Introduction to Machine Learning (67577)

Problem Set 4 – Support Vector Machines and Validation

Due: 2.6.2017

Submission guidelines

- The assignment is to be submitted via Moodle only.
- Files should be zipped to ex_N_First_Last.zip (In this case N stands for 4, First is your first name, Last is your last name. In English, of course).
- The .zip file size should not exceed 10Mb.
- All answers should be submitted as a single pdf file. We encourage you to typeset your answers (either LaTeX, Lyx or Word), but it is not mandatory. Note that if you choose to scan a handwritten solution, then answers written in incomprehensible handwriting or scanned in poor quality will be counted as wrong answers. Moreover, the scans should be integrated with the plots to a single coherent pdf file.
- All of your Python code files used in the practical question should be attached in the zip file. There should be one .py file, other then that there are no special requirements or guidelines for your code. Note that if your code doesn't run on the CS environment you will not get points for this question.

1 Equivalent Definitions for Soft-SVM (10 points)

In the lecture Soft-SVM was introduced as solving the the optimization problem:

$$\min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hinge}(y_i\langle \mathbf{w}, \mathbf{x}_i\rangle),$$

where $\ell^{hinge}(a) = \max\{0, 1-a\}$.

In many books Soft-SVM is introduced as solving the the optimization problem:

$$\min_{\mathbf{w},\{\xi_i\}} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \quad \text{such that } \forall_i, y_i\langle \mathbf{w}, \mathbf{x}_i\rangle \geqslant 1-\xi_i \text{ and } \xi_i \geqslant 0.$$

Show that these two optimization problems yield the same solution.

2 Valid Kernel (10 points)

Let $M < N$ be any two positive integers. For every $x, x' \in \{M, \ldots, N\}$ define

$$K(x, x') = \min\{x, x'\}.$$

Prove that K is a valid kernel; namely, find a mapping $\psi : \{M, \ldots, N\} \to H$ such that

$$\forall x, x' \in \{M, \ldots, N\}, K(x, x') = \langle \psi(x), \psi(x')\rangle.$$

3 Model Selection (30 points)

In this question we will see when is the model selection paradigm beneficial compared to the standard method. To be concrete, we will focus on the case where you are not sure which of k possible hypothesis class to choose from: $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots \subseteq \mathcal{H}_k$, where $\mathcal{H}_k$ is finite.

Suppose we are given m examples $S_{all} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ and, as usual, we would like to learn a hypothesis with small generalization error. In class we discussed the polynomial fitting problem where we had k hypothesis classes to choose from $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots \subseteq \mathcal{H}_k$. In this question we compare two methods for choosing a hypothesis:

1. *Standard Method:* Find the best hypothesis in $\mathcal{H}_k$ using all the m training examples.
2. *Model Selection:* Do the following steps:
    - Divide the m examples into a training set S with size $(1-\alpha)m$ and a validation set V of size αm for some $\alpha \in (0,1)$.
    - For each hypothesis class $\mathcal{H}_i$, $i \in [k]$, find $h_i \in ERM_{\mathcal{H}_i}(S)$
    - Return $h^* \in ERM_{\mathcal{H}}(V)$, where $\mathcal{H} = \{h_1, \ldots, h_k\}$

Assume $\mathcal{H}_k$ is finite and the loss function is bounded by 1.

1. Bound the generalization error using the standard method. Namely, prove that agnostically PAC learning $\mathcal{H}_k$ provides the following bound: for $h^* \in ERM_{\mathcal{H}_k}(S_{all})$, with probability at least $1-\delta$,

$$L_{\mathcal{D}}(h^*) \leqslant \min_{h \in \mathcal{H}_k} L_{\mathcal{D}}(h) + \sqrt{\frac{2\ln(2|\mathcal{H}_k|/\delta)}{m}}.$$

2. Bound the generalization error using model selection. Namely, suppose that $\operatorname{argmin}_{h\in\mathcal{H}_k} L_{\mathcal{D}}(h)$ comes from $\mathcal{H}_j$ for some $j \in [k]$ (this implies that $\operatorname{argmin}_{h\in\mathcal{H}_k} L_{\mathcal{D}}(h) \in \mathcal{H}_{j+1}, \mathcal{H}_{j+2}, \ldots, \mathcal{H}_k$). Prove that with probability at least $1-\delta$,

$$L_{\mathcal{D}}(h^*) \leqslant \min_{h\in\mathcal{H}_k} L_D(h) + \sqrt{\frac{2}{\alpha m} \ln\frac{4k}{\delta}} + \sqrt{\frac{2}{(1-\alpha)m} \ln\frac{4|\mathcal{H}_j|}{\delta}} \ .$$

3. Show that the two bounds are incomparable: describe a case where the standard method is better and a case where model selection is better.

   To further compare the two methods, denote the bounds you got on the estimation error (i.e., $L_{\mathcal{D}}(h^*) - \min_{h\in\mathcal{H}_k} L_{\mathcal{D}}(h)$) of the standard method and model selection by ϵ_{est}^{S} and ϵ_{est}^{MS}, respectively. Show that

$$\frac{\epsilon_{est}^{MS}}{\epsilon_{est}^{S}} = \sqrt{\frac{\ln(4k/\delta)}{\alpha\ln(2|\mathcal{H}_k|/\delta)}} + \sqrt{\frac{\ln(4|\mathcal{H}_j|/\delta)}{(1-\alpha)\ln(2|\mathcal{H}_k|/\delta)}}$$

   while $\frac{\epsilon_{est}^{S}}{\epsilon_{est}^{MS}}$ can be arbitrarily large. This means that while model selection can be worse than the standard model, it cannot be too bad. On the other hand, we cannot say the same thing for the standard method.

4 Polynomial Fitting Using The Least Squares Algorithm (50 points)

In this exercise we will use the Least Squares algorithm in order to perform polynomial fitting. The exercise will demonstrate the Bias-complexity tradeoff.

Both the domain and the label set are equal to $\mathbb{R}$. The prior knowledge is that the relation between the instances and their labels can be approximately explained by a polynomial of degree $d \in [15] := \{1, \ldots, 15\}$. For each $d \in [15]$, let $\mathcal{H}_d$ be the class of polynomials of degree d. Your task is to train each of the classes over the (same) training sequence, perform validation over the 15 resulting hypotheses in order to choose the final output. Finally, you will test the performance of the resulting predictor over the test sequence. Here are the exact details.

1. Download the two text files `X_poly.npy`, and `Y_poly.npy`, which correspond to the instances and their corresponding labels.

2. The overall number of instances is 300. Divide these sets into 3 equal size sets (training, validation, and test sequences).

3. Train each of the classes using the training sequence to obtain for each $d \in [15]$, a single hypothesis h_d which minimizes the loss over the training sequence.

4. Perform validation over the set $\{h_1, \ldots, h_{15}\}$ to obtain a single h^* which minimizes the averaged loss over the validation set.

5. Test the performance of h^* over the test sequence. That is, calculate the averaged test error of h^*. Write your results.

6. Run the k-fold cross validation algorithm with $k = 5$ and the first 200 examples (i.e., the training and validation examples). Does it return h^*?

7. Plot the train and the validation errors of each hypothesis in $\{h_1, \ldots, h_{15}\}$.