Generalization

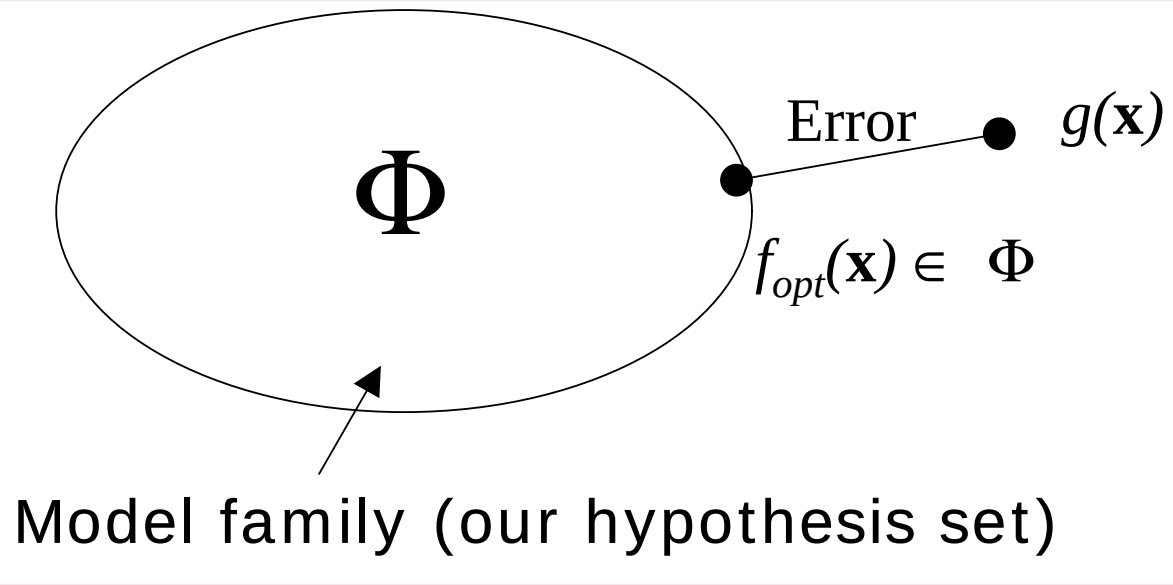
Antanas Verikas
antanas.verikas@hh.se

IDE, Halmstad University

2013

Idealized regression

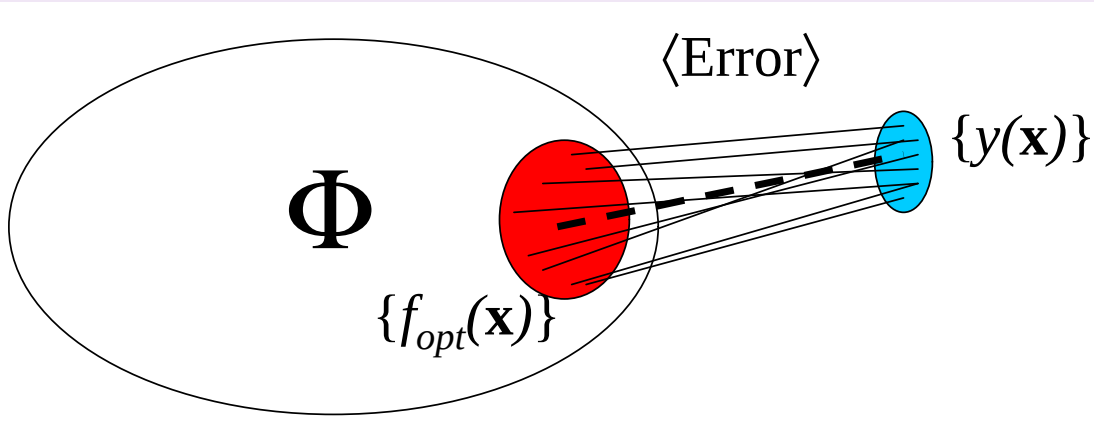
The task: Find an appropriate model family Φ and find $f(\mathbf{x}) \in \Phi$ with a minimum "distance" ("error") to $g(\mathbf{x})$ (true function)



where f is an estimate of g.

The real regression

Find an appropriate model family Φ and minimize the **expected** distance to $y(\mathbf{x})$ ("generalization error")



Data is never noise free, and never available in infinite amounts, thus we get variation in data and model. The generalization error is a function of both the training data and **the hypothesis selection method**.

Well-posed learning problems

- The model can learn the function (i.e. there is no model bias).
- The solution is unique (no local minima)
- The solution is stable under small perturbations of the training set (i.e. the model variance is small)

Unfortunately, this is seldom the case. However, we could aim at making our problems as well-posed as possible.

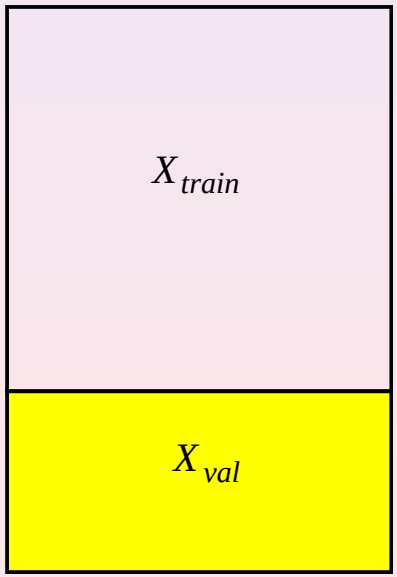
Estimating the generalization error

The generalization error is the average error for unseen (test) data.

- Cross-validation (CV)
- K-fold cross validation (K-CV)
- Leave-one-out estimate (LOO), (LOO equals K-CV when K = number of observations)

Cross-validation

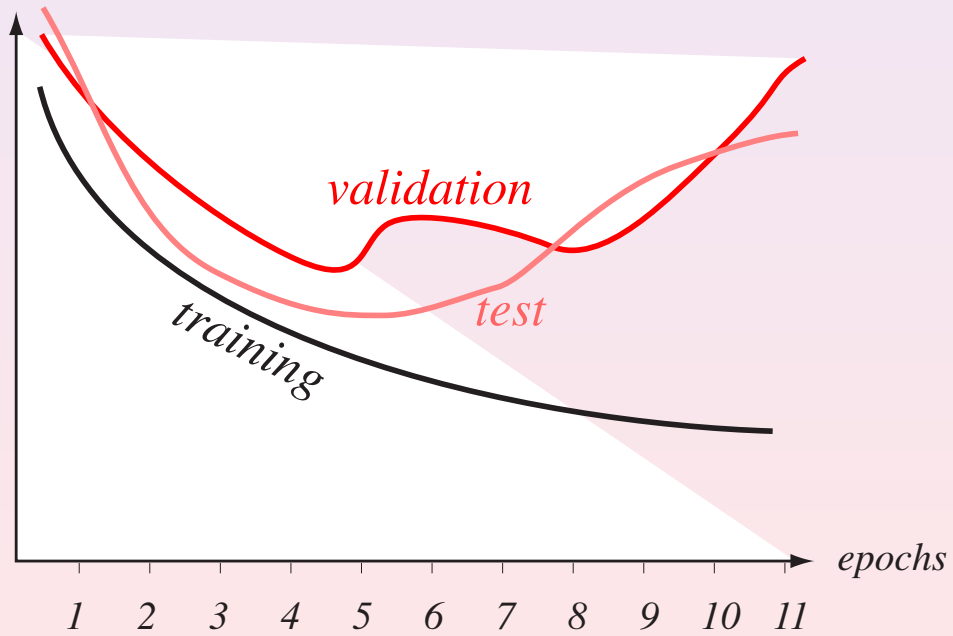
Use a "validation set". Split your data set into two parts: one X_{train} for training your model and the other one X_{val} for validating your model. The error on the validation data is called "validation error" (E_{val})



$$E_{gen} \approx E_{val}$$

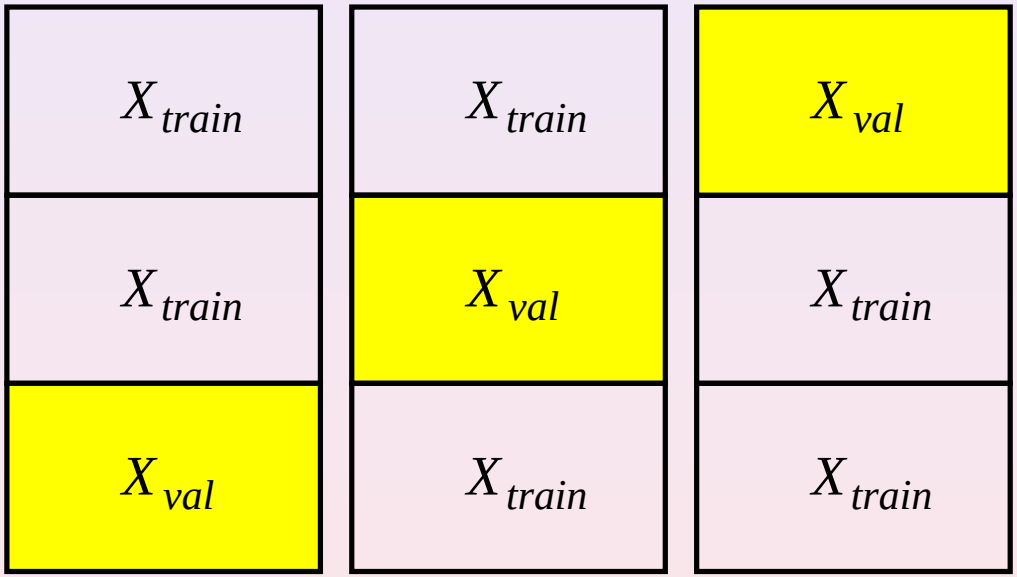
Cross-validation during training

The average error, as a function of the amount of training, indicated by the number of epochs. The validation and the test or generalization error are virtually always higher than the training error.



K-fold Cross-validation

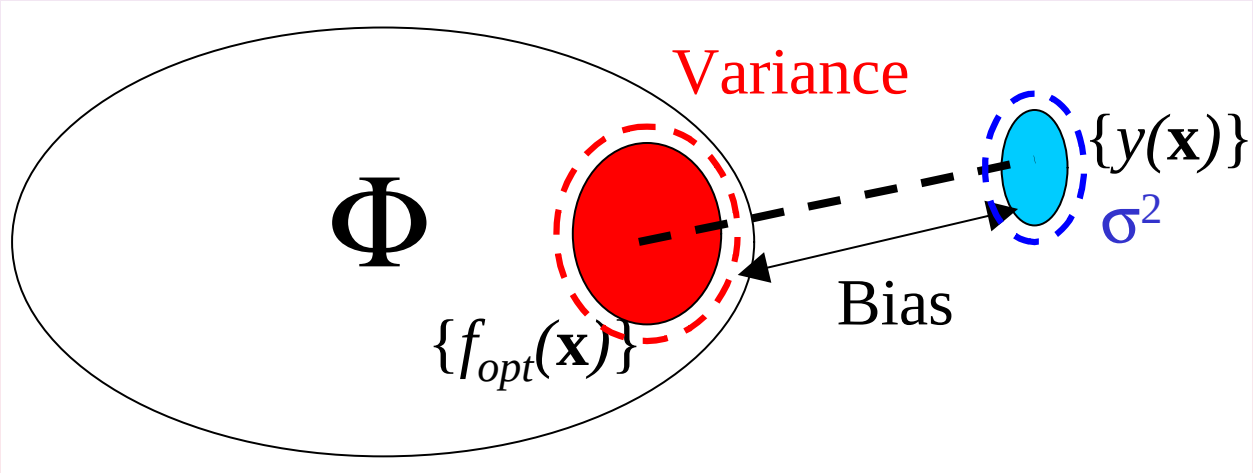
More accurate estimate than using only one validation set.



$$E_{gen} \approx \langle E_{val} \rangle = \frac{1}{K} \sum_{k=1}^{K} E_{val}(k)$$

Model "bias" & model "variance"

$$\langle E_{gen} \rangle = (\texttt{Bias})^2 + (\texttt{Variance}) + \sigma_\varepsilon^2 \qquad (1)$$



Model selection

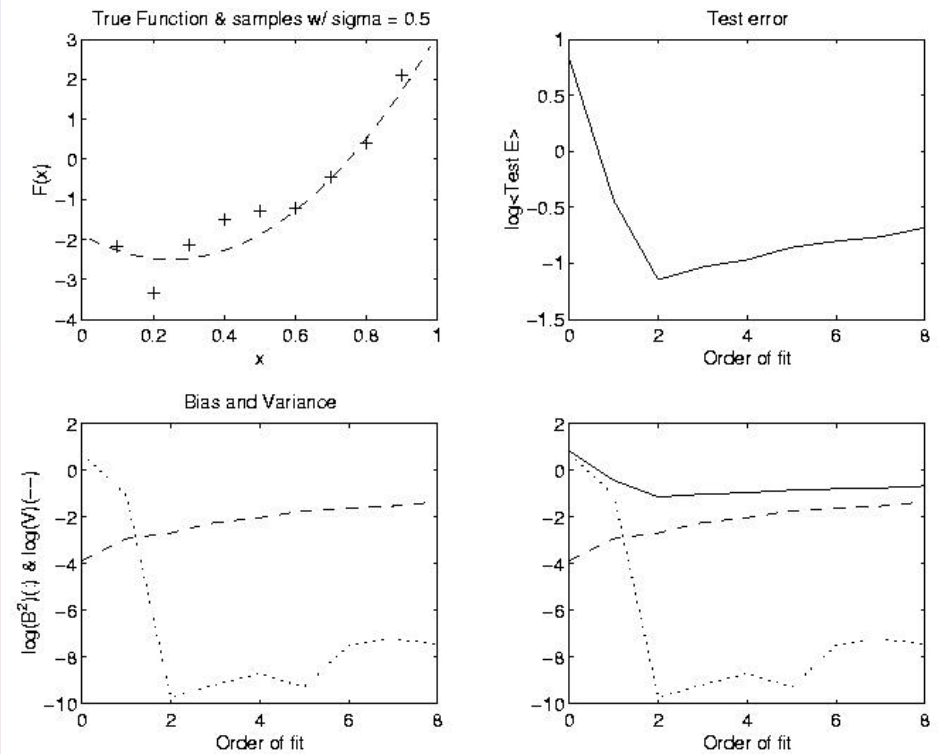
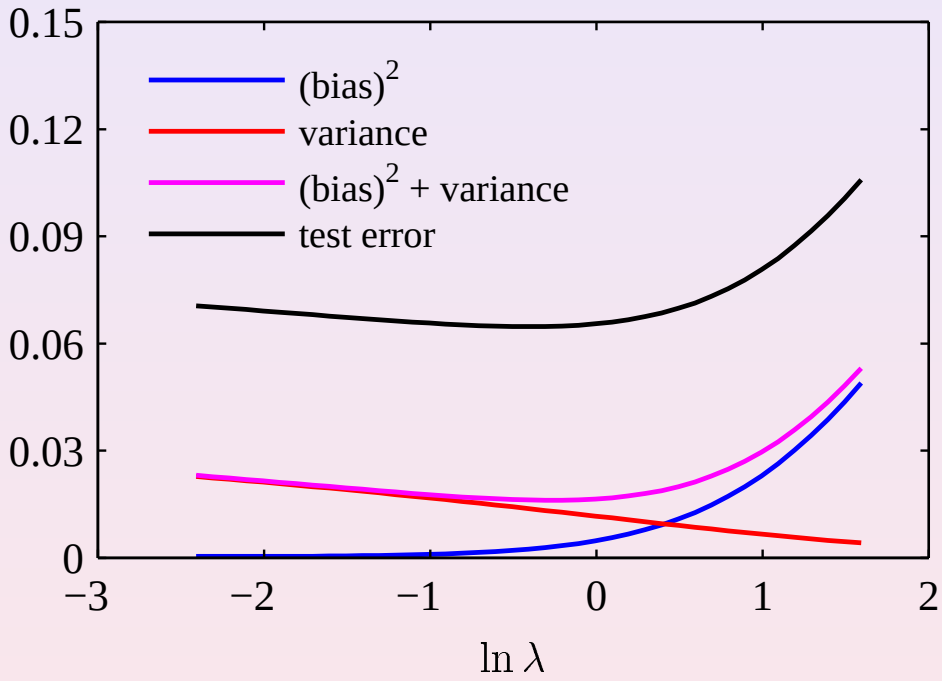


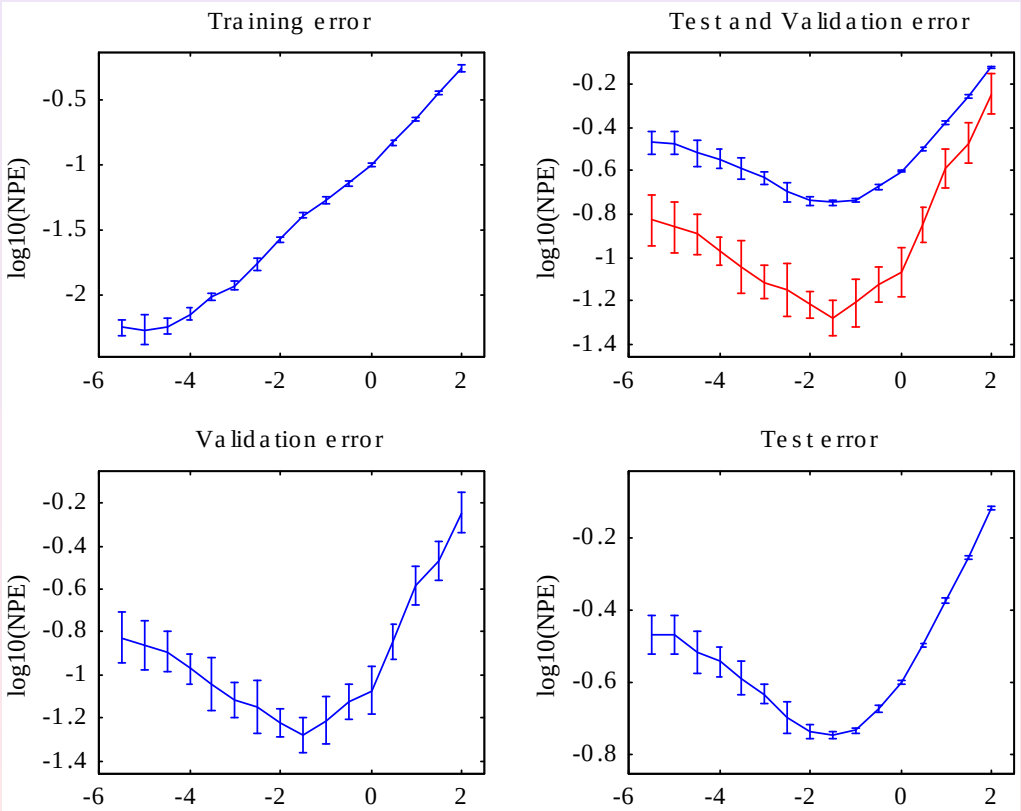
Figure: Model with the lowest generalization error is a bias versus variance trade-off.

Bias/Variance trade-off



λ reflects model complexity. Also shown is the average test set error. Increased model complexity $\Rightarrow$ Increased model variance. Decreased model complexity $\Rightarrow$ Increased model bias.

Selecting model with K-CV



← Model complexity

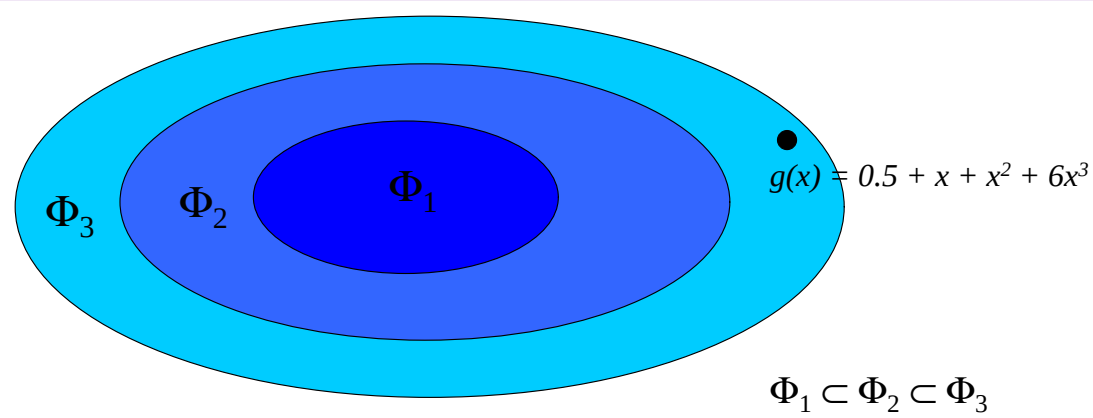
Lines represent averages over the cross-validation sets.

$$\langle \log E \rangle = \frac{1}{K} \sum_{k=1}^{K} \log E_k$$

Error bars are the 95% significance limits for the averages.

Model complexity?

- A "complex" model family Φ contains many models.



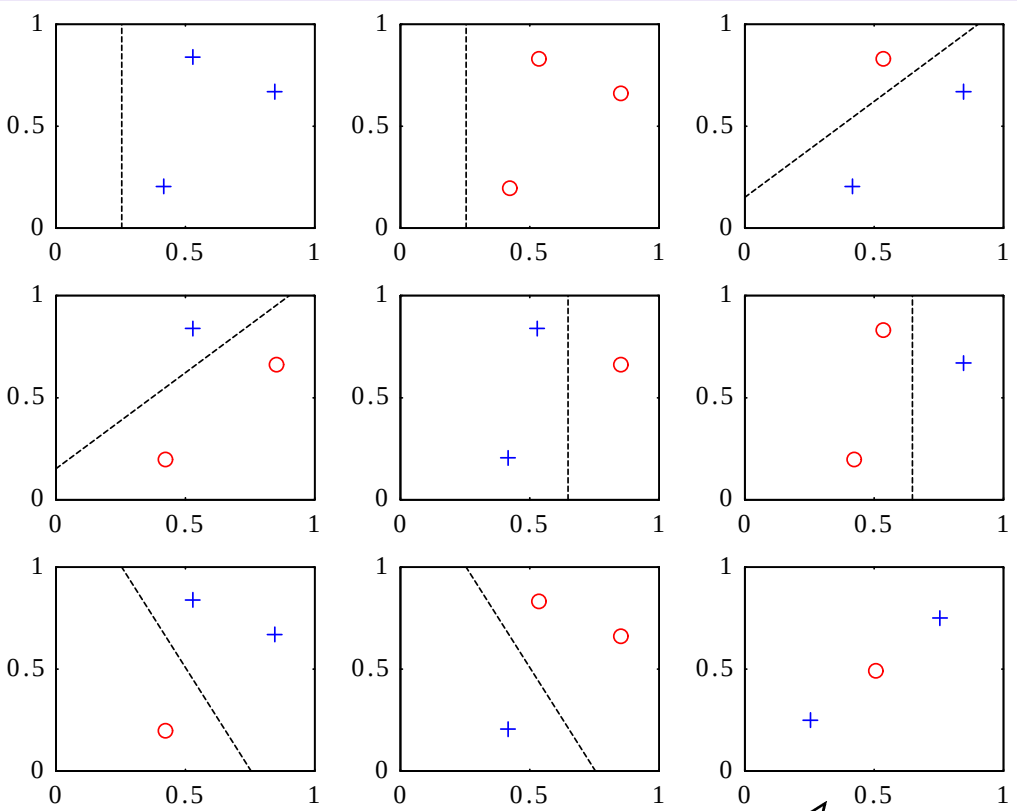
$$\begin{aligned} \text{Linear} \quad & \Phi_1 = \{a + bx\} \\ \text{Quadratic} \quad & \Phi_2 = \{a + bx + cx^2\} \\ \text{Cubic} \quad & \Phi_3 = \{a + bx + cx^2 + dx^3\} \end{aligned}$$

- Measured by the Vapnik-Chervonenkis (VC) dimension.

The VC-dimension

- The VC-dimension of a model family Φ is the size of the largest data set that can be **shattered** by the family Φ.
- A data set is shattered by a model family Φ if all dichotomies of the data set can be realized by functions $f \in \Phi$.

Example: Linear classifiers

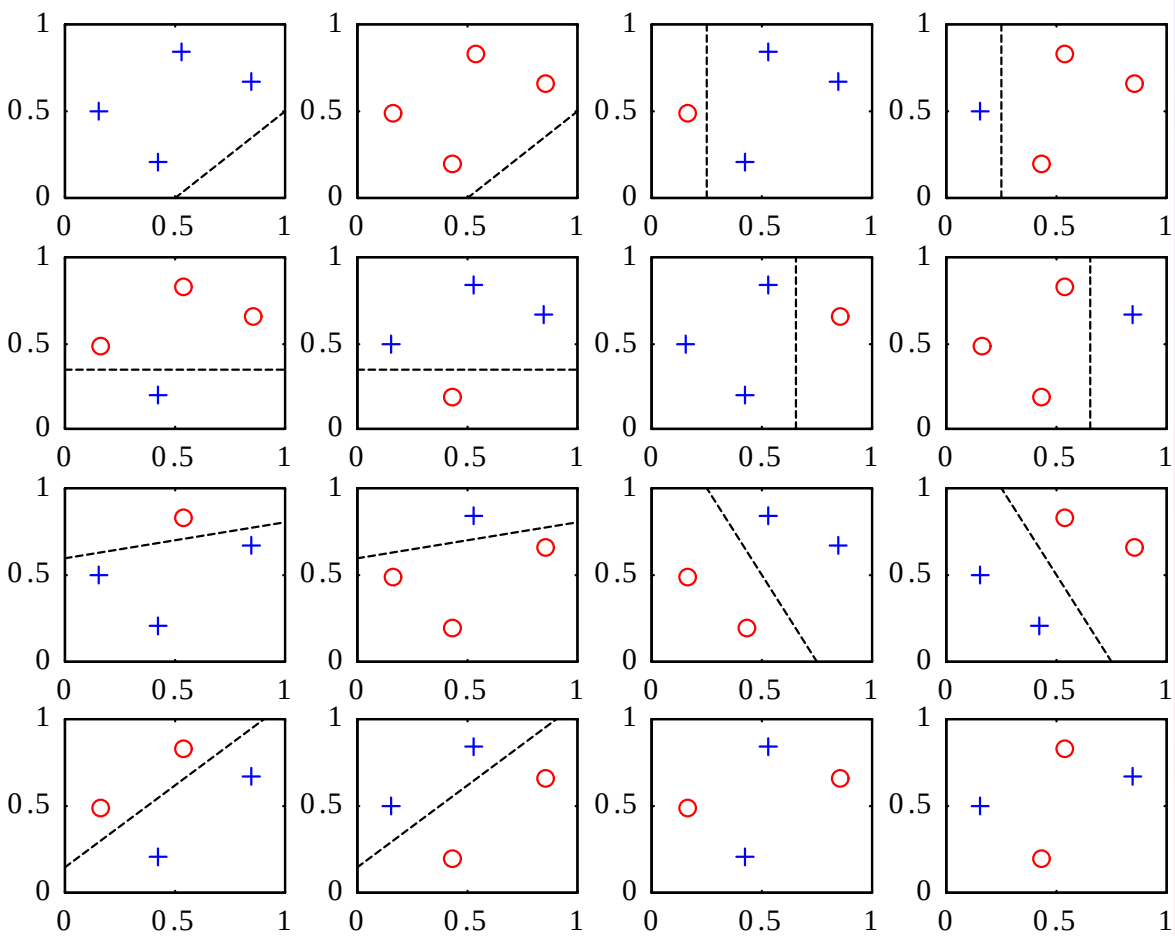


There are 8 possible dichotomies of a set of 3 points in 2 dimensions. They can all be realized with a line.

$\Rightarrow$

$\Phi = \{\text{Lin. Classifiers}\}$ shatters a set of 3 points in 2 dimensions.

Example: Linear classifiers



A set of 4 points in 2 dimensions is not shattered by $\Phi = \{\text{Lin. Classifiers}\}$. However, 87.5% of the dichotomies are linearly separable.

VC-dimension of classifiers

- The VC-dimension of a linear classifier is D+1. This means that if $N \leq D + 1$, the problem is trivial for a linear classifier.
- A classifier with many parameters has a high capacity (high VC-dimension).
- VC-dim of an MLP is proportional to number of weights.
- To ensure good generalization, keep $N \gg$ `VC-dim`, otherwise the model will over-fit the data.

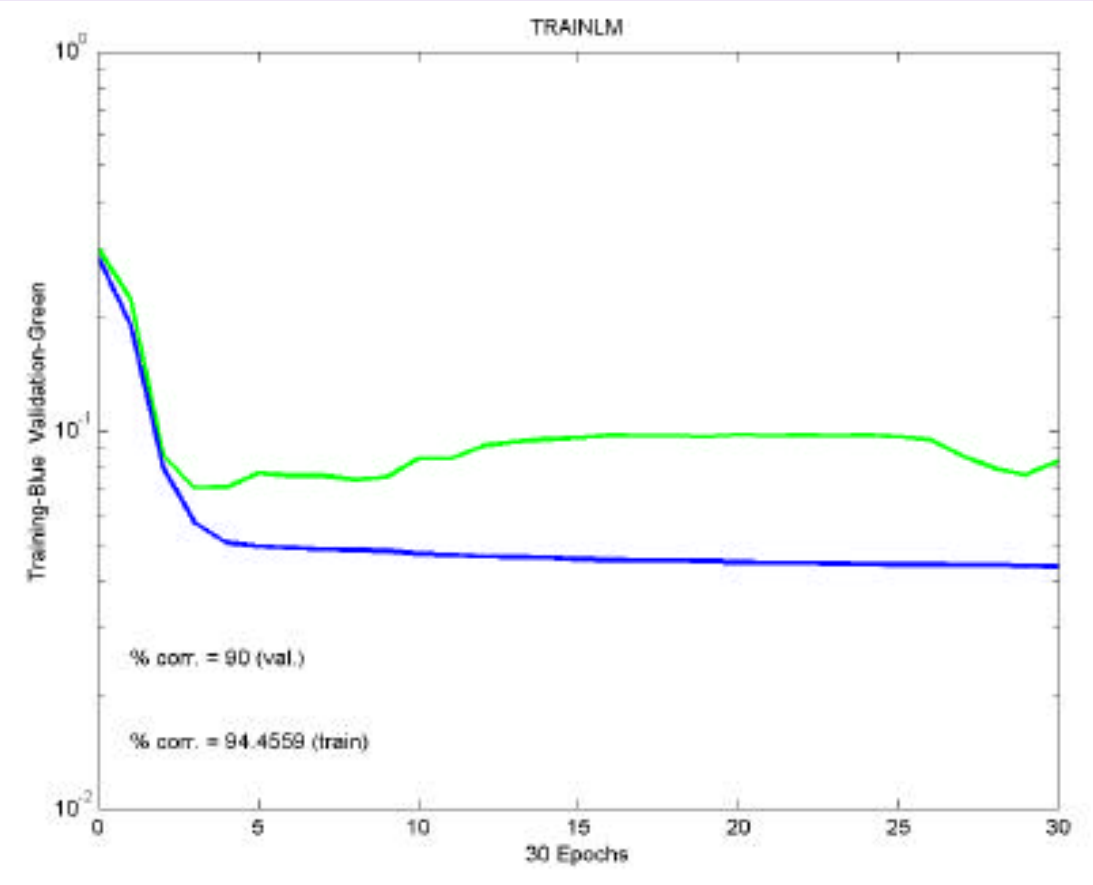
How to control over-fitting?

Controlling over-fitting = controlling the bias/variance trade-off.

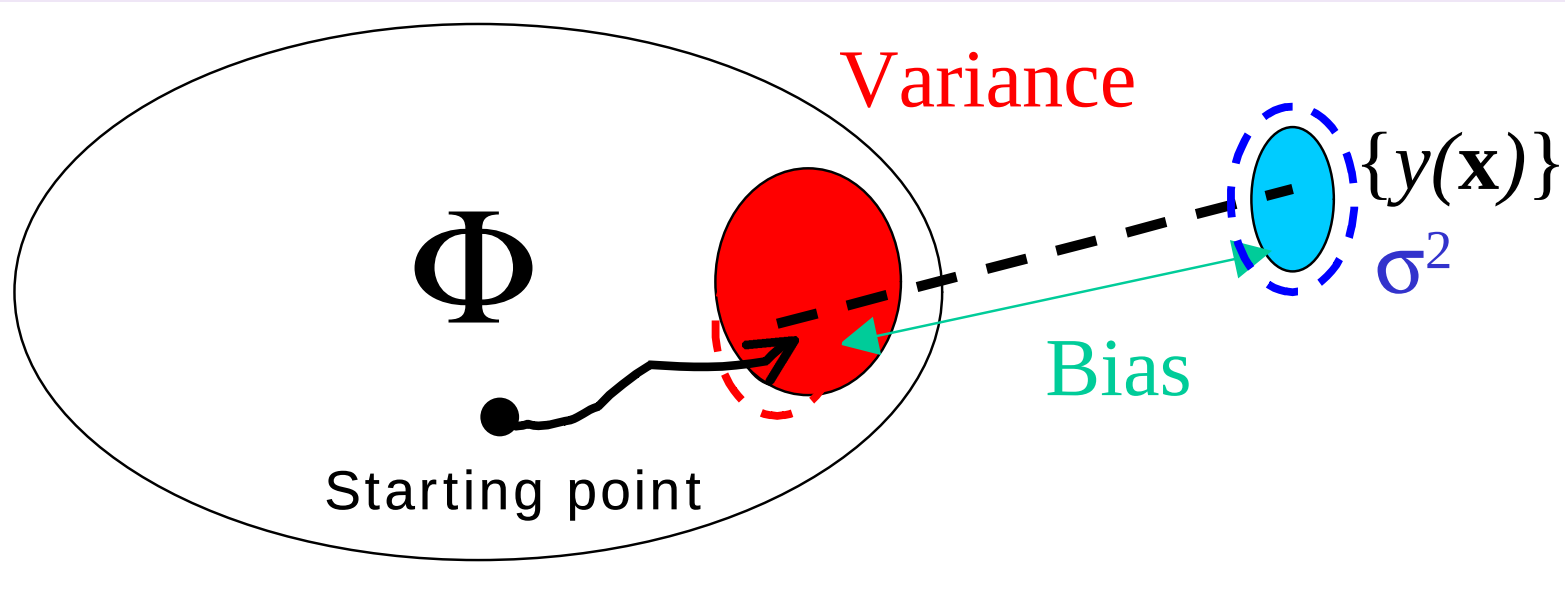
- Early stopping
- Regularization.
- Committees.
- Pruning & growing.
- Variable selection.

Early stopping (1)

Monitor the validation error and stop when it starts to increase.



Early stopping (2)



Bias increases (a little), but variance decreases.

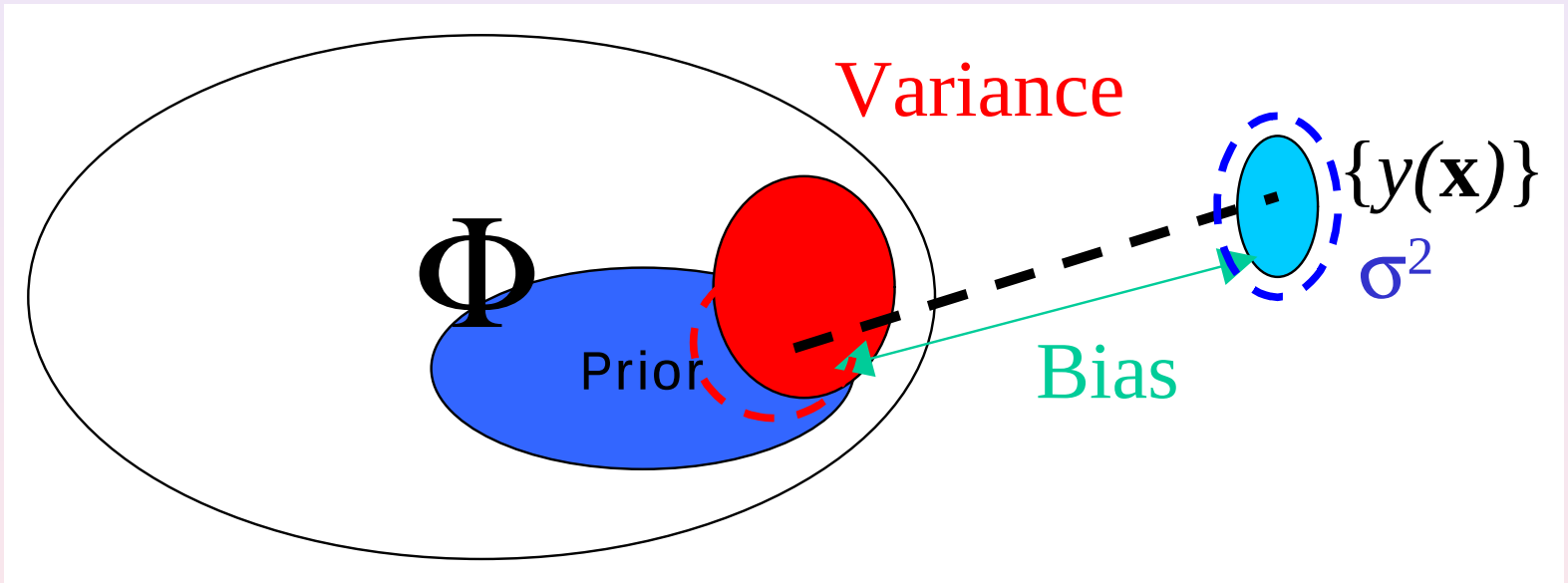
Regularization (1)

Combines usual summed square error with a regularization term (a "prior").

$$E = \sum_{n=1}^{N} [y(n) - \widehat{y}(n)]^2 + \lambda R(\mathbf{w}) \tag{2}$$

The regularization term (negative log of prior) depends only on model parameters $\mathbf{w}$. The regularization parameter λ needs to be selected.

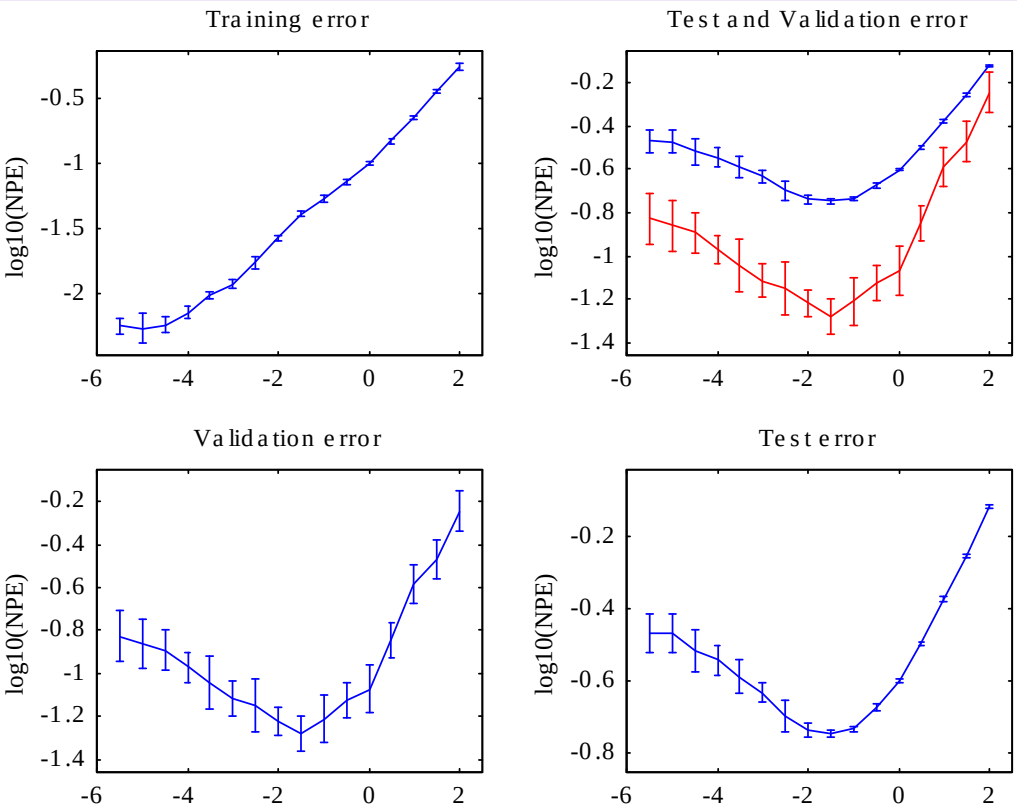
Regularization (2)



Bias increases (a little), but variance decreases.

Regularization with weight decay

Selecting λ for weight decay:



$\leftarrow \log_{10}(\lambda)$

Lines represent averages over the cross-validation sets.

$$\langle \log E \rangle = \frac{1}{K} \sum_{k=1}^{K} \log E_k$$

Error bars are the 95% significance limits for the averages.

Committees (1)

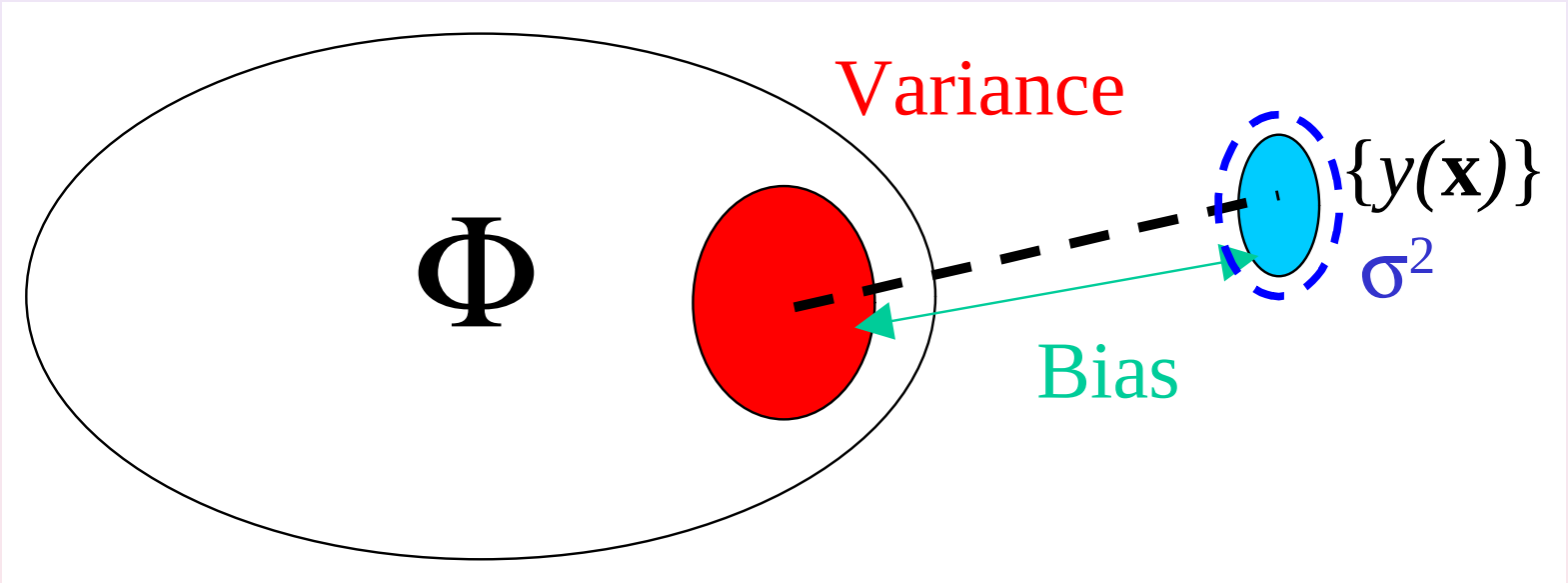
- Averaging ("flat") committee. Always better than a randomly selected member.

$$\widehat{y}_{Aver}(n) = \frac{1}{K}\sum_{k=1}^{K}\widehat{y}_k(n)$$

- Weighted averaging. Always better than the best member - if MSE values are accurate and uncorrelated!

$$\widehat{y}_{Weight}(n) = \frac{\sum_{k=1}^{K}\widehat{y}_k(n)/\mathtt{MSE}_k}{\sum_{k=1}^{K}1/\mathtt{MSE}_k}$$

Committees (2)



Bias $\approx$ unchanged, but variance decreases.

Growing & pruning

Growing: Start with small model, add parts until no improvement is seen.

Pruning: Start with large model, cut away parts that do no good.

Variable selection

More variables imply larger variance. For linear regression models:

$$\langle E_{\texttt{Test}} \rangle = \langle E_{\texttt{Train}} \rangle + \frac{\sigma_\varepsilon^2 (D+1)}{N}$$

$\Rightarrow$ A penalty is payed for each input.
Examples of selection techniques:

- Start with all variables and remove one by one until the generalization error starts to increase (backward selection).
- Start with one variable and add new variables until generalization error ceases decreasing (forward selection).