1 Derivation

To derive the least squares support vector regression (J.A. Suykens LSSVM Book) model we start with the linear expression:

$$y = W^{\mathsf{T}}\mathbf{x} + b \tag{1.1}$$

$$\min_{W,e} J(W,e) = \frac{1}{2}W^{\mathsf{T}}W + C\frac{1}{2}\sum_{k=1}^{n} e_k \tag{1.2}$$

Such that

$$y_k = W^{\mathsf{T}}\mathbf{x}_k + b + e_k \qquad k = 1,\ldots,n \tag{1.3}$$

$$L(W,b,e;\alpha) = \frac{1}{2}W^{\mathsf{T}}W + C\frac{1}{2}\sum_{k=1}^{n} e_k + \sum_{k=1}^{n} \alpha_k \left(W^{\mathsf{T}}\mathbf{x}_k + b + e_k - y_k\right) \tag{1.4}$$

Derive the KKT system

$$\frac{\partial L}{\partial W} = 0 \rightarrow W = \sum_{k=1}^{n} \alpha_k x_k \tag{1.5}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{n} \alpha_k = 0 \tag{1.6}$$

$$\frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = Ce_k \qquad k = 1,\ldots,n \tag{1.7}$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \rightarrow W^{\mathsf{T}}\mathbf{x}_k + b + e_k - y_k = 0 \qquad k = 1,\ldots,n \tag{1.8}$$

$$\tag{1.9}$$

After eliminating W and e, with $\mathbf{1}_n = \langle 1,\ldots,1\rangle$, $\mathbf{y} = [y_1,\ldots,y_n]$, and $\alpha = [\alpha_1,\ldots,\alpha_n]$ the solution is as follows in block matrix notation

$$\begin{bmatrix} 0 & \mathbf{1}_n^{\mathsf{T}} \\ \mathbf{1}_n & \Omega + \frac{I}{C} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \tag{1.10}$$

2 Psuedocode

2.1 Training

The solution in eq. (1.10) is rewritten in psuedocode in algorithm 1

Algorithm 1 LSSVR Training

```
Input: $\mathbf{X}, \mathbf{y}, \gamma$
Output: $\alpha, b$
$\Omega \leftarrow [[]]$    ▷ Construct matrix of inner products in high dimensional space
for $k = 0 \rightarrow n$ do
    for $l = 0 \rightarrow n$ do
        $\Omega_{k,l} \leftarrow K(\mathbf{X}_k, \mathbf{X}_l)$
    end for
end for
$\mathbf{H} \leftarrow \Omega + \frac{\mathbf{I}}{\gamma}$
$\mathbf{A} \leftarrow [[]]$    ▷ Construct left hand side matrix
$\mathbf{A}_{0,0} \leftarrow 0$
for $k = 0 \rightarrow n$ do
    $\mathbf{A}_{k+1,0} \leftarrow 1$
    $\mathbf{A}_{0,k+1} \leftarrow 1$
end for
for $k = 0 \rightarrow n$ do
    for $l = 0 \rightarrow n$ do
        $\mathbf{A}_{k+1,l+1} \leftarrow \mathbf{H}_{k,l}$
    end for
end for
$\mathbf{B} \leftarrow []$    ▷ Construct left hand side of the equation
$\mathbf{B}_0 \leftarrow 0$
for $k = 0 \rightarrow n$ do
    $\mathbf{B}_{k+1} \leftarrow \mathbf{y}_k$
end for
$\mathbf{A}^\dagger \leftarrow pseudoInverse(\mathbf{A})$    ▷ Compute solution using pseudo inverse
$\mathbf{S} \leftarrow \mathbf{A}^\dagger \mathbf{B}$
$b \leftarrow \mathbf{S}_0$
for $k = 0 \rightarrow n$ do
    $\alpha_k \leftarrow \mathbf{S}_{k+1}$
end for
```

The basic psuedocode from the LSSVM Equation for function regression is defined in algorithm 1 for a training set of length n, samples $\mathbf{X}$, and sample labels $\mathbf{y}$. Training the LSSVM means computing the values of langrange multipliers α and bias b. K is the kernel function used to compute the inner products in high dimensional space, here we asume the RBF kernel. $\mathbf{A}$ is a matrix of size $n+1$ by $n+1$. $\mathbf{H}$ is a matrix of size n by n. $\mathbf{I}$ is the identity. $\mathbf{B}$ is a vector of size $n+1$. $\mathbf{S}$ is a vector of size $n+1$.

The use of the Moore Penrose inverse or commonly known as psuedoinverse is such that cases where the inverse is not computable, the psuedoinverse is still guaranteed to exist. Note that matrix A is square, so the use of psuedoinverse is for cases such as singular matrices.

2.2 Prediction

After training the model can be used for prediction of unseen samples. The psuedocode for prediction is shown in algorithm 2 for prediction samples $\mathbf{U}$. The trained model uses the learned multipliers of training samples α, the training samples $\mathbf{X}$, and the bias b.

Algorithm 2 LSSVR Prediction

```
Input: $\mathbf{U}, \alpha, \mathbf{X}, b$
Output: $\mathbf{v}$
$\Omega \leftarrow [[]]$        ▷ Construct matrix of inner products in high dimensional space
for $k = 0 \rightarrow m$ do
    for $l = 0 \rightarrow n$ do
        $\Omega_{k,l} \leftarrow K(\mathbf{U}_k, \mathbf{X}_l)$
    end for
end for
$\mathbf{v} \leftarrow \Omega\alpha + \mathbf{1}_m b$
```

3 Example Calculation

3.1 Training

In this exmple we will be using the function $2x^2+4$. The values of this function can be seen in Table 3.1.

No	x	y
1	0.0	4.0
2	0.33...	4.22...
3	0.66...	4.88...
4	1.0	6.0

Table 3.1: Example data of function $2x^2+4$

For $\Omega_{i,j} = K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
For example, with $\sigma = 1$, $x_i = 0.0$, & $x_j = 0.33\ldots$

$$\begin{aligned} K(0.0, 0.33) &= exp\left(-\frac{\|0.0 - 0.33\|^2}{2(1)^2}\right) \\ &= exp\left(-\frac{0.33^2}{2}\right) \\ &= 0.9460 \end{aligned} \tag{3.1}$$

$$\Omega \leftarrow \begin{bmatrix} 1.0000 & 0.9460 & 0.8007 & 0.6065 \\ 0.9460 & 1.0000 & 0.9460 & 0.8007 \\ 0.8007 & 0.9460 & 1.0000 & 0.9460 \\ 0.6065 & 0.8007 & 0.9460 & 1.0000 \end{bmatrix} \tag{3.2}$$

$$\mathbf{I}\frac{1}{\gamma} \rightarrow \mathbf{I}\frac{1}{5} \rightarrow \begin{bmatrix} 0.2000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.2000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.2000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.2000 \end{bmatrix} \tag{3.3}$$

$$\Omega + \mathbf{I}\frac{1}{5} \rightarrow H \rightarrow \begin{bmatrix} 1.2000 & 0.9460 & 0.8007 & 0.6065 \\ 0.9460 & 1.2000 & 0.9460 & 0.8007 \\ 0.8007 & 0.9460 & 1.2000 & 0.9460 \\ 0.6065 & 0.8007 & 0.9460 & 1.2000 \end{bmatrix} \tag{3.4}$$

$$A \rightarrow \begin{bmatrix} 0.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.2000 & 0.9460 & 0.8007 & 0.6065 \\ 1.0000 & 0.9460 & 1.2000 & 0.9460 & 0.8007 \\ 1.0000 & 0.8007 & 0.9460 & 1.2000 & 0.9460 \\ 1.0000 & 0.6065 & 0.8007 & 0.9460 & 1.2000 \end{bmatrix} \tag{3.5}$$

$$B \rightarrow \begin{bmatrix} 0.0000 \\ 4.0000 \\ 4.2222 \\ 4.8889 \\ 6.0000 \end{bmatrix} \tag{3.6}$$

$$A^{\dagger} \rightarrow \begin{bmatrix} -0.8994 & 0.4348 & 0.0652 & 0.0652 & 0.4348 \\ 0.4348 & 2.0686 & -1.6490 & -0.5292 & 0.1096 \\ 0.0652 & -1.6490 & 3.3774 & -1.1992 & -0.5292 \\ 0.0652 & -0.5292 & -1.1992 & 3.3774 & -1.6490 \\ 0.4348 & 0.1096 & -0.5292 & -1.6490 & 2.0686 \end{bmatrix} \tag{3.7}$$

$$A^{\dagger}B \rightarrow S \rightarrow \begin{bmatrix} 4.9421 \\ -0.6177 \\ -1.3737 \\ -0.5625 \\ 2.5538 \end{bmatrix} \tag{3.8}$$

From the first element of S we get the bias

$$b \rightarrow 4.9421 \tag{3.9}$$

The rest of the elements are our learned multipliers

$$\alpha \rightarrow \begin{bmatrix} -0.6177 \\ -1.3737 \\ -0.5625 \\ 2.5538 \end{bmatrix} \tag{3.10}$$

3.2 Prediction

We are predicting v for U

$$U \rightarrow \begin{bmatrix} 0.3 \\ 0.2 \\ 0.5 \end{bmatrix} \tag{3.11}$$

Computing Ω from the kernel we used in training and the training data we get

$$\Omega \rightarrow \begin{bmatrix} 0.9560 & 0.9994 & 0.9350 & 0.7827 \\ 0.9802 & 0.9912 & 0.8968 & 0.7261 \\ 0.8825 & 0.9862 & 0.9862 & 0.8825 \end{bmatrix} \tag{3.12}$$

Matrix multiplication of Ω with the multipliers we solved for in training we get

$$\Omega\alpha \rightarrow \begin{bmatrix} -0.4904 \\ -0.6170 \\ -0.2008 \end{bmatrix} \tag{3.13}$$

Finaly the after adding the bias to each input we get

$$\Omega\alpha + b\mathbf{1}_m \rightarrow v \rightarrow \begin{bmatrix} 4.4516 \\ 4.3251 \\ 4.7413 \end{bmatrix} \tag{3.14}$$

Where $\mathbf{1}_m$ is a vector of 1s with the length of U.