Lecture 13: Validation

Presentation · July 2019

1 author:

Alaa Tharwat
Fachhochschule Bielefeld

Lecture 13: Validation

Alaa Tharwat

Lecture 13: Validation

- Review of Lecture 12
- Validation set
- Model selection
- Cross validation

Lecture 13: Validation

- Review of Lecture 12
- Validation set
- Model selection
- Cross validation

- Regularization technique is used for solving the overfitting problem by adding an extra term to the cost function

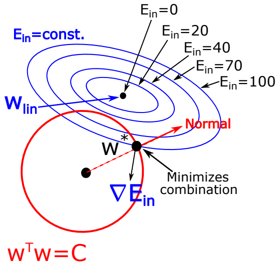
$$C = C_0 + \text{regularization term} = C_0 + \frac{\lambda}{2n} \sum_w w^2$$

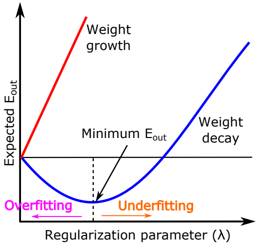
- The regularization term aims to make a balance between minimizing the original cost function and finding small weights
- With a small λ the original cost is minimized, but with a large λ the weights are minimized
- Increasing λ increases the bias (side effect) slightly and reduces the testing error dramatically. Hence, Large λ may lead to a simple model with high bias and high testing error

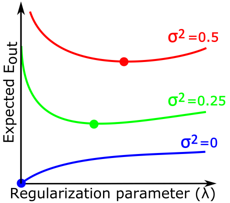
$$\text{minimize: } \frac{1}{N}(\mathbf{Zw} - Y)^T(\mathbf{Zw} - Y)$$
$$\text{subject to: } \mathbf{w}^T\mathbf{w} \leq C, \;\; \Rightarrow \mathbf{w}_{reg} \in H_C \text{ instead of } \mathbf{w}_{lin}$$

- The parameter C puts a constraint on some weights to be small or zero (not exclude any order but gives it different weights)
- With large $C \Rightarrow \lambda \approx 0$, w_{lin} is the solution, just minimize E_{in} as if there is no constraint
- With small $C \Rightarrow \lambda \uparrow$ and the regularization is more severe
- If $C = 0 \Rightarrow \lambda = \infty$ and $w \approx 0$
- Use the validation to get the optimal λ

$$\text{Min } E_{in}(w) + \frac{\lambda}{N} w^T w \qquad \boxed{C \uparrow \lambda \downarrow}$$







Lecture 13: Validation

- Review of Lecture 12
- Validation set
- Model selection
- Cross validation

In the last lecture:

$$E_{out}(h) = E_{in}(h) + \text{overfit penalty}$$

Regularization reduces the overfitting to estimate E_{out}, or we can say **Regularization** estimates this penalty

$$E_{out}(h) = E_{in}(h) + \underbrace{\text{overfit penalty}}_{\text{regularization estimates this term}}$$

Validation: estimates the E_{out}

$$\underbrace{E_{out}(h)}_{\text{validation estimates this term}} = E_{in}(h) + \text{overfit penalty}$$

- Assume we have only one out-of-sample point (x, y), the error is $e(h(x), y)$, where e is any error function[1], if we repeat this process many times we get many errors
- $E_{out}(h) = E[e(h(x), y)]$ (expectation of all errors)
- $Var[e(h(x), y)] = \sigma^2$ (variance of all errors)

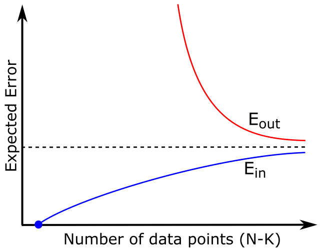
[1] Such as squared error function : $(h(x) - y)^2$ and binary error function : $(h(x) \neq y)$

- Instead of using one point, we use a set and we call it a *validation set* $D_{val} = (x_1, y_1), \ldots, (x_K, y_K)$, the error is $E_{val}(h) = \frac{1}{K}\sum_{k=1}^{K} e(h(x_k), y_k)$
- $E_{out}(h) = \frac{1}{K}\sum_{k=1}^{K} E[e(h(x_k), y_k)] = E[E_{val}(h)]$
- $Var[E_{val}(h)] = \frac{1}{K^2}\sum_{k=1}^{K} E[e(h(x_k), y_k)] = \frac{\sigma^2}{K}$ where K^2 is the number of samples in the covariance matrix[2]
- Hence, $E_{val}(h) = E_{out}(h) \pm O(\frac{1}{\sqrt{K}})$; this means that E_{val} is deviated from E_{out} by amount with order $O(\frac{1}{\sqrt{K}})$ (dependency on K)

K is not a free parameter because it is taken from N

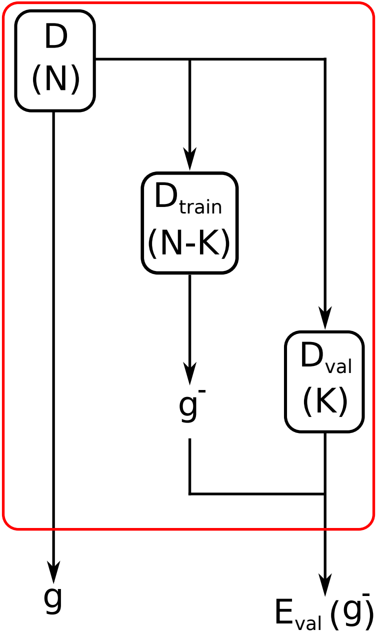
[2] Here, one summation to add the diagonal terms (variances) because all covariances are zeros because we pick the points independently

- Given dataset $D = (x_1, y_1), \dots, (x_N, y_N)$
- K samples/points are used for validation $\underbrace{(K \text{ points})}_{D_{val}}$
- $N - K$ samples are used for training $\underbrace{(N - K \text{ points})}_{D_{train}}$
  - With small $K \Rightarrow$ bad estimation. For example, we select two or three points, this will lead to a bad estimation and the validation error will not be reliable and the variance will be high. Also, with small $K \Rightarrow (N - K) \uparrow$ and $O(\frac{1}{\sqrt{K}}) \uparrow$ and hence E_{val} will be far from E_{out}
  - With large $K \Rightarrow$ the remain data for training the model is not enough $\Rightarrow$ overfitting, but $O(\frac{1}{\sqrt{K}}) \downarrow$ and hence $E_{val} \approx E_{out}$



$$\begin{array}{ccccc} D & \rightarrow & D_{train} & \cup & D_{val} \\ \downarrow & & \downarrow & & \downarrow \\ N & & N-K & & K \end{array}$$

- If we use the whole data for training $D \Rightarrow g$
- Practically, if we use the $(N-K)$ points for training we get g^{-} $(D_{train} \Rightarrow g^{-})$ and D_{val} is used for evaluating g^{-} $(E_{val} = E_{val}(g^{-}))$
- Can we put K back to traning data to get better approximation of E_{out}. No, because this makes a difference between g and g^{-}, and hence the estimation is bad

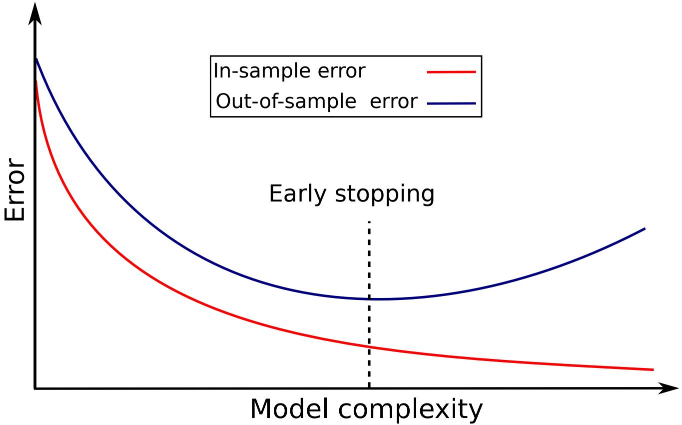


With large K:

- The training data will be small
- After the evaluation we can add the K samples again to the training data to increase the number of training samples. But, if K is large the change of training data will be severe and hence the validation error will be significantly different than the given data
- Large K $\Rightarrow$ bad estimation
- Practically, $K = \frac{N}{5}$

Why using validation?

- Validation is used to make many learning choices
- The figure below shows training and testing errors. Hence, we cannot estimate the stopping point to prevent overfitting
- A validation set is used for (adjust the models' parameters such as regularization parameter) and select the stopping point



What is the difference between test set and validation set?

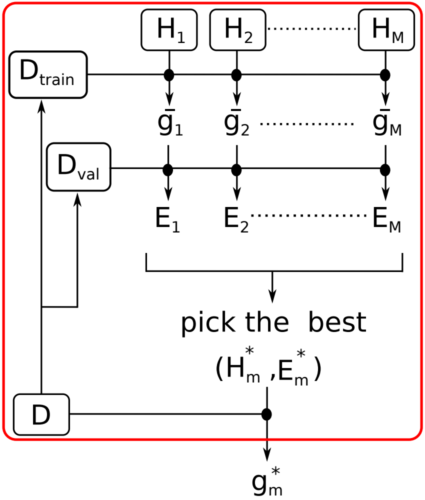
- Assume we have two hypotheses, h_1 and h_2, and each has the same $E_{out} = 0.5$
- Using one point to estimates that error: e_1 and e_2 uniform in $[0, 1]$
- Select one hypothesis $h \in \{h_1, h_2\}$ with $e = min(e_1, e_2)$; hence, $E(e) < 0.5$; thus, we can say the validation set obtains the minimum error and hence it has an optimistic bias

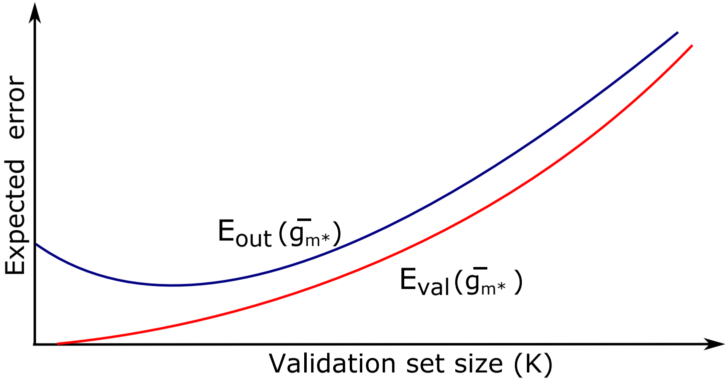
Lecture 13: Validation

- Review of Lecture 12
- Validation set
- Model selection
- Cross validation

We can use D_{val} more than once

- Given M models $H_1, \dots, H_M$
  - different learning algorithms such as SVM, NN, k-NN,..
  - one learning algorithm with different parameters (e.g. NN with different weights)
  - one model with different regularization parameters
- Use D_{train} to train g_m^- for each model $(g_1^-, g_2^-, \dots, g_M^-)$
- Validation set is used to evaluate all models ($E_m = E_{val}(g_m^-),\ m = 1, 2, \dots, M$) and select the best model (H_m^*) with the minimum error (E_m^*) (i.e. $m = m^*$)





- We selected the model H_m^* using the validation set (D_{val})
- $E_{val}(g_{m*}^{-})$ is a biased estimate of $E_{out}(g_{m*}^{-})$
- Increasing K reduces the training data and hence increases E_{out} and this makes E_{val} closer to E_{out}
- Small $K \Rightarrow D_{train} \uparrow \Rightarrow E_{out} \downarrow$
- E_{val} converges to E_{out} when K is large

- Given M models, $H_1, H_2, \ldots, H_M$
- D_{val} is used for training on the finalists model, $H_{val} = \{\bar{g_1}, \bar{g_2}, \ldots, \bar{g_M}\}$ (theses models form a hypotheses set of finallists or trained models)
- From Hoeffding and VC,

$$E_{out}(g_{m*}^{-}) \leq E_{val}(g_{m*}^{-}) + O(\sqrt{\frac{lnM}{K}})$$

- Hence, the regularization can be used for reducing the danger of overfitting and the validation can be used to find an early-stopping threshold
- We can say validation can be used for selecting the best regularization parameter

- We have three types of errors E_{in}, E_{out}, and E_{val}
- Data is contaminated if you use the data to make choices you are contaminating it as far as its ability to estimate the real performance
- What about contamination
  - Training set: totally contaminated (E_{in} is far from E_{out})
  - Testing set: totally clean (i.e. there is bias)
  - Validation set: slightly contaminated

Lecture 13: Validation

- Review of Lecture 12
- Validation set
- Model selection
- Cross validation

The following chain of reasoning:

$$\underset{\text{(small } K)}{E_{out}(g) \approx E_{out}(g^-)} \approx \underset{\text{(large } K)}{E_{val}(g^-)}$$

So, how we can select K? small or large?

In **leave-one-out** algorithm

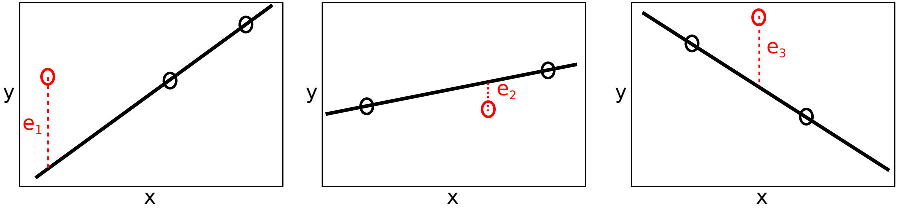
- $N-1$ points are used for training the model and only one point for validation, $D_n = (x_1, y_1), (x_2, y_2) \ldots, \cancel{(x_n, y_n)}, \ldots, (x_N, y_N)$, and the final hypothesis from D_n is g_n^-
- The validation error for one points is

$$e_n = E_{val}(g_n^-) = e(g_n^-(x_n), y_n)$$

Cross-validation error is

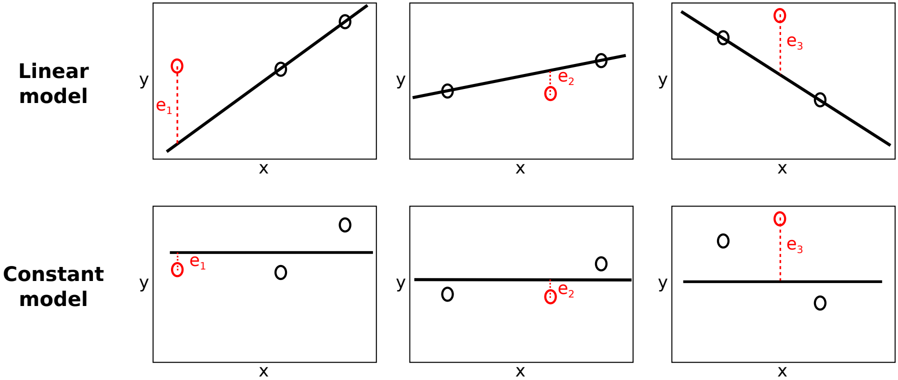
$$E_{cv} = \frac{1}{N} \sum_{n=1}^{N} e_n$$

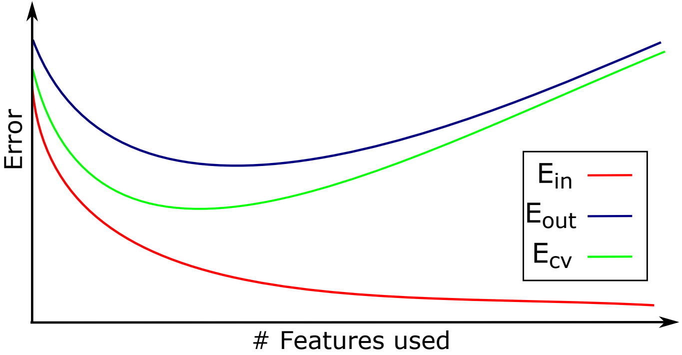
Illustration of cross-validation



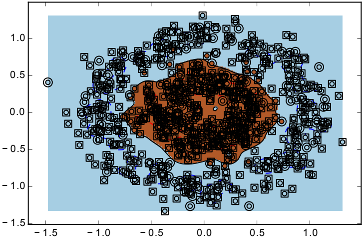
$$E_{cv} = (e_1 + e_2 + e_3)$$

How CV can be used in model selection?

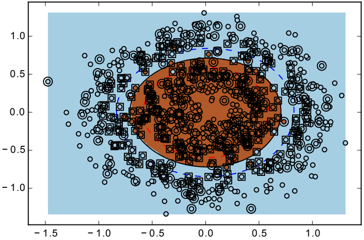




$$(1, x_1, x_2) \xrightarrow{mapping} (1, x_1, x_2, x_1^2, x_1 x_2, \ldots, x_1^5, x_1^4 x_2, x_1^3 x_2^2, x_1^2 x_2^3, x_1 x_2^4, x_2^5)$$



(a) Without validation
$E_{in} = 0.0015625\%$ and
$E_{out} = 2.8\%$



(b) With validation
$E_{in} = 0.0140625\%$ and
$E_{out} = 1.7\%$

- Without validation (i.e. using full model with all features), the decision boundary is sharp and $E_{out} \uparrow$
- With validation, the decision boundary is smooth and the model avoids the overfitting

- In leave one out method, $N-1$ samples are used for training
- In K-fold cross validation, the data is partitioned into K sets and one set is used for validation and the other sets for training the model. Here, we need $\frac{N}{K}$ training sessions/runs, and each has $N-K$ points