# IML 2019 - ex 5 - feature and model selection

May 2019

## Contents

## Submission guidelines

- submission is due 6.6.2019
- The assignment is to be submitted via Moodle only.
- Files should be zipped to **ex_5_First_Last.zip** (First is your first name, Last is your last name. In English, of course).
- The .zip file size should not exceed 10Mb.
- Each question is enumerated, use these numbers when answering your questions.

## Theoretical part

### Model selection

In this question we will see when the model selection paradigm has a benefit over the standard method when choosing between $k$ possible hypothesis classes: $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots \subseteq \mathcal{H}_k$, 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:

- *Standard Method:* Find the best hypothesis in $\mathcal{H}_k$ using all the $m$ training examples.
- *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 $\alpha m$ for some $\alpha \in (0,1)$ (assume that $\alpha m$ is an integer).
  - 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. (4 Pts) 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 $1-\delta$,

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

   *Hint*: Use Hoeffding and the union bound.

2. (4 Pts) 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 $1-\delta$,

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

*Hint*: Use the previous item on the training step and on the validation step (switch in both cases $\delta \to \delta/2$), and recall that the probability of two independent events equals the product of their individual probabilities $P(A \cap B) = P(A)P(B)$ .

3. (7 Pts) Show that the two bounds are incomparable: describe a case where the standard method is better and a case where model selection is better, in terms of the generalization error.

   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 $\epsilon_{est}^{S}$ and $\epsilon_{est}^{MS}$, 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.
   *Hint*: try to think of nested hypothesis classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots \subseteq \mathcal{H}_k$ such that the size of each class is grows extremely fast (more than exponentially fast) with its index.

### Feature selection and Pearson's correlation coefficient

4. (Optional) Prove exercise 1 from recitation 10
5. (Optional) Prove exercise 2 from recitation 10

### Feature transformations

Features transformations may decrease the approximation or estimation errors of our hypothesis class and may yield a faster algorithm. Similarly to the problem of feature selection, here again there are no absolute "good" and "bad" transformations, but rather each transformation that we apply should be related to the learning algorithm we are going to apply on the resulting feature vector as well as to our prior assumptions.

6. Consider linear regression with the squared loss. Let $a > 1$ be a large number, and suppose that the target $y$ is chosen uniformly at random from $\{\pm 1\}$. Set the single feature $x$ to be $y$ with probability $(1 - 1/a)$ and $ay$ with probability $1/a$. Namely, most of the time our feature is bounded but with a very small probability it has a very large absolute value.
   (a) (2 Pts) Write an expression for $L_{\mathcal{D}}(w)$, the expected squared loss of $y = wx$
   (b) (2 Pts) Find the optimal value $w^* = \text{argmin}_w L_{\mathcal{D}}(w)$, and find $L_{\mathcal{D}}(w^*)$. What values do they approach as $a \to \infty$ ?
   (c) (2 Pts) Now, apply the following "clipping" transformation:

$$x \mapsto \text{sign}(x) \cdot \min(1, |x|)$$

   Following this transformation, find the optimal value $w^* = \text{argmin}_w L_{\mathcal{D}}(w)$, and find $L_{\mathcal{D}}(w^*)$. What values do they approach now as $a \to \infty$ ?.
   (d) (4 Pts) Give an example in which the same feature transformation actually hurts performance and increases the approximation error.

# Practical part

## Feature selection

In this question we will illustrate the limitations of feature ranking techniques which treat each feature individually and present several situations (in the context of binary classification) in which the feature dependencies cannot be ignored. In this section we are going to plot a lot of histograms. Here is a nice reference about doing this in Python: **Python Histogram Plotting**

### Can Presumably Redundant Features Help Each Other?

7. (18 Pts - all items have equal points) Perform the following steps:

   Warm-up: draw $m = 1000$ points out of the scalar distribution $x \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$. Plot on the same panel the PDF of this distribution and the histogram of points you sampled.

   (a) Draw two populations of samples, each containing $m = 1000$ points, out of the distribution $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$, where for both $\Sigma = I_2$ (the 2-D identity matrix) but with different centers: $\mu_1 = (1, 1)$ and $\mu_2 = (-1, -1)$.

   (b) Plot a scatter plot of the points in the 2-D plane representing the two features, where the points corresponding to the two populations are marked by different colors.

   (c) In one panel, plot a histogram for each of the two populations (again, using two different colors) representing the first feature $x_1$. In a second panel, do the same for the second feature $x_2$. (so you should have two panels, and two histograms on each panel). Notice the overlap between the two histograms in each panel.

   (d) Rotate the points in the scatter plot by 45 degrees and repeat the two previous items (e.g. using polar coordinates $(r, \theta)$ and adding $\pi/4$ radians to the angle $\theta$). Compare the overlaps between the histograms to the ones in the previous item: what do you see?

   (e) Is it now easier or harder to separate the two populations? Give an explanation.

### How Does Correlation Impact Feature Redundancy?

8. (9 Pts - all items have equal points)

   (a) Repeat items (a)-(c) of the previous question, only this time - use a covariance matrix $\Sigma$ (which is the same for the two populations) where the two features have a strong *positive* correlation- its EVD is: $\Sigma = VDV^T$ where the eigenvectors are $\frac{1}{\sqrt{2}}(1, 1)$ and $\frac{1}{\sqrt{2}}(1, -1)$ and the eigenvalues, in respective order, are 2 and 0.01.

   (b) Repeat item (a) of this question but this time use a covariance matrix $\Sigma$ (which is the same for the two populations) where the two features have a strong *negative* correlation- use the same eigenvectors but the eigenvalues, in respective order, are now 0.01 and 2.

(c) In both items (a) and (b) of this question, the overlap between the histograms (on both panels) should be roughly the same. Now observe the scatter plots: only in one of them a perfect separation can be achieved in the two-dimensional space spanned by the two features, whereas in the other it is not possible to separate the two populations - which panel is which?
Consider a method that scores each feature individually and independently of others. Can we trust it to do well in determining which combination of features would give best performance? Explain.

**Can a Feature that is Useless by Itself be Useful with Others?**

9. (8 Pts - all items have equal points)

   (a) Repeat items (a)-(c) of question 7, only this time - use a covariance matrix $\Sigma$ (which is the same for the two populations) where the eigenvectors are $\frac{1}{2}(\sqrt{3}, 1)$ and $\frac{1}{2}(1, -\sqrt{3})$ and the eigenvalues, in respective order, are 2 and 0.01. The centers of the two populations are: $\mu_1 = (0, 3/2)$ and $\mu_2 = (0, -3/2)$.

   (b) Explain why this shows that a feature that is completely useless by itself can provide a significant performance improvement when taken with others.

## Model selection on Polynomial Fitting

10. (40 Pts - all items have equal points) In this exercise we will perform polynomial fitting. Although we have seen this several times in the course, here the focus will be on the details of cross-validation.

    Both the domain and the label set are real scalars: $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}$.

    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 training set and 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 set. Here are the exact details.

    (a) Generate a data set according to the following:
        i. $\mathcal{X} = [-3.2, 2.2]$ and $x$ is uniformly sampled from this domain.
        ii. The relation between $y$ and $x$ is $y(x) = f(x) + \epsilon$ where
        $$f(x) = (x+3)(x+2)(x+1)(x-1)(x-2)$$
        and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
        iii. Take $\sigma = 1$ and generate $m = 1500$ instances.
        iv. Divide the $m = 1500$ instances into 2 sets: a 1000 for (training + validation) (call it $D$), and 500 for a test set (call it $T$) (you should not touch $T$ until you reach item (g)).

    (b) Split $D$ into two sets of 500 instances each - one for training ($S$) and one for validation ($V$): $D = S \cup V$. Train each of the classes using $S$ to obtain a single hypothesis $h_d$ for each $d \in [15]$, which minimizes the loss over $S$.

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

(d) Items (b)-(c) are $k$-fold cross validation with $k = 2$ (well, almost - each data point was only used once, either for training *or* for validation). Perform $k$-fold cross validation but now with $k = 5$ on the set $D$, which contains 1000 examples.

(e) Plot the training and validation errors (averaged over the $k$ folds) of the polynomials of degree $d \in [15]$ as a function of $d$. Which has the lowest validation error? Denote this by $d^*$.

(f) Perform $\text{ERM}_{\mathcal{H}_{d^*}}$ on the set $D$, namely find the polynomial of degree $d^*$ which has a minimal error on these examples, and denote this polynomial by $h^*$.

(g) Test the performance of $h^*$ over the test sequence $T$. That is, calculate the test error of $h^*$. Is it very different from the error you found in the previous item?

(h) Repeat all of the above steps for $\sigma = 5$. What has changed?

# Useful Python Commands

## Packages

- import numpy as np
- import matplotlib.pyplot as plt

## Reading Files

- np.loadtxt("filename.txt")
- np.load("filename.npy")

## Plots

- plt.plot(X)
- plt.scatter(X,Y)
- plt.show()

## Operations

- np.linalg.pinv(A)
- np.column_stack((u,v))
- np.concatenate((a,b))

## Generating Random Numbers

- np.random.uniform(0,1,1000)
- np.random.normal(mu, sigma, 1000)