CS542 Machine Learning

Homework 4

Chih Wei Tung (U17550453)

6.2

(201) $w = \sum_{n=1}^{N} \alpha_n t_n \phi(x_n)$

$y(x) = sign\left(w^T \phi(x)\right)$

$= sign(\sum_{n=1}^{N} \alpha_n t_n \phi(x_n)^T \phi(x))$

$= sign(\sum_{n=1}^{N} \alpha_n t_n k(x_n, x))$

$\therefore \alpha_n \to \alpha_n + 1$

$\Rightarrow t_n(w^T \phi(x_n)) \geq 0$

$\alpha_n \geq 0$

$\Rightarrow t_n \sum_{m=1}^{N} k(x_m, x_n) \geq 0$

So, the learning algorithm depends on Gram matrix

7.3

Given a data set of two data point $x_1 \in C^+(t_1 = +1)$ and $x_2 \in C^-(t_2 = -1)$, maximum margin hyperplane is determined by solving $\arg\min_{\vec{\theta},\theta_0} \frac{1}{2}\left\|\vec{\theta}\right\|^2$

$\Rightarrow w^T x_1 + b = +1,\ w^T x_2 + b = -1$

Solve $\arg\min\left\{\frac{1}{2}\|w\|^2 + \lambda(w^T x_1 + b - 1) + n(w^T x_2 + b + 1)\right\}$

Taking the derivative

$\Rightarrow 0 = w + \lambda x_1 + n x_2$

$\Rightarrow 0 = \lambda + n$

$\Rightarrow w = \lambda(x_1 - x_2)$

For b, use (216) and (217)

$\Rightarrow 2b = -w^T(x_1 + x_2)$

Use (220)

$\Rightarrow b = \frac{-\lambda}{2}(x_1 - x_2)^T(x_1 + x_2)$

$\Rightarrow b = \frac{-\lambda}{2}({x_1}^T x_1 - {x_2}^T x_2)$

7.4

From figure 4.1 and (7.4)

$$\rho = \frac{1}{\|w\|}$$

$$\Rightarrow \frac{1}{\rho^2} = \|w\|^2$$

From (7.16) and (7.7)

$$\Rightarrow L(w, b, a) = \frac{1}{2}\|w\|^2$$

Use this with (7.8) and (7.10)

$$\Rightarrow \frac{1}{2}\|w\|^2 = \sum_n^N \alpha_n - \frac{1}{2}\|w\|^2$$

Programming

MINST_data.mat file contain 4 matrixes, train_samples, train_samples_labels, test_samples and test_samples_labels. Each row in both train and test samples are digits of handwriting, and both train and test sample labels are labels of 0 to 9.

a. One versus one

The accuracy is 93.4%.

1. All the data in partitions should be cluster to an array[0:10]
2. Separate data to subset
3. Do the training of SVM
4. Get the prediction and calculate the confusion matrix and accuracy

One versus the rest

The accuracy is 93.1%

1. All the data in partitions should be cluster to an array[0:10]
2. Separate data to subset
3. Build data that are [0] to the rest, [1] to the rest, …, [9] to the rest
4. Do the training of SVM
5. Get the prediction and calculate the confusion matrix and accuracy

The confusion matrix and accuracy of on versus one and one versus the rest are shown as below.

one versus one confusion matrix:

Predicted	0	1	2	3	4	5	6	7	8	9	__all__
Actual											
0	85	0	0	0	0	0	1	0	0	0	86
1	0	122	0	0	0	0	0	0	0	0	122
2	0	0	102	1	0	0	0	1	8	1	113
3	0	1	0	105	0	1	2	2	3	1	115
4	0	0	0	0	101	0	2	0	0	5	108
5	3	0	0	4	1	77	1	0	6	0	92
6	3	0	0	0	0	0	83	0	1	0	87
7	1	1	1	1	2	0	0	88	2	3	99
8	1	1	0	0	1	0	0	0	83	0	86
9	0	1	0	0	0	0	0	0	3	88	92
__all__	93	126	103	111	105	78	89	91	106	98	1000

one versus one accuracy: 93.4 %

one versus the rest confusion matrix:

Predicted	0	1	2	3	4	5	6	7	8	9	__all__
Actual											
0	84	0	0	0	0	0	1	0	0	1	86
1	0	121	0	0	0	0	0	0	1	0	122
2	1	0	100	0	0	0	0	2	9	1	113
3	0	0	0	108	0	1	2	0	3	1	115
4	0	0	1	0	99	0	1	0	2	5	108
5	1	0	0	3	2	79	1	0	5	1	92
6	2	0	0	0	0	2	82	0	1	0	87
7	0	0	0	1	3	0	0	86	1	8	99
8	1	0	0	1	1	0	0	0	81	2	86
9	0	0	0	0	0	0	0	0	1	91	92
__all__	89	121	101	113	105	82	87	88	104	110	1000

one versus the rest accuracy: 93.1 %

b. DAGSVM

1. All the data in partitions should be cluster to an array[0:10]
2. Separate data to subset
3. Do the training of SVM
4. Prediction is based on using decision tree

The confusion matrix and accuracy of DAGSVM is shown as below.

DAGSVM confusion matrix:

Predicted	0	1	2	3	4	5	6	7	8	9	__all__
Actual											
0	85	0	0	0	0	0	1	0	0	0	86
1	0	122	0	0	0	0	0	0	0	0	122
2	0	0	101	1	0	0	0	2	8	1	113
3	0	0	0	104	0	1	3	2	4	1	115
4	0	0	0	0	101	0	2	0	0	5	108
5	2	0	0	4	1	77	1	0	7	0	92
6	3	0	0	0	0	0	83	0	1	0	87
7	1	1	0	1	1	0	0	88	2	5	99
8	1	0	0	0	1	0	0	0	84	0	86
9	0	1	0	0	0	0	0	0	3	88	92
__all__	92	124	101	110	104	78	90	92	109	100	1000

DAGSVM accuracy: 93.3 %