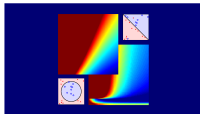


# Machine Learning Foundations

## (機器學習基石)



### Lecture 15: Validation

Hsuan-Tien Lin (林軒田)

[htlin@csie.ntu.edu.tw](mailto:htlin@csie.ntu.edu.tw)

Department of Computer Science  
& Information Engineering

National Taiwan University  
(國立台灣大學資訊工程系)



# Roadmap

- 1 When Can Machines Learn?
- 2 Why Can Machines Learn?
- 3 How Can Machines Learn?
- 4 How Can Machines Learn **Better**?

## Lecture 14: Regularization

minimizes **augmented error**, where the added **regularizer** effectively **limits model complexity**

## Lecture 15: Validation

- Model Selection Problem
- Validation
- Leave-One-Out Cross Validation
- V-Fold Cross Validation

# So Many Models Learned

## Even Just for Binary Classification . . .

$\mathcal{A} \in \{ \text{PLA, pocket, linear regression, logistic regression} \}$

×

$T \in \{ 100, 1000, 10000 \}$

×

$\eta \in \{ 1, 0.01, 0.0001 \}$

×

$\Phi \in \{ \text{linear, quadratic, poly-10, Legendre-poly-10} \}$

×

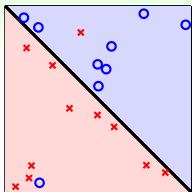
$\Omega(\mathbf{w}) \in \{ \text{L2 regularizer, L1 regularizer, symmetry regularizer} \}$

×

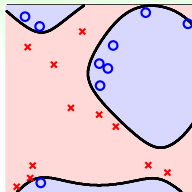
$\lambda \in \{ 0, 0.01, 1 \}$

in addition to your **favorite** combination, may need to try other combinations to get a good  $g$

# Model Selection Problem

 $\mathcal{H}_1$ 

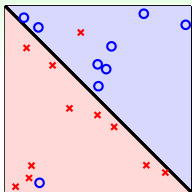
which one do you prefer? :-)

 $\mathcal{H}_2$ 

- given:  $M$  models  $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_M$ , each with corresponding algorithm  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_M$
- goal: select  $\mathcal{H}_{m^*}$  such that  $g_{m^*} = \mathcal{A}_{m^*}(\mathcal{D})$  is of low  $E_{\text{out}}(g_{m^*})$
- **unknown**  $E_{\text{out}}$  due to unknown  $P(\mathbf{x})$  &  $P(y|\mathbf{x})$ , as always :-)
- arguably the **most important** practical problem of ML

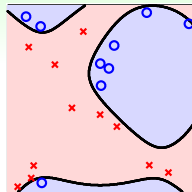
how to select? **visually?**  
—no, remember Lecture 12? :-)

# Model Selection by Best $E_{in}$

 $\mathcal{H}_1$ 

select by best  $E_{in}$ ?

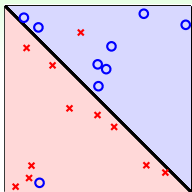
$$m^* = \operatorname{argmin}_{1 \leq m \leq M} (E_m = E_{in}(\mathcal{A}_m(\mathcal{D})))$$

 $\mathcal{H}_2$ 

- $\Phi_{1126}$  always more preferred over  $\Phi_1$ ;  
 $\lambda = 0$  always more preferred over  $\lambda = 0.1$ —**overfitting?**
- if  $\mathcal{A}_1$  minimizes  $E_{in}$  over  $\mathcal{H}_1$  and  $\mathcal{A}_2$  minimizes  $E_{in}$  over  $\mathcal{H}_2$ ,  
 $\implies g_{m^*}$  achieves minimal  $E_{in}$  over  $\mathcal{H}_1 \cup \mathcal{H}_2$   
 $\implies$  ‘**model selection** + learning’ pays  $d_{vc}(\mathcal{H}_1 \cup \mathcal{H}_2)$   
 —**bad generalization?**

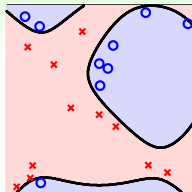
selecting by  $E_{in}$  is **dangerous**

# Model Selection by Best $E_{\text{test}}$

 $\mathcal{H}_1$ 

select by best  $E_{\text{test}}$ , which is evaluated on a fresh  $\mathcal{D}_{\text{test}}$ ?

$$m^* = \operatorname{argmin}_{1 \leq m \leq M} (E_m = E_{\text{test}}(\mathcal{A}_m(\mathcal{D})))$$

 $\mathcal{H}_2$ 

- generalization guarantee (finite-bin Hoeffding):

$$E_{\text{out}}(g_{m^*}) \leq E_{\text{test}}(g_{m^*}) + O\left(\sqrt{\frac{\log M}{N_{\text{test}}}}\right)$$

—**yes! strong guarantee :-)**

- but where is  $\mathcal{D}_{\text{test}}$ ?—**your boss's safe, maybe? :-)**

selecting by  $E_{\text{test}}$  is **infeasible** and **cheating**

# Comparison between $E_{\text{in}}$ and $E_{\text{test}}$

## in-sample error $E_{\text{in}}$

- calculated from  $\mathcal{D}$
- **feasible** on hand
- ‘contaminated’ as  $\mathcal{D}$  also used by  $\mathcal{A}_m$  to ‘select’  $g_m$

## test error $E_{\text{test}}$

- calculated from  $\mathcal{D}_{\text{test}}$
- **infeasible** in boss’s safe
- ‘clean’ as  $\mathcal{D}_{\text{test}}$  never used for selection before

## something in between: $E_{\text{val}}$

- calculated from  $\mathcal{D}_{\text{val}} \subset \mathcal{D}$
- **feasible** on hand
- ‘clean’ **if**  $\mathcal{D}_{\text{val}}$  never used by  $\mathcal{A}_m$  before

selecting by  $E_{\text{val}}$ : **legal cheating :-)**

# Fun Time

For  $\mathcal{X} = \mathbb{R}^d$ , consider two hypothesis sets,  $\mathcal{H}_+$  and  $\mathcal{H}_-$ . The first hypothesis set contains all perceptrons with  $w_1 \geq 0$ , and the second hypothesis set contains all perceptrons with  $w_1 \leq 0$ . Denote  $g_+$  and  $g_-$  as the minimum- $E_{\text{in}}$  hypothesis in each hypothesis set, respectively. Which statement below is true?

- 1 If  $E_{\text{in}}(g_+) < E_{\text{in}}(g_-)$ , then  $g_+$  is the minimum- $E_{\text{in}}$  hypothesis of all perceptrons in  $\mathbb{R}^d$ .
- 2 If  $E_{\text{test}}(g_+) < E_{\text{test}}(g_-)$ , then  $g_+$  is the minimum- $E_{\text{test}}$  hypothesis of all perceptrons in  $\mathbb{R}^d$ .
- 3 The two hypothesis sets are disjoint.
- 4 None of the above



# Fun Time

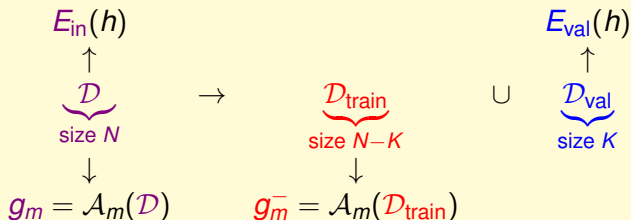
For  $\mathcal{X} = \mathbb{R}^d$ , consider two hypothesis sets,  $\mathcal{H}_+$  and  $\mathcal{H}_-$ . The first hypothesis set contains all perceptrons with  $w_1 \geq 0$ , and the second hypothesis set contains all perceptrons with  $w_1 \leq 0$ . Denote  $g_+$  and  $g_-$  as the minimum- $E_{\text{in}}$  hypothesis in each hypothesis set, respectively. Which statement below is true?

- ① If  $E_{\text{in}}(g_+) < E_{\text{in}}(g_-)$ , then  $g_+$  is the minimum- $E_{\text{in}}$  hypothesis of all perceptrons in  $\mathbb{R}^d$ .
- ② If  $E_{\text{test}}(g_+) < E_{\text{test}}(g_-)$ , then  $g_+$  is the minimum- $E_{\text{test}}$  hypothesis of all perceptrons in  $\mathbb{R}^d$ .
- ③ The two hypothesis sets are disjoint.
- ④ None of the above

Reference Answer: ①

Note that the two hypothesis sets are not disjoint (sharing ' $w_1 = 0$ ' perceptrons) but their union is all perceptrons.

# Validation Set $\mathcal{D}_{\text{val}}$



- $\mathcal{D}_{\text{val}} \subset \mathcal{D}$ : called **validation set**—‘on-hand’ simulation of test set
- to connect  $E_{\text{val}}$  with  $E_{\text{out}}$ :  
 $\mathcal{D}_{\text{val}} \stackrel{iid}{\sim} P(\mathbf{x}, y) \iff$  select  $K$  examples from  $\mathcal{D}$  at random
- to make sure  $\mathcal{D}_{\text{val}}$  ‘clean’:  
 feed only  $\mathcal{D}_{\text{train}}$  to  $\mathcal{A}_m$  for model selection

$$E_{\text{out}}(g_m^-) \leq E_{\text{val}}(g_m^-) + O\left(\sqrt{\frac{\log M}{K}}\right)$$

# Model Selection by Best $E_{\text{val}}$

$$m^* = \underset{1 \leq m \leq M}{\operatorname{argmin}} (E_m = E_{\text{val}}(\mathcal{A}_m(\mathcal{D}_{\text{train}})))$$

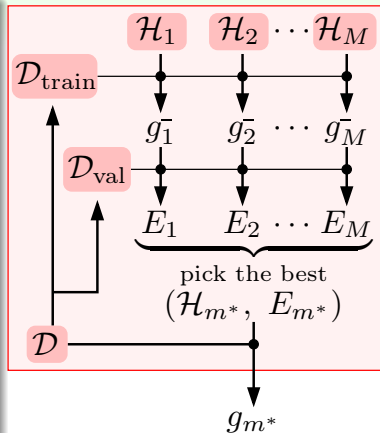
- generalization guarantee for all  $m$ :

$$E_{\text{out}}(\underline{g}_m) \leq E_{\text{val}}(\underline{g}_m) + O\left(\sqrt{\frac{\log M}{K}}\right)$$

- heuristic gain from  $N - K$  to  $N$ :

$$E_{\text{out}}\left(\underbrace{g_{m^*}}_{\mathcal{A}_{m^*}(\mathcal{D})}\right) \leq E_{\text{out}}\left(\underbrace{g_{m^*}^-}_{\mathcal{A}_{m^*}(\mathcal{D}_{\text{train}})}\right)$$

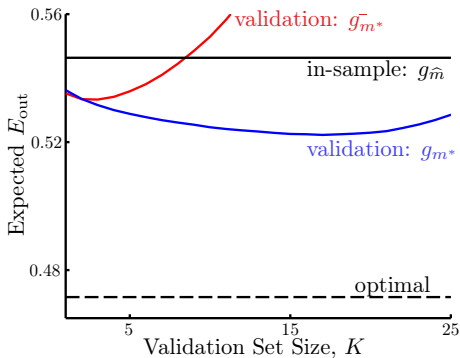
—learning curve, remember? :-)



$$E_{\text{out}}(\underline{g}_{m^*}) \leq E_{\text{out}}(\underline{g}_{m^*}^-) \leq E_{\text{val}}(\underline{g}_{m^*}^-) + O\left(\sqrt{\frac{\log M}{K}}\right)$$

# Validation in Practice

use validation to select between  $\mathcal{H}_{\Phi_5}$  and  $\mathcal{H}_{\Phi_{10}}$



- in-sample: selection with  $E_{\text{in}}$
- optimal: cheating-selection with  $E_{\text{test}}$
- **sub- $g$** : selection with  $E_{\text{val}}$  and report  $g_m^-$
- **full- $g$** : selection with  $E_{\text{val}}$  and report  $g_m^*$   
 —  $E_{\text{out}}(g_m^*) \leq E_{\text{out}}(g_m^-)$  indeed

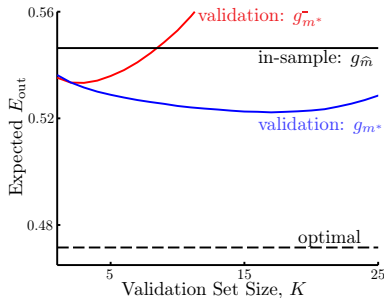
why is **sub- $g$**  worse than in-sample some time?

# The Dilemma about $K$

reasoning of validation:

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

- large  $K$ : **every**  $E_{\text{val}} \approx E_{\text{out}}$ ,  
but all  $g_m^-$  much worse than  $g_m$
- small  $K$ : every  $g_m^- \approx g_m$ ,  
but  $E_{\text{val}}$  far from  $E_{\text{out}}$



practical rule of thumb:  $K = \frac{N}{5}$

## Fun Time

For a learning model that takes  $N^2$  seconds of training when using  $N$  examples, what is the total amount of seconds needed when running the whole validation procedure with  $K = \frac{N}{5}$  on 25 such models with different parameters to get the final  $g_{m^*}$ ?

- ①  $6N^2$
- ②  $17N^2$
- ③  $25N^2$
- ④  $26N^2$

## Fun Time

For a learning model that takes  $N^2$  seconds of training when using  $N$  examples, what is the total amount of seconds needed when running the whole validation procedure with  $K = \frac{N}{5}$  on 25 such models with different parameters to get the final  $g_{m^*}$ ?

- ①  $6N^2$
- ②  $17N^2$
- ③  $25N^2$
- ④  $26N^2$

Reference Answer: ②

To get all the  $g_m^-$ , we need  $\frac{16}{25}N^2 \cdot 25$  seconds.  
Then to get  $g_{m^*}$ , we need another  $N^2$  seconds.  
So in total we need  $17N^2$  seconds.

# Extreme Case: $K = 1$

reasoning of validation:

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

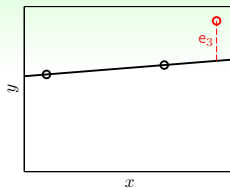
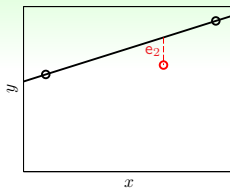
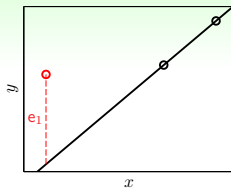
- take  $K = 1$ ?  $\mathcal{D}_{\text{val}}^{(n)} = \{(\mathbf{x}_n, y_n)\}$  and  $E_{\text{val}}^{(n)}(\textcolor{red}{g}_n^-) = \text{err}(\textcolor{red}{g}_n^-(\mathbf{x}_n), y_n) = \textcolor{violet}{e}_n$
- make  $\textcolor{violet}{e}_n$  closer to  $E_{\text{out}}(\textcolor{violet}{g})$ ?—average over possible  $E_{\text{val}}^{(n)}$
- leave-one-out cross validation estimate:

$$E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) = \frac{1}{N} \sum_{n=1}^N \textcolor{violet}{e}_n = \frac{1}{N} \sum_{n=1}^N \text{err}(\textcolor{red}{g}_n^-(\mathbf{x}_n), y_n)$$

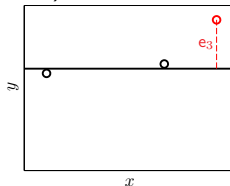
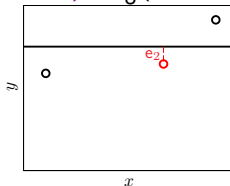
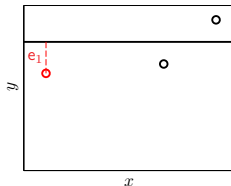
hope:  $E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) \approx E_{\text{out}}(\textcolor{violet}{g})$



# Illustration of Leave-One-Out



$$E_{\text{loocv}}(\text{linear}) = \frac{1}{3}(e_1 + e_2 + e_3)$$



$$E_{\text{loocv}}(\text{constant}) = \frac{1}{3}(e_1 + e_2 + e_3)$$

which one would you choose?

$$m^* = \underset{1 \leq m \leq M}{\operatorname{argmin}} (E_m = E_{\text{loocv}}(\mathcal{H}_m, \mathcal{A}_m))$$

# Theoretical Guarantee of Leave-One-Out Estimate

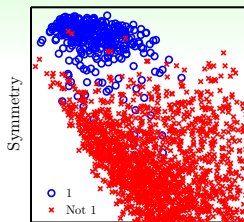
does  $E_{\text{loocv}}(\mathcal{H}, \mathcal{A})$  say something about  $E_{\text{out}}(g)$ ?

**yes, for average  $E_{\text{out}}$  on size- $(N - 1)$  data**

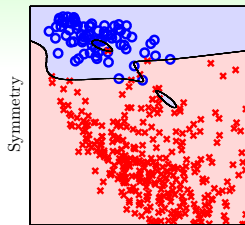
$$\begin{aligned}
 \mathbb{E}_{\mathcal{D}} E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) &= \mathbb{E}_{\mathcal{D}} \frac{1}{N} \sum_{n=1}^N e_n = \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{\mathcal{D}} e_n \\
 &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{\mathcal{D}_n(\mathbf{x}_n, y_n)} \mathbb{E} \text{err}(\mathbf{g}_n^-(\mathbf{x}_n), y_n) \\
 &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{\mathcal{D}_n} E_{\text{out}}(\mathbf{g}_n^-) \\
 &= \frac{1}{N} \sum_{n=1}^N \overline{E_{\text{out}}(N-1)} = \overline{E_{\text{out}}(N-1)}
 \end{aligned}$$

expected  $E_{\text{loocv}}(\mathcal{H}, \mathcal{A})$  says something about expected  $E_{\text{out}}(g^-)$   
 —often called ‘almost unbiased estimate of  $E_{\text{out}}(g)$ ’

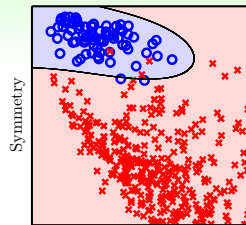
# Leave-One-Out in Practice



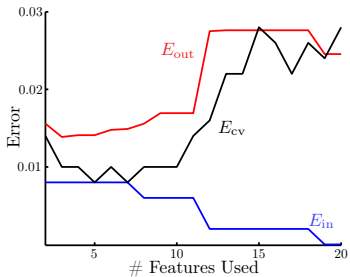
Average Intensity



Average Intensity

select by  $E_{in}$ 

Average Intensity

select by  $E_{loocv}$ 

$E_{loocv}$  much better than  $E_{in}$

## Fun Time

Consider three examples  $(\mathbf{x}_1, y_1)$ ,  $(\mathbf{x}_2, y_2)$ ,  $(\mathbf{x}_3, y_3)$  with  $y_1 = 1$ ,  $y_2 = 5$ ,  $y_3 = 7$ . If we use  $E_{\text{loocv}}$  to estimate the performance of a learning algorithm that predicts with the average  $y$  value of the data set—the optimal constant prediction with respect to the squared error. What is  $E_{\text{loocv}}$  (squared error) of the algorithm?

- 1 0
- 2  $\frac{56}{9}$
- 3  $\frac{60}{9}$
- 4 14

## Fun Time

Consider three examples  $(\mathbf{x}_1, y_1)$ ,  $(\mathbf{x}_2, y_2)$ ,  $(\mathbf{x}_3, y_3)$  with  $y_1 = 1$ ,  $y_2 = 5$ ,  $y_3 = 7$ . If we use  $E_{\text{loocv}}$  to estimate the performance of a learning algorithm that predicts with the average  $y$  value of the data set—the optimal constant prediction with respect to the squared error. What is  $E_{\text{loocv}}$  (squared error) of the algorithm?

- 1 0
- 2  $\frac{56}{9}$
- 3  $\frac{60}{9}$
- 4 14

Reference Answer: 4

This is based on a simple calculation of  $e_1 = (1 - 6)^2$ ,  $e_2 = (5 - 4)^2$ ,  $e_3 = (7 - 3)^2$ .

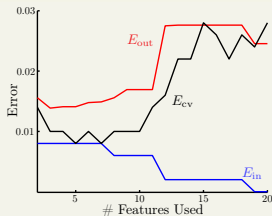
# Disadvantages of Leave-One-Out Estimate

## Computation

$$E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) = \frac{1}{N} \sum_{n=1}^N e_n = \frac{1}{N} \sum_{n=1}^N \text{err}(\mathbf{g}_n^-(\mathbf{x}_n), y_n)$$

- $N$  ‘additional’ training per model, not always feasible in practice
- except ‘special case’ like analytic solution for linear regression

## Stability—due to variance of single-point estimates

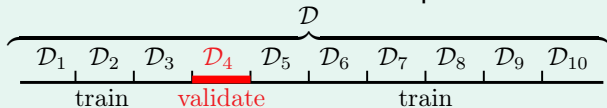


$E_{\text{loocv}}$ : not often used practically

# V-fold Cross Validation

how to **decrease computation need** for cross validation?

- essence of leave-one-out cross validation: partition  $\mathcal{D}$  to  $N$  parts, taking  $N - 1$  for training and 1 for validation orderly
- V-fold cross-validation: random-partition of  $\mathcal{D}$  to  **$V$  equal parts**,



**take  $V - 1$  for training and 1 for validation orderly**

$$E_{cv}(\mathcal{H}, \mathcal{A}) = \frac{1}{V} \sum_{v=1}^V E_{val}^{(v)}(g_{\bar{v}})$$

- selection by  $E_{cv}$ :  $m^* = \operatorname{argmin}_{1 \leq m \leq M} (E_m = E_{cv}(\mathcal{H}_m, \mathcal{A}_m))$

practical rule of thumb:  **$V = 10$**

# Final Words on Validation

## 'Selecting' Validation Tool

- **V-Fold** generally preferred over single validation if computation allows
- **5-Fold or 10-Fold** generally works well:  
not necessary to trade V-Fold with Leave-One-Out

## Nature of Validation

- all training models: select among hypotheses
- all validation schemes: **select among finalists**
- all testing methods: just **evaluate**

validation still **more optimistic than testing**

do not fool yourself and others :-),  
**report test result**, not **best validation result**



# Fun Time

For a learning model that takes  $N^2$  seconds of training when using  $N$  examples, what is the total amount of seconds needed when running 10-fold cross validation on 25 such models with different parameters to get the final  $g_{m^*}$ ?

- ①  $\frac{47}{2} N^2$
- ②  $47 N^2$
- ③  $\frac{407}{2} N^2$
- ④  $407 N^2$

## Fun Time

For a learning model that takes  $N^2$  seconds of training when using  $N$  examples, what is the total amount of seconds needed when running 10-fold cross validation on 25 such models with different parameters to get the final  $g_{m^*}$ ?

- ①  $\frac{47}{2} N^2$
- ②  $47 N^2$
- ③  $\frac{407}{2} N^2$
- ④  $407 N^2$

Reference Answer: ③

To get all the  $E_{cv}$ , we need  $\frac{81}{100} N^2 \cdot 10 \cdot 25$  seconds. Then to get  $g_{m^*}$ , we need another  $N^2$  seconds. So in total we need  $\frac{407}{2} N^2$  seconds.

# Summary

- 1 When Can Machines Learn?
- 2 Why Can Machines Learn?
- 3 How Can Machines Learn?
- 4 How Can Machines Learn **Better**?

## Lecture 14: Regularization

## Lecture 15: Validation

- Model Selection Problem  
dangerous by  $E_{\text{in}}$  and dishonest by  $E_{\text{test}}$
- Validation  
select with  $E_{\text{val}}(\mathcal{A}_m(\mathcal{D}_{\text{train}}))$ ; returning  $\mathcal{A}_{m^*}(\mathcal{D})$
- Leave-One-Out Cross Validation  
huge computation but almost unbiased
- V-Fold Cross Validation  
reasonable computation and performance

- **next: something 'up my sleeve'**