Homework 4

Yuan Yuan

February 28, 2018

1 Contructing Kernels

1.(a) $K(x,z) = aK_1(x,z)$

K(x,z) is not a valid kernel. Since $K_1(x,z)$ is a valid kernel, $V^T K_1 V \geq 0$. If $a \leq 0$, $V^T K V = aV^T K_1 V \leq 0$, which dissatisfies the PSD condition.

1.(b) $K(x,z) =< x,z >^3 + (< x,z > -1)^2$

According to the lecture notes, for any integer $d \geq 2, k(x,z) = (x^T z + c)^d$. So both $< x,z >^3$ and $(< x,z > -1)^2$ are valid kernels. And since the sum of two kernels is a kernel, K(x,z) is a valid kernel.

1.(c) $K(x,z) =< x,z >^2 + \exp(-\|x\|^2)\exp(-\|z\|^2)$

$\exp(-\|x\|^2)$ can be represented by g(x), where g maps x from vector space to real space. Due to the property that g(x)g(z) is a valid kernel, $\exp(-\|x\|^2)\exp(-\|z\|^2)$ is a valid kernel. Since $< x,z >^2$ is a valid kernel and the sum of two kernels is a valid kernel, K(x,z) is a valid kernel.

2 Reproducing kernel Hilbert spaces

2 Let's define the inner product in space $\mathscr{F}$ as 3 times the dot product in Euclidean space:
$< x,y >_{\mathscr{F}} = 3 < x,y >$
First, we need to prove that $< f,f > \geq 0$:
$< f,f >= 3a^2 \geq 0$
Second, we need to prove that $< f(\cdot), k(\cdot,x) >_{\mathscr{F}} = f(x)$:
According to the definition, $< f(\cdot), k(\cdot,x) >_{\mathscr{F}} = \int_0^1 3f(y)k(y,x)dy = \int_0^1 3ay \cdot xydy = ax = f(x)$.
So, $\mathscr{F}$ is a RKHS with kernel K(x,y)=xy.

3 3 Convexity and KKT conditions

3.a $L = L(w, \epsilon, \epsilon^\star, \alpha, \alpha^\star, \beta, \beta^\star) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^n(\eta_i + \eta_i^\star) - \sum_{i=1}^n(\beta_i\eta_i + \beta_i^\star\eta_i^\star) - \sum_{i=1}^n \alpha_i(\epsilon + \eta_i - y_i + < w, x_i >) - \sum_{i=1}^n \alpha_i^\star(\epsilon + \eta_i^\star - y_i + < w, x_i >)$
and $\alpha_i, \alpha_i^\star, \beta_i, \beta_i^\star \geq 0, (i = 1, ..., n)$.

$\partial_w L = w - sum_{i=1}^n(\alpha_i + \alpha_i^\star)x_i = 0$
$\partial_{\epsilon_i} L = C - \alpha_i - \beta_i = 0$
$\partial_{\epsilon_i^\star} L = C - \alpha_i^\star - \beta_i^\star = 0$
From the last two equations we have that:
$0 \leq \beta_i = C - \alpha_i$
$0 \leq \beta_i^\star = C - \alpha_i^\star$
Plugging in those into Lagrangian, we get:
$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^n(\eta_i + \eta_i^\star) - \sum_{i=1}^n(\beta_i\eta_i + \beta_i^\star\eta_i^\star) - \sum_{i=1}^n \alpha_i(\epsilon + \eta_i - y_i + < w, x_i >) - \sum_{i=1}^n \alpha_i^\star(\epsilon + \eta_i^\star - y_i + < w, x_i >)$

$= \frac{1}{2}\|\sum_{i=1}^n(\alpha_i - \alpha_i^\star)x_i\|^2 + \sum_{i=1}^n \epsilon_i(C - \beta_i - \alpha_i) + \sum_{i=1}^n \epsilon_i^\star(C - \beta_i^\star - \alpha_i^\star) - \epsilon\sum_{i=1}^n(\alpha_i + \alpha_i^\star) + \sum_{i=1}^n y_i(\alpha_i - \alpha_i^\star) + \sum_{i=1}^n(\alpha_i^\star - \alpha_i) < w, x_i >$

$= -\frac{1}{2}\sum_{i,j=1}^n(\alpha_i - \alpha_i^\star)(\alpha_i - \alpha_i^\star) < x_i, x_j > - \epsilon\sum_{i=1}^n(\alpha_i + \alpha_i^\star) + \sum_{i=1}^n y_i(\alpha_i - \alpha_i^\star)$

The dual problem is $\max_{\alpha,\alpha^\star} -\frac{1}{2}\sum_{i,j=1}^n(\alpha_i - \alpha_i^\star)(\alpha_i - \alpha_i^\star) < x_i, x_j > - \epsilon\sum_{i=1}^n(\alpha_i + \alpha_i^\star) + \sum_{i=1}^n y_i(\alpha_i - \alpha_i^\star)$
$s.t. \alpha_i - \alpha_i^\star \in [0, C]$

3.b The slackness conditions of KKT are:
$\alpha_i(\epsilon + \eta_i - y_i + < w, x_i >) = 0$
$\alpha_i^\star(\epsilon + \eta_i^\star - y_i - < w, x_i >) = 0$
$\beta_i\eta_i = 0$
$\beta_i^\star\eta_i^\star = 0$
for all i=1,...,n
The first equation implies that if $\alpha_i > 0$, $(\epsilon + \eta_i - y_i + < w, x_i >) = 0$
So if $\eta_i = 0$, x_i is on the border of the region, which means it's a margin support vector. If $\eta_i > 0$, x_i is outside the region, so it's a non-margin support vector. Similarly, if $\eta_i^\star = 0$, x_i is a margin support vector; and if $\eta_i^\star > 0$, x_i is a non-margin support vector.

3.c Since ϵ defines the region inside which errors are ignored. So small ϵ leads to overfitting. That is to say, increasing ϵ make the model less likely to overfit in general.

3.d C measures how strongly we penalize errors. We want to minimize $C\sum_{i=1}^n(\eta_i + \eta_i^\star)$, and large C means small $\sum_{i=1}^n(\eta_i + \eta_i^\star)$. η and $\eta^\star$ account for errors in points that lie outside the region. So it will have low tolerance to the noise and that leads to overfitting. So, increasing C make the model more likely to overfit in general.

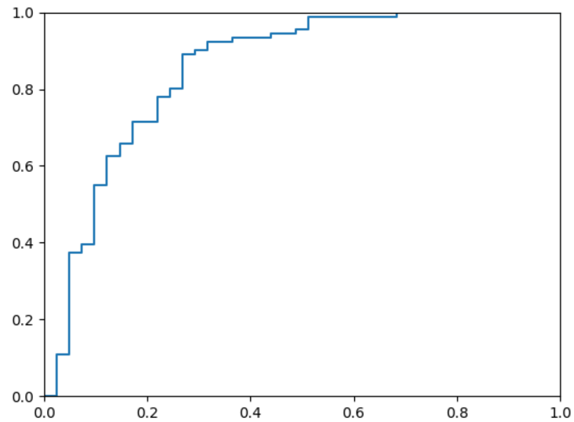
3.e $f(x) =< w, x >= \sum_{i=1}^{n} (\alpha_i - \alpha_i^\star) < x_i, x >$

4 SVM Implementation

(a)

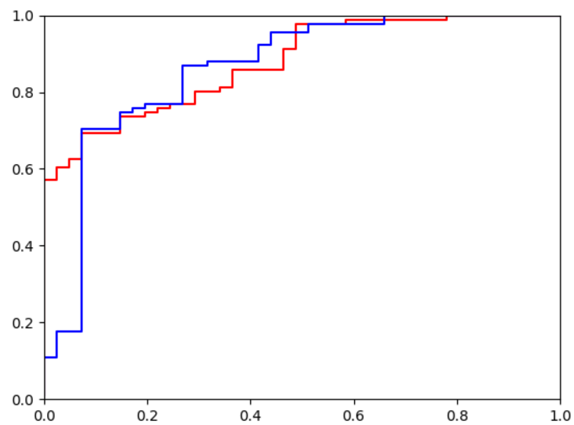
```
def test(x, w, b):
    return np.sign(np.dot(x, w)+b)

def train(x, y):
    n_samples, n_features = x.shape

    # Gram matrix
    K = np.zeros((n_samples, n_samples))
    for i in range(n_samples):
        for j in range(n_samples):
            K[i,j] = np.dot(x[i], x[j])

    P = matrix(np.outer(y,y) * np.inner(x,x))
    q = matrix(-np.ones((n_samples, 1)))
    G = matrix(np.eye(n_samples) * -1)
    h = matrix(np.zeros(n_samples))
    A = matrix(y.reshape(1, -1))
    b = matrix(np.zeros(1))
    solvers.options['show_progress'] = False
    sol = solvers.qp(P, q, G, h, A, b)
    a = np.ravel(sol['x'])

    # Support vectors have non zero lagrange multipliers
    sv = a > 1e-10
    ind = np.arange(len(a))[sv]
    a = a[sv]
    sv_x = x[sv]
    sv_y = y[sv]

    # Weight vector
    w = np.zeros(n_features)
    for n in range(len(a)):
        w += a[n] * sv_y[n] * sv_x[n]

    cond = sv_y == 1
    b_ = sv_y[cond]-np.dot(sv_x[cond],w)
    if b_.size==0:
        return "false"
    b=b_[0]

    return (w, b)
```

(b)



accuracy = 0.7954545454545454
auc:0.8520503886357546

(c)



Red curve is when $\sigma^2 = 1/5$; blue curve is when $\sigma^2 = 1/25$.

$\sigma^2 = 1/5$:
accuracy = 0.8106060606060606
auc:0.8775127311712678
$\sigma^2 = 1/25$:
accuracy = 0.7878787878787878
auc:0.860895202358617