Assignment – 2

Foundations of Machine Learning

Roll Number: SM21MTECH14003

Questions: Theory

1) Support Vector Machines: In the derivation for the Support Vector Machine, we assumed that the margin boundaries are given by $w.x+b = +1$ and $w.x+b = -1$. Show that, if the +1 and -1 on the right-hand side were replaced by some arbitrary constants $+\gamma$ and $-\gamma$ where $\gamma > 0$, the solution for the maximum margin hyperplane is unchanged. (You can show this for the hard-margin SVM without any slack variables.)

Answer:

1) We have,
margin boundaries are given by,

$$w \cdot x + b = \pm 1 \quad — (1)$$

Now if we choose any number except '1' here let's say γ, then we get,

$$w \cdot x + b = \pm \gamma \quad (\gamma > 0)$$

But we can return to our original form of equation (1) by dividing by γ

i.e $$\frac{1}{\gamma} w \cdot x + \frac{1}{\gamma} b = \pm 1 \quad — (2)$$

comparing (1) with (2),
we get,

$$\hat{w} = \frac{1}{\delta} w \quad \text{and} \quad \hat{b} = \frac{1}{\delta} b$$

$\therefore$ eqn (1) gets defined as:

$$\hat{w} \cdot x + \hat{b} = \pm 1$$

Now since our goal is to minimize $||w||$ (in order to maximise the size of margin, $\frac{2}{||w||}$) $\therefore$ It really doesn't matter if we scale w by some constant γ and some can use $\hat{w}$ in place of w as the choice of γ is irrelevant except it should be positive real number. since the choice of γ is irrelevant we generally consider it as 1 just for simplification.

2) Consider the half-margin of maximum-margin SVM defined by ρ, i.e. $\rho = \frac{1}{||\mathbf{w}||}$. Show that ρ is given by:

$$\frac{1}{\rho^2} = \sum_{i=1}^{N} \alpha_i$$

Where αi are the Lagrange multipliers given by the SVM dual.

Answer:

2)

From slide 30 of SVM lectures PPT:

SVM dual: $\max_{\vec{\alpha} > 0} \min_{\vec{w}, b} \frac{1}{2} ||w|| - \sum_j \alpha_j \left[(\vec{w} \cdot \bar{x}_j + b) \cdot y_j - 1 \right]$

After solving for optimal w, b as a function of α, we get,

$w = \sum_j \alpha_j y_j X_j \quad \ldots (1)$ and $\sum \alpha_j y_j = 0 \quad \ldots (2)$

Now, we have

$\vec{w} x_i + b = y_i$

∴ Substituting w from (1), we get

$\sum_j \alpha_j y_j x_j x_i + b = y_i$

$\therefore b = y_i - \sum_j \alpha_j y_j x_j x_i$

Now,

Multiply by $\alpha_i y_i$ and take summation,

$$\therefore \sum_i \alpha_i y_i b = \sum_i \alpha_i y_i^2 - \sum_{i,j} \alpha_i \alpha_j y_i y_j X_i X_j$$

$\because y_i^2 = 1$ $\therefore$ substituting values from ① and ②, we get.

$$0 * (b) = \sum_i \alpha_i (X_i) - \|\omega\|^2$$

$$\therefore 0 = \sum_i \alpha_i - \|\omega\|^2$$

Now given that, half margin, $\rho = \frac{1}{\|\omega\|}$

$$\therefore \|\omega\|^2 = \sum_{i=1}^{N} \alpha_i = \frac{1}{\rho^2}$$

$$\boxed{\therefore \frac{1}{\rho^2} = \sum_{i=1}^{N} \alpha_i}$$

3) Let k1 and k2 be valid kernel functions. Comment about the validity of the following kernel functions, and justify your answer with proof or counter-examples as required:

(a) k(x, z) = k1(x, z) + k2(x, z)

(b) k(x, z) = k1(x, z)k2(x, z)

(c) k(x, z) = h(k1(x, z)) where h is a polynomial function with positive co-efficients

(d) k(x, z) = exp(k1(x, z))

(e) k(x, z) = exp((-||x-z||2)/σ2)

Answer:

a) It is a valid kernel. Proof:

a) $k(x,z) = k_1(x,z) + k_2(x,z)$

Let the feature maps for k_1 and k_2 be –

$k_1 : \phi^1(x) = (\phi_1^1(x), \ldots\ldots, \phi_{N_1}^1(x))$

$k_2 : \phi^2(x) = (\phi_1^2(x), \ldots\ldots, \phi_{N_2}^2(x))$

$\therefore$ Concatenating k_1 and k_2 we get,

$\phi(x) = (\phi_1^1(x), \ldots\ldots, \phi_{N_1}^1(x), \phi_1^2(x), \ldots\ldots, \phi_{N_2}^2(x))$

$\therefore$ This mapping satisfies:

$\phi(x) \cdot \phi(y) = \phi^1(x) \cdot \phi^1(y) + \phi^2(x) \cdot \phi^2(y)$

$\therefore K : \phi(x) \Rightarrow K(x,z) = k_1(x,z) + k_2(x,z)$

b) It is a valid kernel. Proof:

b) $K(x,z) = K_1(x,z) \cdot K_2(x,z)$

$K_1(x,z) : \sum_i \phi_i^1(x) \cdot \phi_i^1(z)$

$K_2(x,z) : \sum_j \phi_j^2(x) \phi_j^2(z)$

$\therefore K_1(x,z) \cdot K_2(x,z) = \left(\sum_i \phi_i^1(x) \phi_i^1(z)\right) \cdot \left(\sum_j \phi_j^2(x) \cdot \phi_j^2(z)\right)$

$= \sum_{i,j} \phi_i^1(x) \phi_i^1(z) \phi_j^2(z) \phi_j^2(x)$

Since, each ϕ outputs a scalar, let
$\phi_k(y) = \phi_i^1(y) \phi_j^2(y)$

$\therefore$ we can say, $K(x,z) = K_1(x,z) \cdot K_2(x,z)$

c) It is a valid kernel. Proof:

c) $K(x,z) = h(K_1(x,z))$ where h is the polynomial function with positive co-efficients.

here as given, h is a polynomial function with positive co-efficients,

Now we know a polynomial function is nothing but sum of products of several terms.
i.e product of two or more terms and then ~~so~~ sum of several such terms.

$\therefore$ Every term in this function is either sum or product of two or more kernels and from results from a) and b) we can say that resulting kernel obtained by $h(K_1(x,z))$ is a valid kernel.

i.e $K(x,z) = h(K_1(x,z))$ is a valid kernel ... h is +ve polynomial function with positive coefficients.

d) It is a valid kernel. Proof:

d) $K(x,z) = \exp(K_1(x,z))$

we know that,

$$\exp(x) = \lim_{n \to \infty} 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

$\therefore$ We can express $K(x,z)$ as:

$$K(x,z) = \lim_{n \to \infty} K_n(x,z)$$

$\therefore$ from prev. result (c) we can consider exp() as function of 'h' and so we can say that,

$K(x,z) = \exp(K_1(x,z))$ is a valid kernel function.

e) It is a valid kernel. Proof:

e) $K(x,z) = \exp\left(\frac{-\|x-z\|_2^2}{\sigma^2}\right)$

we ~~know~~ can write above eqn as

$$K(x,z) = \exp\left(\frac{-\|x-z\|_2^2}{\sigma^2}\right)$$

$$= \exp\left(\frac{-\|x\|^2 - \|z\|^2 + 2x^Tz}{\sigma^2}\right)$$

$$= \exp\left(\frac{-\|x\|^2}{\sigma^2}\right) \exp\left(\frac{-\|z\|^2}{\sigma^2}\right) \exp\left(2\frac{x^Tz}{\sigma^2}\right)$$

$$= h_1(x) h_2(z) \exp(K_1(x,z))$$

from c) ~~and~~ we can say $h_1(x)$ and $h_2(z)$ are valid kernels

from d) we can say $\exp(K_1(x,z))$ is a valid kernel

from b) we can say product of kernels is also a kernel.

$\therefore$ we can say,

$$K(x,z) = \exp\left(\frac{-\|x-z\|_2^2}{\sigma^2}\right)$$

is a valid kernel.

In fact, it is called as Gaussian kernel.

Questions: Programming

4) SVMs: In this question, you will be working on a soft-margin SVM. We will apply soft-margin SVM to handwritten digits from the processed US Postal Service Zip Code data set. The data (extracted features of intensity and symmetry) for training and testing are available at:

- http://www.amlbook.com/data/zip/features.train
- http://www.amlbook.com/data/zip/features.test

In this dataset, the 1st column is digit label and 2nd and 3rd columns are the features. We will train a one-versus-one (one digit is class +1 and another digit is class -1) classifier for the digits '1' (+1) and '5' (-1). (In the original dataset, only consider data samples(rows) with the label as either 1 or 5, for both train and test settings. Then for training details, you may find this link at http://scikit-learn.org/stable/modules/svm.html helpful.)

(a) Consider the linear kernel $K(x_n, x_m) = x^T_n x_m$. Train using the provided training data and test using the provided test data, and report your accuracy over the entire test set, and the number of support vectors.

Answer:

Accuracy over entire test set: **97.877**%
Total support vectors : **28**

(b) In continuation, train only using the first {50, 100, 200, 800} points with the linear kernel. Report the accuracy over the entire test set, and the number of support vectors in each of these cases.

Answer:

Number of points	Accuracy	Number of support vectors
50	98.11%	4
100	98.11%	4
200	98.34%	4
800	97.87%	16

(c) Consider the polynomial kernel

$K(x_n, x_m) = (1 + x^T_n x_m)^Q$, where Q is the degree of the polynomial. Comparing Q = 2 with Q = 5, comment whether each of the following statements is TRUE or FALSE.

i) When C = 0.0001, training error is higher at Q = 5.

Answer:

Training error at Q=2: 0.3267141803883019
Training error at Q=5: 0.01921643319406885

Training error is higher at Q=2. Therefore, statement (i) is **false**.

ii) When C = 0.001, the number of support vectors is lower at Q = 5.

Answer:

Number of Support Vectors For Q=2: [228 228] = 456
Number of Support Vectors For Q=5: [36 36] = 72

Number of support vectors is lower at Q = 5.
Therefore, statement (ii) is **true**.

iii) When C = 0.01, training error is higher at Q = 5.

Answer:

Training error at Q=2 : 0.006404112239452765
Training error at Q=5 : 0.004485131481936522

As we can see training error is slightly higher at Q = 5.
Therefore, statement (iii) is **false.**

iv) When C = 1, test error is lower at Q = 5.

Answer:
Test error at Q=2 : 0.021226415094339646
Test error at Q=5 : 0.02358490566037741

As we can see error at Q = 5 is slightly higher than error at Q = 2. Therefore, above statement is **false.**

(d) Consider the radial basis function (RBF) kernel $K(x_n, x_m) = e(-||x_n - x_m||^2)$ in the soft-margin SVM approach. Which value of $C \in \{0.01,1,100,104,106\}$ results in the lowest training error? The lowest test error? Show the error values for all the C values.

Answer:

Training error is Minimum at C = 0.01
Test error is Minimum at C = 100

C value	Training error	Test error
0.01	0.005	0.02358
1	0.006	0.02122
100	0.005	0.01886
10000	0.0109	0.02358
1000000	0.0109	0.02358

5) Following is given GISETTE dataset (https://archive.ics.uci.edu/ml/datasets/Gisette) is a handwritten digit recognition problem. The problem is to separate the highly confusible digits '4' and '9'. This dataset is one of five datasets of the NIPS 2003 feature selection challenge.

(a) Standard run: Use all the 6000 training samples from the training set to train the model, and test over all test instances, using the linear kernel. Report the train error, test error, and number of support vectors.

Answer:

Training error : 0.027333333333333332

Test error: 0.024000000000000002

Total support vectors : [542 542] = 1084

(b) Kernel variations: In addition to the basic linear kernel, investigate two other standard kernels: RBF (a.k.a. Gaussian kernel; set $\gamma = 0.001$), Polynomial kernel (set degree = 2, coef0 = 1; e.g, $(1 + x^T x)^2$). Which kernel yields the lowest training error? Report the train error, test error, and number of support vectors for both these kernels.

Answer:

In this case it was observed that polynomial kernel yields lowest training error.

1) RBF Kernel:

Training error: 0.5
Testing error: 0.5
Number Of Support Vectors: [3000 3000] = 6000

2) Polynomial Kernel:

Training error: 0.023833333333333328
Testing error: 0.020000000000000018
Number Of Support Vectors: [820 937] = 1757