quad \forall i$

$\Rightarrow y_i(\sum_{j=1}^{N} \alpha_j y_j K(\phi(\boldsymbol{x_j}), \phi(\boldsymbol{x_i})) + b) > 1 \quad \forall i$

$\Rightarrow y_i((\alpha_i y_i) + b) > 1 \quad \forall i$

Setting $b = 0$ and we know $\alpha_i \geq 0 \quad \forall i$,

$\Rightarrow \alpha_i y_i^2 > 1$

This is true for all the data points.Thus the points can be linearly separated if σ very small

(b) $\sigma \to \infty$

$K(x, y) = 1$

For points to be correctly classified in SVM,

$y_i(\boldsymbol{w}^T\phi(\boldsymbol{x_i}) + b) > 1 \quad \forall i......(1)$

$\boldsymbol{w} = \sum_{j=1}^{N} \alpha_j y_j \phi(\boldsymbol{x}_j)......(2)$

$\Rightarrow y_i(\sum_{j=1}^{N} \alpha_j y_j K(\phi(\boldsymbol{x_j}), \phi(\boldsymbol{x_i})) + b) > 1 \quad \forall i$

$\Rightarrow y_i b > 1$

We know $y_i = \{-1, 1\}$ and $b \in \mathbb{R}$. This condition is not true for all the data points.

For a larger sigma, the decision tends to be flexible and smooth, it tends to make wrong classification while predicting, but avoids the hazard of overfitting. For a smaller sigma, the decision boundary tends to be strict and sharp,and it tends to overfit.

If the distance between x_i and x_j is much larger than σ, the kernel function tends to be zero. Thus, if the σ is very small, only the x_i within the certain distance can affect the predicting point. In other words, smaller sigma tends to make a local classifier, larger sigma tends to make a much more general classifier.

To conclude, $\sigma \to 0$ for all the points to be classified correctly

Problem 7

Classify vector $\boldsymbol{x}$

k training Samples $\Rightarrow \mathcal{N} = \{\boldsymbol{x}^{(s_1)}, \boldsymbol{x}^{(s_2)}, ..., \boldsymbol{x}^{(s_k)}\}$

$K(\boldsymbol{x}, \boldsymbol{y}) = \phi^T(\boldsymbol{x})\phi(\boldsymbol{x})......(1)$

For the transformed space , the distance is given by $||\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}^{(s_i)}||_2$
Squaring this distance, as it will not affect result
$||\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}^{(s_i)}||_2^2 = \phi^T(\boldsymbol{x})\phi(\boldsymbol{x}) + \phi^T(\boldsymbol{x}^{(s_i)})\phi(\boldsymbol{x}^{(s_i)}) - 2\phi^T(\boldsymbol{x})\phi(\boldsymbol{x}^{(s_i)}$
The first term is same for all data points, thus it is a constant.The distance is given by
$D(\phi(\boldsymbol{x}), \phi(\boldsymbol{x}^{(s_i)}) = \phi^T(\boldsymbol{x}^{(s_i)})\phi(\boldsymbol{x}^{(s_i)}) - 2\phi^T(\boldsymbol{x})\phi(\boldsymbol{x}^{(s_i)})$.....(2)
Using (1) and (2) , we have,
$\boxed{D(\phi(\boldsymbol{x}), \phi(\boldsymbol{x}^{(s_i)}) = K(\boldsymbol{x}^{(s_i)}, \boldsymbol{x}^{(s_i)}) - 2K(\boldsymbol{x}, \boldsymbol{x}^{(s_i)})}$