1. **Kernel Ridge Regression** (20 points)

This problem examines the similarities and differences between KRR with and without offset.

(a) (3 points) If we just center the training data in the original input space (by subtracting off the mean of the feature vectors and responses) and then apply KRR without offset, is that equivalent to KRR with offset? Explain.

1) a

a) with offset

$$\tilde{K}_W = \langle \Phi(x_i), \Phi(x_j) \rangle - \frac{1}{n}\sum_{r=1}^{n} \langle \Phi(x_i), \Phi(x_r) \rangle - \frac{1}{n}\sum_{s=1}^{n} \langle \Phi(x_s), \Phi(x_j) \rangle + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} \langle \Phi(x_r), \Phi(x_s) \rangle$$

$$k_W(x) = \langle \Phi(x_i), \Phi(x) \rangle - \frac{1}{n}\sum_{r} \langle \Phi(x_i), \Phi(x_r) \rangle - \frac{1}{n}\sum_{s} \langle \Phi(x_s), \Phi(x) \rangle + \frac{1}{n^2}\sum_{r}\sum_{s} \langle \Phi(x_r), \Phi(x_s) \rangle$$

without offset

$$\tilde{K}_{wo} = \langle \Phi(x_i - \bar{x}), \Phi(x_j - \bar{x}) \rangle \neq \tilde{K}_W$$

$$k_{wo}(x) = \langle \Phi(x_i - \bar{x}), \Phi(x - \bar{x}) \rangle \neq k_{wo}(x)$$

(b) (3 points) In KRR with offset, give a formula for the offset b using the kernel.

1b)

$$b = \bar{y} - \hat{w}^T \bar{x} = \bar{y} - \left\{ \frac{1}{\mu} \left[X^T - X^T (\tilde{K} + \mu I)^{-1} \tilde{K} \right] \tilde{y} \right\}^T \bar{x}$$

$$= \bar{y} - \left\{ \frac{1}{\mu} X^T \left[I - (\tilde{K} + \mu I)^{-1} \tilde{K} \right] \tilde{y} \right\}^T \bar{x}$$

$$= \bar{y} - \frac{1}{\mu} \tilde{y}^T \left[I - (\tilde{K} + \mu I)^{-1} \tilde{K} \right]^T X \bar{x}$$

In above term:

$$X \bar{x} = \begin{bmatrix} \langle \tilde{x}_1, \bar{x} \rangle \\ \vdots \\ \langle \tilde{x}_n, \bar{x} \rangle \end{bmatrix} = \begin{bmatrix} \langle x_1 - \bar{x}, \bar{x} \rangle \\ \vdots \\ \langle x_n - \bar{x}, \bar{x} \rangle \end{bmatrix}$$

The entries of this vector are:

$$\langle x_i - \bar{x}, \bar{x} \rangle = \frac{1}{n} \sum_r \langle x_i, x_r \rangle - \frac{1}{n^2} \sum_r \sum_s \langle x_r, x_s \rangle$$

For term $\left[I - (\tilde{K} + \mu I)^{-1} \tilde{K} \right]^T$:

$$\left[I - (\tilde{K} + \mu I)^{-1} \tilde{K} \right]^T = I - \tilde{K} (\tilde{K} + \mu I)^{-1} = \left[\tilde{K} + \mu I - \tilde{K} \right] (\tilde{K} + \mu I)^{-1}$$

$$= \mu (\tilde{K} + \mu I)^{-1} \longrightarrow b = \bar{y} - \tilde{y}^T (\tilde{K} + n\lambda I)^{-1} \left\{ \frac{1}{n} \sum_r \langle x_i, x_r \rangle - \frac{1}{n^2} \sum_r \sum_s \right.$$

To kernelize, substitute $\langle \Phi(x), \Phi(x') \rangle$ for $\langle x, x' \rangle$.

$$b = \bar{y} - \tilde{y}^T (\tilde{K} + n\lambda I)^{-1} \left\{ \frac{1}{n} \sum_r \langle \Phi(x_i), \Phi(x_r) \rangle - \frac{1}{n^2} \sum_r \sum_s \langle \Phi(x_r), \Phi(x_s) \rangle \right\}_{i}^{n}$$

(c) (5 points) This problem shows you how to compute the KRR with offset solution without a bunch of loops. The train-train kernel matrix is the $n \times n$ matrix $K = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]$. The "centered" train-train kernel matrix $\tilde{K}$ is the $n \times n$ matrix whose entries are $\langle \tilde{\Phi}(\boldsymbol{x}_i), \tilde{\Phi}(\boldsymbol{x}_j) \rangle$, where $\tilde{\Phi}(\boldsymbol{x}) := \Phi(\boldsymbol{x}) - \frac{1}{n} \sum_{\ell=1}^{n} \Phi(\boldsymbol{x}_\ell)$ and Φ is a feature map corresponding to the kernel k. This is the matrix that arises in KRR with offset. Calculation of $\tilde{K}$ is facilitated by the formula

$$\tilde{K} = K - KO - OK + OKO, \tag{1}$$

where O is a square matrix with all entries equal to $1/n$. You should verify this fact but you don't need to turn your work in.

Now consider a test data set $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_m$, and let K' be the $n \times m$ train-test matrix with entries $k(\boldsymbol{x}_i, \boldsymbol{x}'_j)$, and let $\tilde{K}'$ be the $n \times m$ centered train-test matrix, whose entries are $\langle \tilde{\Phi}(\boldsymbol{x}_i), \tilde{\Phi}(\boldsymbol{x}'_j) \rangle$. Determine a formula analogous to (1) for relating $\tilde{K}'$ to K'. Also, determine a formula for computing the predicted outputs on all the test points. Your formula should yield a column vector of length m.

1(c)
page 1

Train-test

$$\tilde{K}' = \left[\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x_j') \rangle\right]_{i,j=1}^{n,m} = \left[\langle \Phi(x_i) - \frac{1}{n}\sum_l \Phi(x_l), \Phi(x_j') - \frac{1}{n}\sum_l \Phi(x_l) \rangle\right]_{i,j}^{n,m}$$

$$= \left[\langle \Phi(x_i), \Phi(x_j') \rangle\right]_{i,j=1}^{n,m} - \frac{1}{n}\left[\langle \Phi(x_i), \sum_l \Phi(x_l) \rangle\right]_{i,j=1}^{n,m} - \frac{1}{n}\left[\langle \sum_l x(l), \Phi(x_j') \rangle\right]_{i,j}^{n,m}$$

$$+ \frac{1}{n^2}\left[\langle \sum_l \Phi(x_l), \sum_l \Phi(x_l) \rangle\right]_{i,j=1}^{n,m}$$

$$\left[\langle \Phi(x_i), \Phi(x_j') \rangle\right] = K'$$

$$\frac{1}{n}\left[\langle \Phi(x_i), \sum_l (\Phi(x_l)) \rangle\right]_{i,j=1}^{n,m} = \frac{1}{n}\langle \Phi(x_i), \sum_l \Phi(x_l) \rangle \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times m}$$

$$= \frac{1}{n}\langle \Phi(x_i), \begin{bmatrix} \Phi(x_1) & \cdots & \Phi(x_n) \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \rangle \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times m}$$

$$= \left[\langle \Phi(x_i), \Phi(x_j') \rangle\right]_{i,j=1}^{n,m} \times \frac{1}{n} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times m}$$

$$= \left[\langle \Phi(x_i), \Phi(x_j') \rangle\right]_{i,j=1}^{n,m} \times \begin{bmatrix} 1/n & \cdots & 1/n \\ \vdots & & \vdots \\ 1/n & \cdots & 1/n \end{bmatrix}_{n\times m} = K\,O_{nm}$$

$$\frac{1}{n}\left[\langle \sum_l x(l), \Phi(x_j') \rangle\right]_{i,j=1}^{n,m} = \frac{1}{n} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \langle \sum_l \Phi(x_l), \Phi(x_j') \rangle$$

$$= \frac{1}{n} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \langle \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times n} \begin{bmatrix} \Phi(x_1) \\ \vdots \\ \Phi(x_n) \end{bmatrix}, \Phi(x_j') \rangle$$

$$= \frac{1}{n} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times n} \left[\langle \Phi(x_i), \Phi(x_j') \rangle\right]_{i,j=1}^{n,m} = \begin{bmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix}_{n\times n} \left[\langle \Phi(x_i), \Phi(x_j') \rangle\right]_{i,j}^{n,m}$$

$$= O\,K'$$

1(c)
page 2

$$\frac{1}{n^2}\left[\left\langle \sum_{l} \Phi(x_l), \sum_{l} \Phi(x_l)\right\rangle\right]_{i,j=1}^{n} = \frac{1}{n^2}\underbrace{\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}}_{n\times 1}\left\langle \sum_{l} \Phi(x_l), \sum_{l} \Phi(x_l)\right\rangle \underbrace{\begin{bmatrix}1 & \cdots & 1\end{bmatrix}}_{1\times m}$$

$$= \frac{1}{n^2}\underbrace{\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}}_{n\times 1}\left\langle \underbrace{\begin{bmatrix}1 & \cdots & 1\end{bmatrix}}_{1\times n}\begin{bmatrix}\Phi(x_1)\\ \vdots\\ \Phi(x_n)\end{bmatrix}, \begin{bmatrix}\Phi(x_1) & \cdots & \Phi(x_n)\end{bmatrix}\underbrace{\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}}_{n\times 1}\right\rangle \underbrace{\begin{bmatrix}1 & \cdots & 1\end{bmatrix}}_{1\times m}$$

$$= \frac{1}{n^2}\underbrace{\begin{bmatrix}1 & \cdots & 1\\ \vdots & & \\ 1 & \cdots & 1\end{bmatrix}}_{n\times n}\left[\left\langle \Phi(x_i), \Phi(x_j)\right\rangle\right]_{i,j=1}^{n,n}\underbrace{\begin{bmatrix}1 & \cdots & 1\\ \vdots & & \\ 1 & \cdots & 1\end{bmatrix}}_{n\times m}$$

$$= \begin{bmatrix}\frac{1}{n} & \cdots & \frac{1}{n}\\ \vdots & & \vdots\\ \frac{1}{n} & \cdots & \frac{1}{n}\end{bmatrix}\left[\left\langle \Phi(x_i), \Phi(x_j)\right\rangle\right]_{i,j=1}^{n,n}\underbrace{\begin{bmatrix}1/n & \cdots & 1/n\\ \vdots & & \vdots\\ 1/n & \cdots & 1/n\end{bmatrix}}_{n\times m} = OKO_{nm}$$

$$\tilde{K}' = K' - KO_{nm} - OK' + OKO_{nm}$$

Let's revisit the body fat data which we have seen earlier. We saw that a linear fit was reasonable, but now let's try a nonlinear fit.

Use the first 150 examples for training, and the remainder for estimating the mean squared error.

You will be asked to implement two variants of kernel ridge regression. For the next two problems please turn in

- the mean squared error on the training data
- the mean squared error on the test inputs
- the offset b in part **(d)** using your formula from **(b)**

(d) (3 points) Implement kernel ridge regression *with* offset. Remember to report the offset in addition to the other requested items.

MSE_training= 28.7940, MSE_test= 34.1488, b=1.0314

Code for KRR with offset:

```
close all; clear all; clc
load bodyfat_data.mat
n=150;
lambda=3e-3;
d=2;
x_train=X(1:n,:);
y_train=y(1:n);
x_test=X(n+1:end,:);
y_test=y(n+1:end);
m=numel(y_test);
xbar=sum(x_train,1)/n;
ybar=sum(y_train)/n;
xtilde=x_train-repmat(xbar,[n 1]);
ytilde=y_train-ybar;
O=1/n*ones(n);
for i=1:n
    for j=1:n
        K(i,j)=gaus_ker(x_train(i),x_train(j));
    end
end
Ktilde=K-K*O-O*K+O*K*O;
yhat_train=ybar+ytilde'*(Ktilde+n*lambda*eye(n))^-1*Ktilde;
yhat_train=yhat_train';
e_train=y_train-yhat_train;
MSE_train=sum(e_train.^2)/98
for i=1:n
    for j=1:m
        Kprime(i,j)=gaus_ker(x_train(i),x_test(j));
    end
end
Onm=1/n*ones(n,m);
Ktildeprime=Kprime-K*Onm-O*Kprime+O*K*Onm;
yhat_test=ybar+ytilde'*(Ktilde+n*lambda*eye(n))^-1*Ktildeprime;
yhat_test=yhat_test';
e_test=y_test-yhat_test;
```

```
MSE_test=sum(e_test.^2)/98
b=ybar-ybar-ytilde'*(Ktilde+n*lambda*eye(n))^-1*(1/n*sum(K,2)-
(1/n^2)*sum(K)*ones(n,1))
```

Code for calculation of Gaussian kernel:

```
function k=gaus_ker(u,v)
sigma=1.5;
k=exp(-1/(2*sigma^2)*norm(u-v)^2);
end
```

(e) (3 points) Implement kernel ridge regression *without* offset. Comment on any differences in performance with respect to part **(b)**. This problem should be easy once you have solved **(d)**

MSE_training=28.8242, MSE_test=73.9540

Code for KRR without offset:

```
close all; clear all; clc
load bodyfat_data.mat

n=150;
lambda=3e-3;
d=2;

x_train=X(1:n,:);
y_train=y(1:n);

xbar=sum(x_train,1)/n;
ybar=sum(y_train)/n;

xtilde=x_train-repmat(xbar,[n 1]);
ytilde=y_train-ybar;

%%%%%%%%% Ktilde %%%%%%%%
for i=1:n
    for j=i:n
        sum2=0;
        for r=1:n
            sum2=sum2+gaus_ker(x_train(i),x_train(r));
        end
        sum3=0;
        for s=1:n
            sum3=sum3+gaus_ker(x_train(s),x_train(j));
        end
        sum45=0;
        for r=1:n
            for s=1:n
                sum45=sum45+gaus_ker(x_train(r),x_train(s));
            end
        end
```

```
        Ktilde(i,j)=gaus_ker(x_train(i),x_train(j))-1/n*sum2-
1/n*sum3+1/n^2*sum45;
        Ktilde(j,i)=Ktilde(i,j);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_test=X(n+1:end,:);
y_test=y(n+1:end);

for i=1:numel(y_test)
    yhat_test(i)=ybar+ytilde'*(Ktilde+n*lambda*eye(n))^-1*ktilde(x_test(i));
end
e=y_test-yhat_test;
MSE=sum(e.^2)/98
```

2. Support Vector Regression (20 points)

Support vector regression (SVR) is a method for regression analogous to the support vector classifier. Let $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$ be training data for a regression problem.

In the case of *linear regression*, SVR solves

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) \\ \text{s.t.} \quad & y_i - \mathbf{w}^T\boldsymbol{x}_i - b \le \epsilon + \xi_i^+ \quad \forall i \\ & \mathbf{w}^T\boldsymbol{x}_i + b - y_i \le \epsilon + \xi_i^- \quad \forall i \\ & \xi_i^+ \ge 0 \quad \forall i \\ & \xi_i^- \ge 0 \quad \forall i \end{aligned}$$

where $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, $\xi^+ = (\xi_i^+, \ldots, \xi_n^+)^T$, and $\xi^- = (\xi_i^-, \ldots, \xi_n^-)^T$.

Here $\epsilon > 0$ is fixed.

a. (5 points) Show that for an appropriate choice of λ, SVR solves

$$\min_{\mathbf{w}, b} \ \frac{1}{n}\sum_{i=1}^{n} \ell_\epsilon(y_i, w^T\boldsymbol{x}_i + b) + \lambda\|w\|_2^2$$

where $\ell_\epsilon(y, t) = \max\{0, |y - t| - \epsilon\}$ is the so-called ϵ-*insensitive loss*, which does not penalize prediction errors below a level of ϵ.

2a (page 1)

SVR $\rightarrow$

$$\min_{w,b,\xi^+,\xi^-} \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_i (\xi_i^+ + \xi_i^-)$$

$$\text{s.t.} \quad y_i - w^Tx_i - b \le \varepsilon + \xi_i^+ \quad \forall i$$

$$w^Tx_i + b - y_i \le \varepsilon + \xi_i^- \quad \forall i$$

$$\xi_i^+ \ge 0, \quad \xi_i^- \ge 0 \quad \forall i$$

Constraints: $y_i - w^Tx_i - b \le \varepsilon + \xi_i^+ \rightarrow \xi_i^+ \ge y_i - w^Tx_i - b - \varepsilon$

$w_i^Tx_i + b - y_i \le \varepsilon + \xi_i^- \rightarrow \xi_i^- \ge w^Tx_i + b - y_i - \varepsilon$

$$\left.\begin{array}{l} \xi_i^+ \ge y_i - w^Tx_i - b - \varepsilon \\ \xi_i^+ \ge 0 \end{array}\right\} \rightarrow \xi_i^+ = \max\{0, y_i - w^Tx_i - b - \varepsilon\}$$

$$\left.\begin{array}{l} \xi_i^- \ge w^Tx_i + b - y_i - \varepsilon \\ \xi_i^- \ge 0 \end{array}\right\} \rightarrow \xi_i^- = \max\{0, w^Tx_i + b - y_i - \varepsilon\}$$

Constrained optimization problem transforms to unconstrained optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_i \max\{0, y_i - w^Tx_i - b - \varepsilon\} + \max\{0, w^Tx_i + b - y_i - \varepsilon\}$$

① $$\left.\begin{array}{l} y_i - w^Tx_i - b - \varepsilon < 0 \\ w^Tx_i + b - y_i - \varepsilon < 0 \end{array}\right\} \rightarrow |y_i - w^Tx_i - b| - \varepsilon < 0$$

For this case, $\sum(0+0) = 0$

② $$\left.\begin{array}{l} y_i - w^Tx_i - b - \varepsilon < 0 \\ w^Tx_i + b - y_i - \varepsilon > 0 \end{array}\right\} \rightarrow y_i - w^Tx_i - b < -\varepsilon \rightarrow |y_i - w^Tx_i - b| > \varepsilon$$

For this case, $\sum 0 + (w^Tx_i + b - y_i - \varepsilon) = \sum |y_i - w^Tx_i - b| - \varepsilon$

③ $$\left.\begin{array}{l} y_i - w^Tx_i - b - \varepsilon > 0 \\ w^Tx_i + b - y_i - \varepsilon < 0 \end{array}\right\} \rightarrow y_i - w^Tx_i - b > \varepsilon \rightarrow |y_i - w^Tx_i - b| > \varepsilon$$

For this case $\sum (y_i - w^Tx_i - b - \varepsilon) + 0 = \sum |y_i - w^Tx_i - b| - \varepsilon$

2a
(page 2)

④ $\left.\begin{array}{l} y_i - w^Tx_i - b - \varepsilon > 0 \\ w^Tx_i + b - y_i - \varepsilon > 0 \end{array}\right\} \rightarrow \begin{array}{l} y_i - w^Tx_i - b > \varepsilon \\ y_i - w^Tx_i - b < -\varepsilon \end{array} \xrightarrow{\varepsilon > 0}$ impossible

Based on cases ①, ..., ④:

$$\sum_i \max\{0, y_i - w^Tx_i - b - \varepsilon\} + \max\{0, w^Tx_i + b - y_i - \varepsilon\}$$
$$= \sum_i \max\{0, |y_i - w^Tx_i - b| - \varepsilon\}$$

$$\rightarrow \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_i \max\{0, y_i - w^Tx_i - b - \varepsilon\} + \max\{0, w^Tx_i + b - y_i - \varepsilon\}$$
$$= \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_i \max\{0, |y_i - w^Tx_i - b| - \varepsilon\}$$
$$= \frac{1}{2}\|w\|_2^2 + C/n \sum_i l_\varepsilon(y_i, w^Tx_i + b)$$

objective function can be divided by any constant value.
Devide objective function by C:

$$\underbrace{\frac{1}{2C}}_{\gamma}\|w\|_2^2 + \frac{1}{n}\sum_i l_\varepsilon(y_i, w^Tx_i + b)$$

b. (8 points) The optimization problem is convex with affine constraints, and therefore strong duality holds. Use the KKT conditions to derive the dual optimization problem in a manner analogous to the support vector classifier. As in the SVC, you should eliminate the dual variables corresponding to the constraints $\xi_i^+ \geq 0$, $\xi_i^- \geq 0$.

2b (page 1)

SVC

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

s.t. $y_i(w^T x_i + b) \geq 1-\xi_i \quad \forall i$

$\xi_i \geq 0 \quad \forall i$

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i - \sum \alpha_i\left[y_i(w^T x_i + b) - 1 + \xi_i\right] - \sum \beta_i \xi_i$$

$$\max_{\alpha\geq 0,\ \beta \geq 0} \ \min_{w,b,\xi} \ L(w,b,\xi,\alpha,\beta)$$

$$\frac{\partial L}{\partial w} = w - \sum \alpha_i y_i x_i = 0$$

$$\frac{\partial L}{\partial b} = -\sum \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = \frac{C}{n} - \alpha_i - \beta_i = 0$$

SVR

$$\min_{w,b,\xi^+,\xi^-} \ \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-)$$

s.t. $y_i - w^T x_i - b \leq \varepsilon + \xi_i^+ \quad \forall i$

$w^T x_i + b - y_i \leq \varepsilon + \xi_i^- \quad \forall i$

$\xi_i^+ \geq 0, \quad \xi_i^- \geq 0$

$$L(w, b, \xi^+, \xi^-, \alpha, \beta, \gamma, \lambda) = \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) + \sum \alpha_i(y_i - w^T x_i - b - \varepsilon - \xi_i^+) + \sum \beta_i(w^T x_i + b - y_i - \varepsilon - \xi_i^-) - \sum \gamma_i \xi_i^+ - \sum \lambda_i \xi_i^-$$

$$\max_{\substack{\alpha\geq 0,\ \beta\geq 0\\ \gamma\geq 0,\ \lambda\geq 0}} \ \min_{w,b,\xi^-,\xi^+} \ L(w,b,\xi^-,\xi^+,\alpha,\beta,\lambda,\gamma)$$

$$\frac{\partial L}{\partial w} = w - \sum \alpha_i x_i + \sum \beta_i x_i = 0$$

$$\frac{\partial L}{\partial b} = -\sum \alpha_i + \sum \beta_i = 0$$

$$\frac{\partial L}{\partial \xi_i^+} = \frac{C}{n} - \alpha_i - \gamma_i = 0$$

$$\frac{\partial L}{\partial \xi_i^-} = \frac{C}{n} - \beta_i - \lambda_i = 0$$

2 b
(page 2)

SVC

$$w = \sum \alpha_i y_i x_i$$

$$\sum \alpha_i y_i = 0$$

$$\frac{C}{n} = \alpha_i + \beta_i$$

$$\frac{1}{2}\left(\sum_i \alpha_i y_i x_i\right)^T\left(\sum_j \alpha_j y_j x_j\right) + \cancel{\sum(\alpha_i+\beta_i)\xi_i}$$

$$-\left(\sum \alpha_i y_i x_i\right)^T\left(\sum \alpha_i y_i x_i\right) - b\cancel{\sum \alpha_i y_i}$$

$$+\sum \alpha_i - \cancel{\sum \alpha_i \xi_i} - \cancel{\sum \beta_i \xi_i}$$

$$= -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i$$

$$\max_{\alpha,\beta} \; -\frac{1}{2}\sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum \alpha_i$$

s.t. $\sum \alpha_i y_i = 0 \quad \forall i$

$\alpha_i + \beta_i = \frac{C}{n} \quad \forall i$

$\alpha_i \geq 0, \; \beta_i \geq 0$

SVR

$$w = -\sum(\beta_i - \alpha_i)x_i$$

$$\sum \alpha_i = \sum \beta_i$$

$$\frac{C}{n} = \gamma_i + \alpha_i = \lambda_i + \beta_i$$

$$\frac{1}{2}\left(\sum(\beta_i-\alpha_i)x_i\right)^T\left(\sum(\beta_j-\alpha_j)x_j\right)$$

$$+\cancel{\frac{C}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-)}$$

$$+\sum \alpha_i y_i + \sum_j\left(\sum_i(\beta_i-\alpha_i)x_i\right)^T \alpha_j x_j$$

$$-\cancel{\sum \alpha_i b} - \cancel{\sum \alpha_i \varepsilon} - \cancel{\sum \alpha_i \xi_i^+}$$

$$-\sum_j\left(\sum_i(\beta_i-\alpha_i)x_i\right)^T \beta_j x_j + \cancel{\sum \beta_i b}$$

$$-\sum \beta_i y_i - \cancel{\sum \beta_i \varepsilon} - \cancel{\sum \beta_i \xi_i}$$

$$-\cancel{\sum \gamma_i \xi_i^+} - \cancel{\sum \lambda_i \xi_i^-}$$

$$= \frac{-1}{2}\sum_{i,j=1}^{n}(\beta_i-\alpha_i)(\beta_j-\alpha_j)x_i^T x_j - \sum(\beta_i-\alpha_i)y_i$$

$$\sum(\beta_i-\alpha_i)y_i$$

$$\max_{\alpha,\beta,\gamma,\lambda} \; \frac{-1}{2}\sum_{i,j}^{n}(\beta_i-\alpha_i)(\beta_j-\alpha_j)x_i^T x_j -$$

s.t. $\sum \alpha_i - \sum \beta_i = 0 \quad \forall i$

$\gamma_i + \alpha_i = \lambda_i + \beta_i = \frac{C}{n} \quad \forall i$

$\alpha_i \geq 0, \; \beta_i \geq 0, \; \gamma_i \geq 0 \; \lambda_i \geq 0$

2b
(page 3)

SVC

eliminate β_i

$$\max_{\alpha} \; -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum \alpha_i$$

s.t. $\sum \alpha_i y_i = 0$

$0 \le \alpha_i \le \frac{C}{n} \quad \forall i$

1st KKT: $w^* + \sum_{i=1}^{n} \alpha_i^* (-y_i x_i) = 0$

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

Complementary Slackness $\alpha_i^* \left[1 - \xi_i^* - y_i (w^{*T} x_i + b^*)\right] = 0$

$$\beta_i^* \xi_i^* = 0$$

$\alpha_i^* < \frac{C}{n} \rightarrow \beta_i^* > 0 \rightarrow \xi_i^*$

$i: \; 0 < \alpha_i^* < C/n$

$\xi_i^* = 0$

$1 - \xi_i^* - y_i (w^{*T} x_i + b^*) = 0$

$\rightarrow b^* = y_i - w^{*T} x_i$

SVR

eliminate γ_i & λ_i

$$\max_{\alpha, \beta} \; -\frac{1}{2}\sum_{i,j} (\beta_i - \alpha_i)(\beta_j - \alpha_j) x_i^T x_j - \sum_i (\beta_i - \alpha_i) y_i$$

s.t. $\sum \alpha_i - \sum \beta_i = 0 \quad \forall i$

$0 \le \alpha_i \le C/n \quad \forall i$

$0 \le \beta_i \le C/n \quad \forall i$

$$w^* + \sum_{i=1}^{n} \alpha_i^* (-x_i) + \sum_{i=1}^{n} \beta_i^* x_i = 0$$

$$w^* = \sum_{i=1}^{n} (\alpha_i^* - \beta_i^*) x_i$$

$\alpha_i^* \left[y_i - w^{*T} x_i - b^* - \varepsilon - \xi_i^{+*}\right] = 0$

$\beta_i^* \left[w^{*T} x_i + b^* - y_i - \varepsilon - \xi_i^{-*}\right] = 0$

$\gamma_i^* \xi_i^{+*} = 0 \qquad \lambda_i^* \xi_i^{-*} = 0$

$\alpha_i^* < C/n \rightarrow \gamma_i^* > 0 \rightarrow \xi_i^{+*} = 0$

$\beta_i^* < C/n \rightarrow \lambda_i^* > 0 \rightarrow \xi_i^{-*} = 0$

$i: \; 0 < \alpha_i^* < C/n$

$\xi_i^{+} = 0$

$y_i - w^{*T} x_i - b^* - \varepsilon - \xi_i^{+*} = 0$

$\rightarrow b^* = y_i - w^{*T} x_i - \varepsilon$

$i: \; 0 < \beta_i^* < C/n$

$\xi_i^{-} = 0$

$w^{*T} x_i + b^* - y_i - \varepsilon - \xi_i^{-*} = 0$

$\rightarrow b^* = y_i - w^T x_i + \varepsilon$

c. (4 points) Explain how to kernelize SVR. Be sure to explain how to determine b^*.

2(c)

SVC

$$f(x) = \text{sign}\{w^{*T}x + b^*\}$$

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

$$b^* = y_i - w^{*T}x_i$$

$$i : (0 < \alpha_i^* < \tfrac{C}{n})$$

$$f(x) = \text{sign}\left\{\sum_{i=1}^{n} \alpha_i^* y_i k(x, x_i) + b^*\right\}$$

where

$$b^* = y_i - \sum_{j=1}^{n} \alpha_j^* y_j k(x_j, x_i)$$

$$i : (0 < \alpha_i^* < \tfrac{C}{n})$$

SVR

$$f(x) = w^{*T}x + b^*$$

$$w^* = \sum_{i=1}^{n} (\alpha_i^* - \beta_i^*) x_i$$

$$b^* = y_i - w^{*T}x_i - \varepsilon \quad (0 < \alpha_i^* < \tfrac{C}{n})$$

$$= y_i - w^{*T}x_i + \varepsilon \quad (0 < \beta_i^* < \tfrac{C}{n})$$

$$f(x) = \sum_{i=1}^{n} (\alpha_i^* - \beta_i^*) k(x, x_i) + b^*$$

where

$$b^* = y_i - \sum_{j=1}^{n} (\alpha_j^* - \beta_j^*) k(x_j, x_i) -$$

$$i : (0 < \alpha_i^* < \tfrac{C}{n})$$

$$b^* = y_i - \sum_{j=1}^{n} (\alpha_j^* - \beta_j^*) k(x_j, x_i)$$

$$i : (0 < \beta_i^* < \tfrac{C}{n})$$

d. (3 points) Argue that the final predictor will only depend on a subset of training examples, and characterize those training examples.

2(d)

Final classifier only depends on samples for which $0 < \alpha_i^* < \frac{C}{n}$

Final predictor depends only on samples for which

$0 < \alpha_i^* < \frac{C}{n}$

$0 < \beta_i^* < \frac{C}{n}$