# EE6550 Machine Learning, Spring 2017

## Homework Assignment #4

Please submit your solutions of the word problems in class (during the first recess time) on May 8th Monday and your programs, including source codes, report (in pdf format) and user manual (in pdf format), to iLMS by 23:59 on May 10th Wednesday. Late submission will not be accepted unless the instructor gives a pre-approval. You are encouraged to consult or collaborate with other students while solving the problems, but you will have to turn in your own solutions and programs with your own words and work. If you find any resources in the internet to assist you, you should understand but not copy them. Copying will not be tolerated.

**Part I:** Word problem set.

1. (1.5%) With a PDS kernel $K$ over the input space $\mathscr{I}$, a feature mapping $\Phi$ of the input space $\mathscr{I}$ can be taken to be the mapping from $\mathscr{I}$ to the RKHS $\mathbb{H}$ of the kernel $K$ such that $\Phi(\omega) = K(\omega, \cdot)$ for all $\omega \in \mathscr{I}$ and $\langle \Phi(\omega), \Phi(\omega') \rangle_{\mathbb{H}} = K(\omega, \omega')$. The primal problem of SVR on page 79 of Lecture 8: Regression for feature vectors in $\mathbb{R}^N$ can be extended to one for feature vectors in the RKHS $\mathbb{H}$ of the kernel $K$ as follows:

$$\begin{array}{ll} \text{Minimize} & F(h, b, \eta, \eta') = \frac{1}{2}\|h\|_{\mathbb{H}}^2 + C\sum_{i=1}^{m}(\eta_i + \eta_i') \\ \text{Subject to} & (\langle h, \Phi(\omega_i)\rangle_{\mathbb{H}} + b) - c(\omega_i) - \epsilon - \eta_i \leq 0, \ i \in [1, m] \\ & c(\omega_i) - (\langle h, \Phi(\omega_i)\rangle_{\mathbb{H}} + b) - \epsilon - \eta_i' \leq 0, \ i \in [1, m] \\ & -\eta_i \leq 0, \ i \in [1, m] \\ & -\eta_i' \leq 0, \ i \in [1, m] \\ & (h, b, \eta, \eta') \in \mathbb{H} \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m, \end{array}$$

where $S = (\omega_1, \omega_2, \ldots, \omega_m)$ is a random sample of size $m$ drawn i.i.d. from the input space $\mathscr{I}$ according to an unknown distribution $P$ with labels $(c(\omega_1), c(\omega_2), \ldots, c(\omega_m))$. Please use the Representer Theorem (Theorem 5.4) of Lecture 4: Kernel Methods to show that the primal problem in above is equivalent to

$$\begin{array}{ll} \text{Minimize} & F(h, b, \eta, \eta') = \frac{1}{2}\|h\|_{\mathbb{H}_S}^2 + C\sum_{i=1}^{m}(\eta_i + \eta_i') \\ \text{Subject to} & (\langle h, \Phi(\omega_i)\rangle_{\mathbb{H}_S} + b) - c(\omega_i) - \epsilon - \eta_i \leq 0, \ i \in [1, m] \\ & c(\omega_i) - (\langle h, \Phi(\omega_i)\rangle_{\mathbb{H}_S} + b) - \epsilon - \eta_i' \leq 0, \ i \in [1, m] \\ & -\eta_i \leq 0, \ i \in [1, m] \\ & -\eta_i' \leq 0, \ i \in [1, m] \\ & (h, b, \eta, \eta') \in \mathbb{H}_S \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m, \end{array}$$

where

$$\begin{aligned} \mathbb{H}_S &\triangleq \text{Span}\{K(\omega_j, \cdot), j = 1, 2, \ldots, m\} \\ &= \left\{ \sum_{j=1}^{m} \alpha_j K(\omega_j, \cdot) \mid \alpha_j \in \mathbb{R}, \ 1 \leq m \leq m \right\}, \end{aligned}$$

is a finite-dimensional subspace of the RKHS $\mathbb{H}$ spanned by the feature vectors $\Phi(\omega_j) = K(\omega_j, \cdot)$ in $\mathbb{H}$ corresponding to items $\omega_j$ in the random sample $S$.

2. In this problem, you will derive the update rule of an extended version of the sequential minimal optimization (SMO) algorithm which is used to implement an efficient SVR-learning algorithm for regression. Please see the article *A Supplement of Kernel-based SMO-SVR Algorithm for HW#4 Programming Problem*, which will be referred to as *the article*.

   (a) (0.5%) Assume that we want to solve the Lagrangian dual problem in Eq. (34) of *the article* only over two variables $\beta_i$ and $\beta_j$, by fixing the values of other variables $\beta_k, k \neq i, j$ to their most recently updated values. Please show that the dual problem in Eq. (34) of *the article* can be reduced to the dual problem in Eq. (35) of *the article*.

   (b) (0.5%) Please show that the derivative $\frac{\Psi_2(\beta_j)}{d\beta_j}$ in Eq. (37) of *the article* has a jump of size $2\epsilon$ at $\beta_j = 0, \gamma_{ij}^*$.

   (c) (0.5%) When $\eta_{ij} = 0$, please show that the dual problem in Eq. (36) becomes the dual problem in Eq. (44).

**Part II:**

(12%) Programming problem:

Implementation of an extended version of the sequential minimal optimization (SMO) algorithm for kernel-based support vector regression (SVR). Please refer to *the article*.

**Input:**

1. A data file which contains a labeled training sample $S$. This labeled training sample is used to train the kernel based SMO-SVR algorithm which will return a hypothesis $h_S^{SVR}$ after $n$-fold cross-validation.
2. A data file which contains a labeled testing sample $\tilde{S}$. This labeled testing sample is used to evaluate the performance of the returned hypothesis $h_S^{SVR}$ from the SMO-SVR algorithm based on the labeled training sample $S$.
3. Set the values of the insensitivity parameter $\epsilon$ and the tolerance $\tau$, see *the article*.
   - It is reasonable to preset the tolerance $\tau$ to be 1%-10% of $\epsilon$.
   - It is suggestive to set $\epsilon$ to be 0.1 or 1%-10% of the range of the labels of the items in the sample $S$.
4. Choice of the kernel function $K(\mathbf{x}, \mathbf{x}')$. There are two choices:
   (a) The polynomial kernel of degree $d$: $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^d$. (See pages 13-14 of Lecture 4 on Kernel with $c = 1$.)
   (b) The Gaussian kernel: $K(\mathbf{x}, \mathbf{x}') = \exp\left\{\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right\}$.
5. Choice of $n$-fold cross-validation, where $n = 5$ or $n = 10$.
   The free model parameter vector $\boldsymbol{\theta}$ is $(C, d)$ if the chosen kernel is the polynomial kernel of degree $d$ and $(C, \sigma)$ if the chosen kernel is the Gaussian kernel. As discussed in Lecture 1, we use $n$-fold cross-validation to determine the best value of the free parameter vector $\boldsymbol{\theta}$.
   - Randomly partition a given training sample $S$ of $m$ labeled items into $n$ subsamples or folds.

- $((\omega_{i1}, c(\omega_{i1})), ..., (\omega_{im_i}, c(\omega_{im_i})))$: the $i$th fold of size $m_i$, $1 \leq i \leq n$.
  - Usually $m_i = \frac{m}{n}$ for all $i$.
- For any $i \in [1, n]$, the SMO-SVR algorithm is trained on all but the $i$th fold to generate a hypothesis $h_i$, and the performance of $h_i$ is tested on the $i$th fold.
- $\hat{R}_{CV}(\boldsymbol{\theta})$: the cross-validation error under the model parameter $\boldsymbol{\theta}$.

$$\hat{R}_{CV}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}|h_i(\omega_{ij}) - c(\omega_{ij})|_\epsilon = \frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{m_i}\max(0, |h_i(\omega_{ij}) - c(\omega_{ij})| - \epsilon).$$

- Choose the parameter vector $\boldsymbol{\theta}^*$ which minimizes the cross-validation error $\hat{R}_{CV}(\boldsymbol{\theta})$.
- Train the kernel-based SMO-SVR algorithm with the best parameter setting $\boldsymbol{\theta}^*$ over the full training sample $S$ of size $m$. The resulted hypothesis will be the returned hypothesis $h_S^{SVR}$ from the kernel-based SMO-SVR algorithm.

**Output:**

1. The optimal value $\boldsymbol{\theta}^*$ of the free parameter vector $\boldsymbol{\theta}$ with $\boldsymbol{\theta} = (C, d)$ for the polynomial kernel and $\boldsymbol{\theta} = (C, \sigma)$ for the Gaussian kernel (with the precision of the optimal value $C^*$ up to the 2nd decimal point) for both $n = 5$ and $n = 10$ by the $n$-fold cross-validation.
2. The hypothesis $h_S^{SVR}$ returned by the kernel-based SMO-SVR algorithm for both the polynomial kernel and the Gaussian kernel and for both $n = 5$ and $n = 10$ from the $n$-fold cross-validation,

$$h_S^{SVR}(\omega) = \sum_{k=1}^{m}\beta_k^{SVR}K(\omega_k, \omega) + b^{SVR}.$$

For the computation of the offset $b^{SVR}$ returned by the SMO-SVR algorithm, please see *the article*.

(a) To represent the hypothesis $h_S^{SVR}$ returned from the kernel-based SVM-learning algorithm, please output a .csv file with the file name SVR_hypothesis_header.csv by appending two additional columns with headers *beta* and *offset* respectively to the training data file, which already has several attribute columns and one output column, as follows: For the $i$th item $\omega_i$, append $\beta_i^{SVR}$ to the *lambda* column and, if $0 < |\beta_i^{SVR}| < C$, append $b_i^{SVR}$ to the *offset* column, where

$$b_i^{SVR} = \begin{cases} \epsilon - F_i^{SVR}, & \text{if } -C < \beta_i < 0, \\ -\epsilon - F_i^{SVR}, & \text{if } 0 < \beta_i < C \end{cases}$$

as stated in *the article*.

(b) Please output the sample mean $b^{SVR}$ and the sample standard deviation $\sigma_{b^{SVR}}$ of the offsets $b_j^{SVR}$,

$$\begin{aligned} b^{SVR} &= \frac{1}{m_b} \sum_{j=1, 0<|\beta_j^{SVR}|<C}^{m} b_j^{SVR}, \\ \sigma_{b^{SVR}} &= \sqrt{\frac{1}{m_b - 1} \sum_{j=1, 0<|\beta_j^{SVR}|<C}^{m} (b_j^{SVR} - b^{SVR})^2} \end{aligned}$$

where $m_b = \sum_{j=1,0<|\beta_j^{SVR}|<C}^{m} 1$ is the number of items $\omega_j$ such that $0 < |\beta_j^{SVR}| < C$. (Note: For a good learning result, $\sigma_{b^{SVR}}$ should be small.)

3. Performance of the returned hypothesis $h_S^{SVR}$ on the labeled testing sample $\tilde{S}$ for both the polynomial kernel and the Gaussian kernel and for both $n = 5$ and $n = 10$ in the $n$-fold cross-validation.

**What to submit?** You should submit the following items:

1. The source codes of your kernel-based SMO-SVR learning algorithm.
   Please indicate which programming language you have used to write the programs in the title of the submission entry with one of the following formats:
   - HW4_yourstudentid_yourname_matlab (The environment you use should be compatible with the version licensed to the NTHU Computer Center.)
   - HW4_yourstudentid_yourname_python36
   - HW4_yourstudentid_yourname_cpp14
   - HW4_yourstudentid_yourname_c11

   This will facilitate the distribution of homeworks to graders for grading. Please have your code compilable/interpretable by a standard compiler/interpreter environment: Matlab (NTHU CC), Python3.x, C++14, and C11.

2. A printed report consisting of at least:
   (a) a table of the cross-validation error $\hat{R}_{CV}(\boldsymbol{\theta})$ as a function of the parameter vector $\boldsymbol{\theta}$ for both the polynomial kernel and the Gaussian kernel and for both $n = 5$ and $n = 10$ by the $n$-fold cross-validation and a discussion of how you determine the optimal value $\boldsymbol{\theta}^*$ from such a table. (with the precision of the optimal parameter $C^*$ up to the 2nd decimal point);

   (b) an output file which represents the hypothesis $h_S^{SVR}$ returned by the kernel-based SMO-SVR learning algorithm as well as the sample mean $b^{SVR}$ and the sample standard deviation $\sigma_{b^{SVR}}$ of the offsets with the optimal parameter vector $\boldsymbol{\theta}^*$ for both the polynomial kernel and the Gaussian kernel and for both $n = 5$ and $n = 10$ from the $n$-fold cross-validation. In total, there are four such files;

   (c) the performance of the returned hypothesis $h_S^{SVR}$ on the labeled testing sample $\tilde{S}$ for both the polynomial kernel and the Gaussian kernel and for both $n = 5$ and $n = 10$ in the $n$-fold cross-validation.

   Please submit your report in pdf format (please do not submit word files).

3. A user manual which should include instructions of

    (a) how to compile the source code with a standard compiler/interpreter;

    (b) how to run, i.e., execute the kernel-based SMO-SVR learning algorithm, including the required formats of input parameters or data files;

    (c) how to use an output file which represents a returned hypotheses to do the testing;

    (d) what results are reported.

    These instructions should support the test scenarios in the grading session. Please submit your manual in pdf format (please do not submit word files).

**Test scenarios in the grading session:** the grader will test your algorithm with your source codes by the following procedure:

1. inputting a training data file, which contains a labeled training sample $S$. This labeled training sample is used to train the kernel-based SMO-SVR learning algorithm which will return a hypothesis $h_S^{SVR}$ after the $n$-fold cross-validation.
2. inputting a testing data file, which contains a labeled testing sample $\tilde{S}$. This labeled testing sample is used to evaluate the performance of the returned hypothesis $h_S^{SVR}$ from the kernel-based SMO-SVR learning algorithm based on the labeled training sample $S$.
3. choosing a PDS kernel. The free parameter vector $\boldsymbol{\theta}$ is $(C, d)$ for the polynomial kernel and $(C, \sigma)$ for the Gaussian kernel.
4. inputting a positive integer $n$ to perform $n$-fold cross-validation to determine the optimal parameter vector $\boldsymbol{\theta}^*$ of the free parameter $\boldsymbol{\theta}$ to minimize the cross-validation error $\hat{R}_{CV}(\boldsymbol{\theta})$.
5. checking the output table of the cross-validation error $\hat{R}_{CV}(\boldsymbol{\theta})$ as a function of the parameter vector $\boldsymbol{\theta}$ and the obtained optimal value $\boldsymbol{\theta}^*$ with a prepared reference table and optimal parameter vector value (with the precision of the optimal parameter $C^*$ up to the 2nd decimal point).
6. inputting the obtained optimal value $\boldsymbol{\theta}^*$ to a prepared reference kernel-based SMO-SVR learning program to return a learned hypothesis $h_S^{SVR}$ represented by an output file grading_SVR_hypothesis_header.csv as well as the sample mean $b^{SVR}$ and the sample standard deviation $\sigma_{b^{SVR}}$ of the offset.
7. checking the output file SVR_hypothesis_header.csv which represents the hypothesis $h_S^{SVR}$ returned by student's kernel-based SMO-SVR learning algorithm as well as the sample mean $b^{SVR}$ and the sample standard deviation $\sigma_{b^{SVR}}$ of the offset with the file grading_SVR_hypothesis_header.csv generated from the prepared reference program as well as the generated sample mean $b^{SVR}$ and the sample standard deviation $\sigma_{b^{SVR}}$ of the offset.
8. checking the performance of the returned hypothesis $h_S^{SVR}$ on the labeled testing sample $\tilde{S}$ by inputting the student's output file SVR_hypothesis_header.csv to a prepared program.