1 Bias Variance Trade-off

1.1

$$J(\beta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda||\beta||_2^2$$

$$\frac{\partial L}{\partial \lambda} = \frac{2}{n}\sum_{i=1}^{n}(y_i - x_i^T\beta)(-x_i^T) + 2\lambda\beta = 0$$

$$\beta = \frac{1}{n\lambda}\sum_{i=1}^{n}(y_i - x_i^T\beta)x_i^T$$

2 Kernel Construction

2.1

$$k(x,y) =< \phi(x), \phi(y) >$$

$$\int k(x,y)\phi(x)\phi(y)\,dx\,dy > 0$$

$$\int \phi(x)\phi(y)[a_1k_1(x,y) + a_2k_2(x,y)]\,dx\,dy$$

$$= a_1\int \phi(x)\phi(y)k_1(x,y)\,dx\,dy + a_2\int \phi(x)\phi(y)k_2(x,y)\,dx\,dy$$

$$= a_1\int \phi(x)\phi(x^{'})k_1(x,x^{'})\,dx\,dy + a_2\int \phi(x)\phi(x^{'})k_2(x,x^{'})\,dx\,dy$$

$$a_1\int \phi(x)\phi(x^{'})k_1(x,x^{'})\,dx\,dy >= 0$$

$$a_2\int \phi(x)\phi(x^{'})k_2(x,x^{'})\,dx\,dy \geq 0$$

$$k_3(x,x^{'}) = a_1\int \phi(x)\phi(x^{'})k_1(x,x^{'})\,dx\,dy + a_2\int \phi(x)\phi(x^{'})k_2(x,x^{'})\,dx\,dy \geq 0$$

k_3 is a valid kernel function

2.2

$$K = [f(x_1), f(x_2), \ldots, f(x_n)]^T[f(x_1), f(x_2), \ldots, f(x_n)] = X^TX$$

$For any vector C \in R^N$, $C^TKC = C^TX^TXC = (XC)^T(XC) = ||XC||_2^2 \geq 0$ k_4 is a valid kernel function

2.3

$k_1(x,x^{'}) = \phi_1(x)^T\phi_1(x^{'})$, $k_2(x,x^{'}) = \phi_2(x)^T\phi_2(x^{'})$

$$k_5 = k_1(x,x^{'})k_2(x,x^{'}) = \sum_{ij}\phi_1(x_i)\phi_2(x_j)\phi_1(x_i^{'})\phi_2(x_j^{'})$$
$$= \sum_{ij}\phi(x_{ij})\phi(x_{ij}^{'})$$
$$= \phi(x)^T\phi(x^{'})$$

k_5 is a valid kernel function

3 Kernel Regression

3.1

$$J = \min_w \sum_n (y_i - \omega^T x_i)^2 + \lambda||\omega||_2^2$$
$$\frac{\partial J}{\partial \omega} = -2\sum_n (y_n - \omega^T x_i)x_i + 2\lambda\omega = 0$$
$$w^* = (\sum_n x_i x_i^T + \lambda I_D)^{-1}X^T y$$
$$w^* = (X^T X + \lambda I_D)^{-1}X^T y$$

3.2

We can use the transform:$w^* = (\phi^T\phi + \lambda I_T)^{-1}\phi^T y$

$$(\phi^T\phi + \lambda I_T)^{-1}\phi^T = \phi^T\phi\phi^T + \lambda\phi_T = \phi^T(\phi\phi^T + \lambda I_N)$$
$$\phi^T = (\phi^T\phi + \lambda I_T)^{-1}\phi^T(\phi\phi^T + \lambda I_N)$$
$$\phi^T(\phi\phi^T + \lambda I_N)^{-1} = (\phi^T\phi + \lambda I_T)^{-1}\phi^T$$
$$\phi^T(\phi\phi^T + \lambda I_N)^{-1}y = (\phi^T\phi + \lambda I_T)^{-1}\phi^T y = \omega^*$$
$$\omega^* = \phi^T(\phi\phi^T + \lambda I_N)^{-1}y$$

3.3

$$\hat{y} = (w^*)^T\phi(x)$$
$$= [\phi^T(\phi\phi^T + \lambda I_N)^{-1}y]^T\phi(x)$$
$$= y^T[(\phi\phi^T + \lambda I_N)^T]^{-1}\phi\phi(x)$$
$$= y^T(\phi^T\phi + \lambda I_N)^{-1}\phi\phi(x)$$
$$= y^T(K + \lambda I_N)^{-1}k(x)$$
$$\hat{y} = y^T(K + \lambda I_N)^{-1}k(x)$$

3.4

for linear regression, the compute complexity is $O(D^3 + ND^2)$ for kernel regression, the compute complexity is $O(N^3)$ that is to say, if D ¿¿ N(like has infinite dimension), kernel regression will have large advantage.

4 SVM

4.1

No

4.2

$$\omega = (0, 0, 0, 1)^T$$

4.3

see next page

4.4

$$\phi(x) = [1, x_1, x_2, x_1x_2]$$
$$K(x, x) = \phi(x)\phi(x)^T = 1 + x_1^2 + x_2^2 + x_1x_2x_1x_2$$

5 SVM and slack

5.1

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 + C\sum \gamma_i$$

s.t

$$y_i[< \omega, x_i > +b] \geq 1 - \gamma_i$$
$$\gamma_i \geq 0$$

for large values of C, penalty will decrease the margin heavily, so the penalty for misclassifying points is very high, so the decision boundary will perfectly separate the data if possible.

5.2

the classifier will max the margin between most of the points, since the penalty is low, it will misclassify a few points.

5.3

C = 0, because we don't trust any specific points and C=0 max the margin

5.4

see next page

5.5

see next page

Linear and Kernel SVM

Use linear SVM in LIBSVM

4^{-6}	4^{-5}	4^{-4}	4^{-3}	4^{-2}	4^{-1}	1	4^{1}	4^{2}
55.75% 1.027	88.7% 0.957	91.4% 0.723	92.75% 0.553	93.95% 0.377	94.25% 0.345	94.6% 0.345	93.75% 0.348	94.45% 0.557

Kernel SVM in LIBSVM

(a) Polynomial kernel.

C	1	2	3
4^{-3}	55.75% 1.08	55.75% 1.012375	55.75% 1.037377
4^{-2}	90% 0.869	88.8% 0.786067	77.1% 1.062065
4^{-1}	91.05% 0.569065	91.8% 0.653	92.25% 0.79526
1	93.45% 0.421332	93.8% 0.516	92.35% 0.542126
4	94.65% 0.321	94.8% 0.345203	95% 0.422138
4^{2}	94.5% 0.296	95.8% 0.289461	96.35% 0.322438
4^{3}	94.4% 0.330	96.5% 0.296389	96.65% 0.301495
4^{4}	94.35% 0.449	96.95% 0.282	96.75% 0.290414
4^{5}	94.7% 0.638	97.05% 0.335209	96.5% 0.326066
4^{6}	94.65% 2.030596	96.8% 0.320021	95.7% 0.291362

4^{7}	94.45% 8.670022	95.95% 0.32766	96.75% 0.291561

(b) RBF kernel

	4^{-7}	4^{-6}	4^{-5}	4^{-4}	4^{-3}	4^{-2}	4^{-1}
4^{-3}	55.75% 1.063	55.75% 1.058	55.75% 1.041	55.75% 1.043	56% 1.043	87.75% 1.030	60.6% 1.164
4^{-2}	55.75% 1.084	55.75% 1.066	55.75% 1.176	63.85% 1.0762	90.6% 0.904	92% 0.693	92.55% 0.936
4^{-1}	55.75% 1.084	55.75% 1.0661	66.7% 1.061	90.65% 0.8586	91.35% 0.5688	93.95% 0.500	96.3% 0.628
1	55.75% 1.0705	67.4% 1.095	90.7% 0.837	91.35% 0.555	93.5% 0.4084	95.55% 0.380	97.55% 0.4638
4	67.5% 1.076	90.5% 0.853	91.2% 0.596	93.4% 0.4364	95.2% 0.316	96.6% 0.3013	96.65% 0.446
4^{2}	90.5% 0.836	91.1% 0.5418	93.7% 0.406	94.95% 0.3346	95.75% 0.2928	96.85% 0.284	97.45% 0.445
4^{3}	91.35% 0.561	93.5% 0.394	94.15% 0.321	95.05% 0.324	96.4% 0.328	97.05% 0.282	97.05% 0.4164
4^{4}	93.5% 0.4159	94.6% 0.3264	93.9% 0.318	94.9% 0.308	96.55% 0.293	96.1% 0.297	96.8% 0.437
4^{5}	94.45% 0.326	94.45% 0.313	95.05% 0.312	95.75% 0.327	96.25% 0.304429	96.45% 0.269	96.95% 0.422
4^{6}	93.95% 0.344	94.45% 0.3220	95.55% 0.3654	96.85% 0.3688	96.35% 0.342	96.65% 0.267	96.85% 0.425
4^{7}	94.7% 0.3372	94.9% 0.3676	97.1% 0.507	97% 0.528	96.05% 0.35626	96.25% 0.293	97.3% 0.4087

c)Predict

I will choose RBF with C $=4^{7}$, gamma = 4^{-1}, since it has almost highest accuracy in training set, and the accuracy in test data is 97.2%