Lecture 1: Learning without Over-learning

Isabelle Guyon

isabelle@clopinet.com

Machine Learning

- **Learning machines include**:
  - Linear discriminant (including Naïve Bayes)
  - Kernel methods
  - Neural networks
  - Decision trees
- **Learning is tuning:**
  - Parameters (weights **w** or α, threshold b)
  - Hyperparameters (basis functions, kernels, number of units)

How to Train?

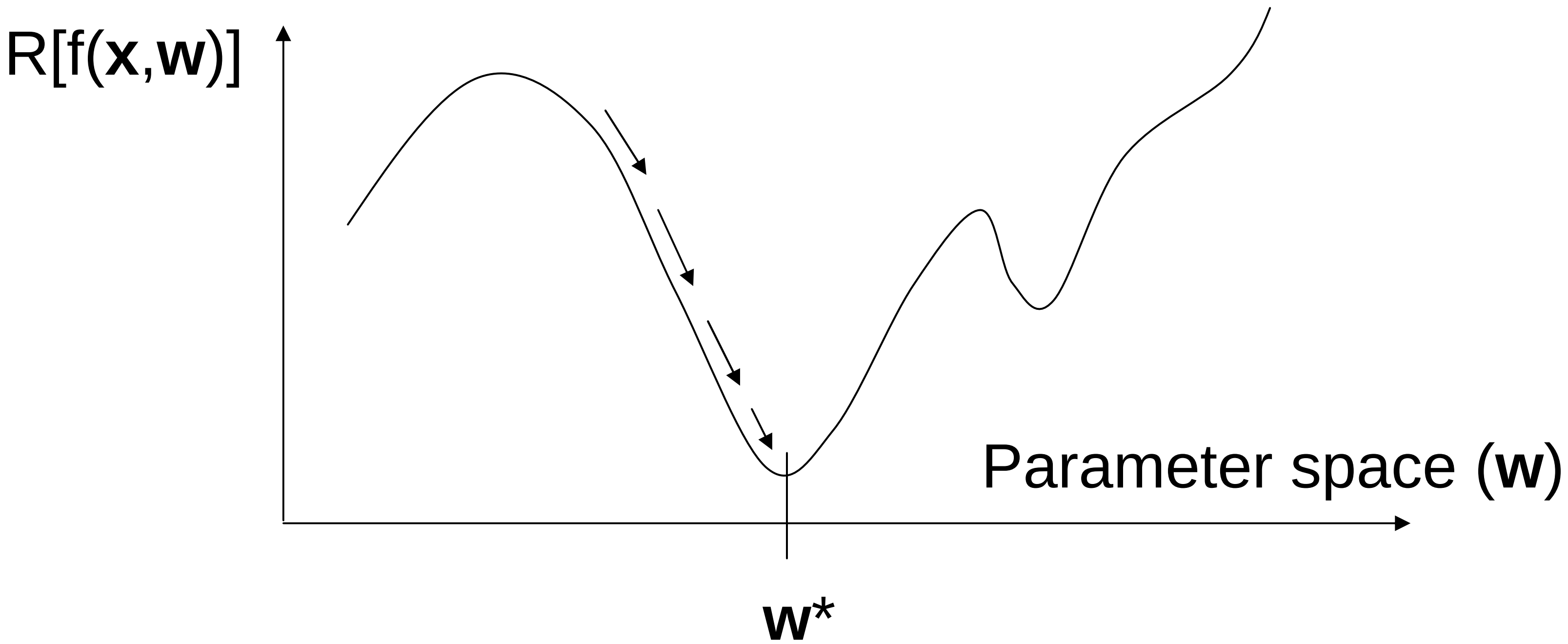
- Define a risk functional $R[f(\mathbf{x},\mathbf{w})]$
- Find a method to optimize it, typically "gradient descent"

$$w_j \leftarrow w_j - \eta\ \partial R/\partial w_j$$

or any optimization method (mathematical programming, simulated annealing, genetic algorithms, etc.)

What is a Risk Functional?

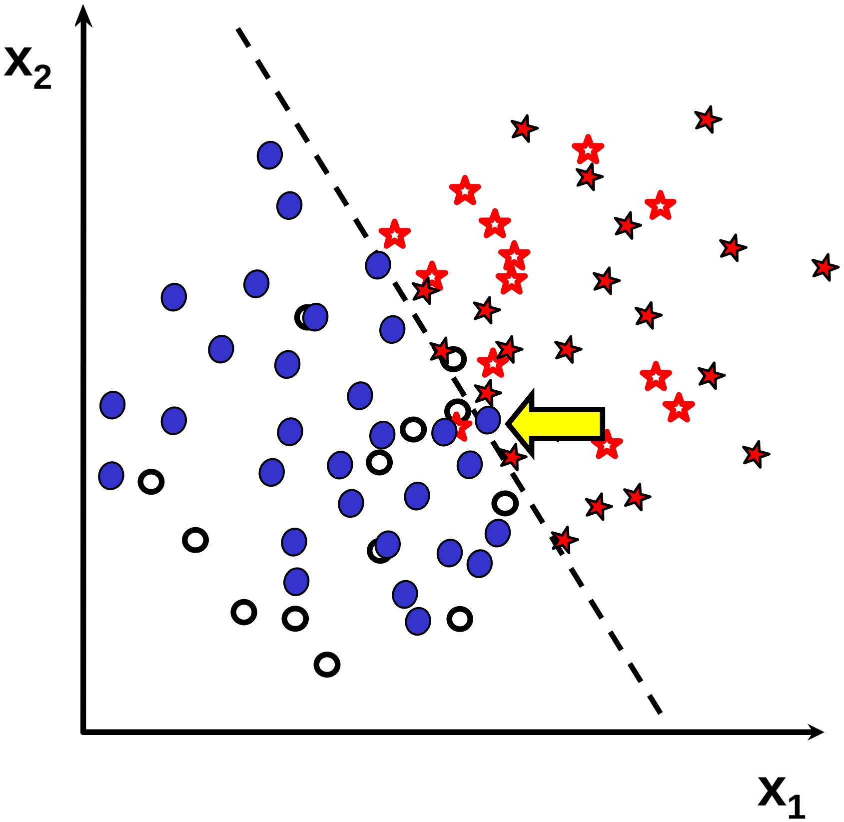
- A function of the parameters of the learning machine, assessing how much it is expected to fail on a given task.

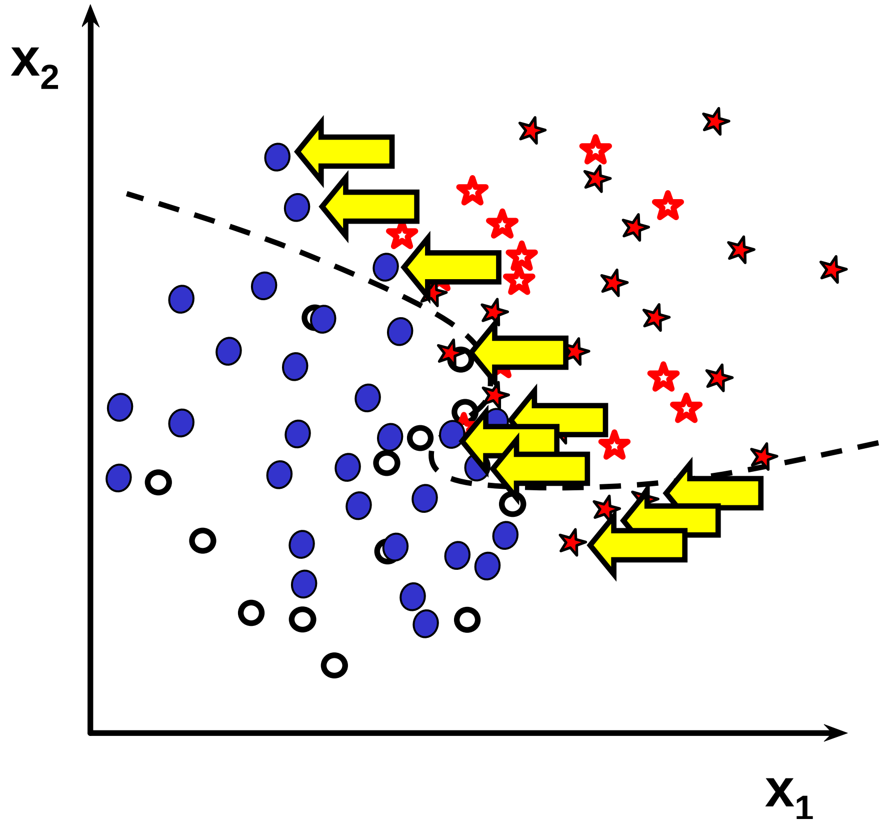


Example Risk Functionals

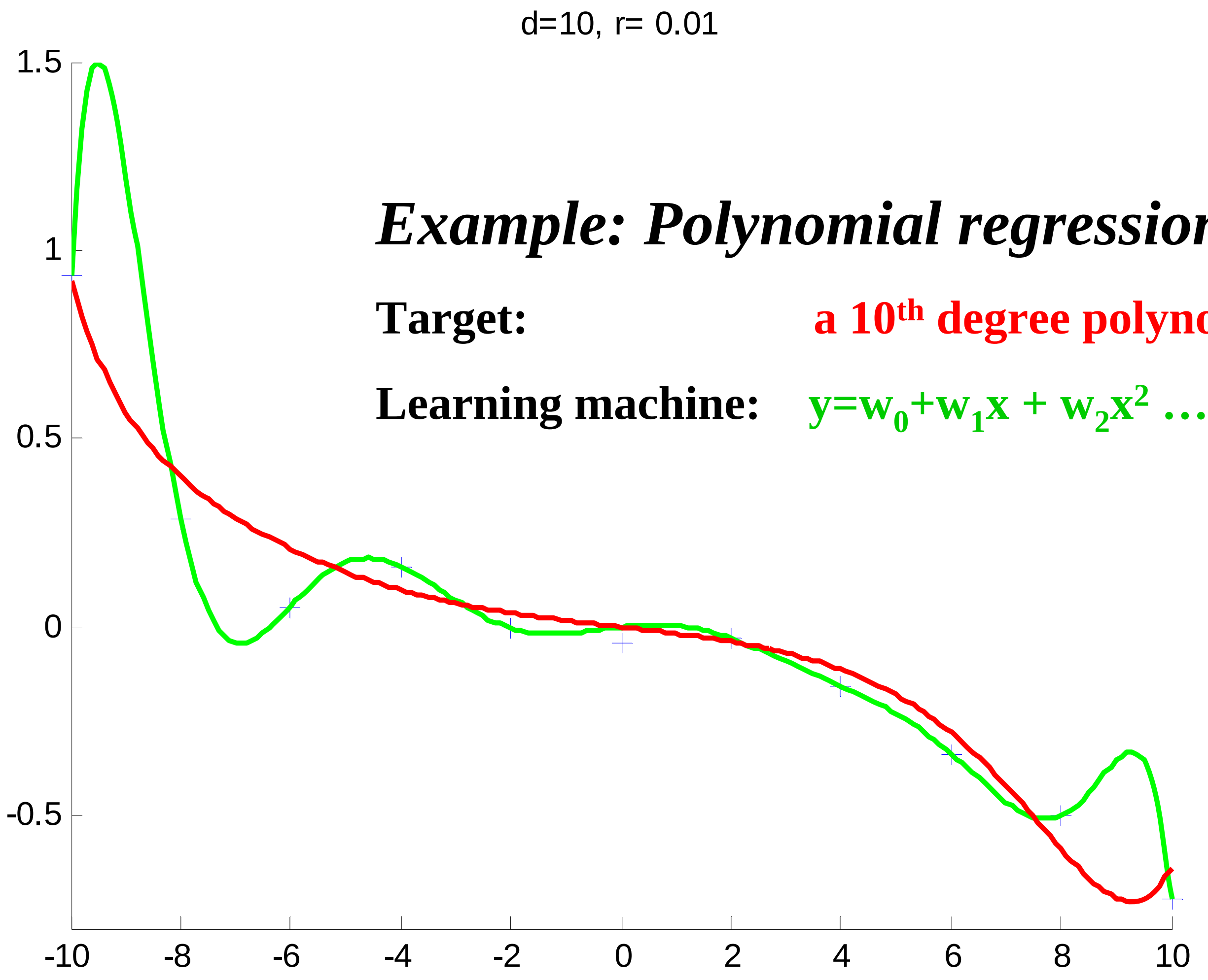
- Classification:
  - –the error rate
- Regression:
  - –the mean square error

Fit / Robustness Tradeoff





Overfitting



Example: Polynomial regression

Target: **a 10^{th} degree polynomial + noise**

Learning machine: $y=w_0+w_1x + w_2x^2 \ldots + w_{10}x^{10}$

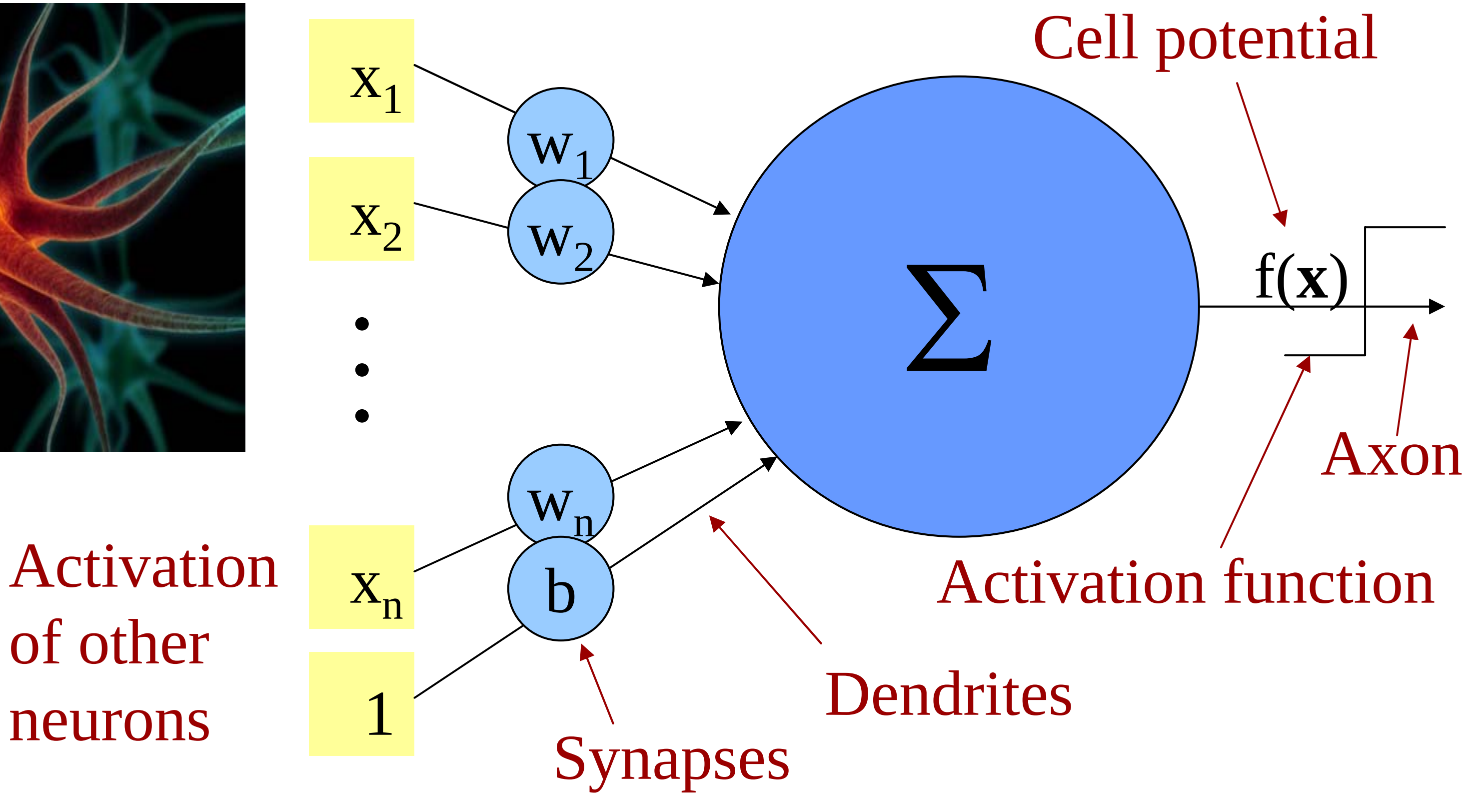
Ockham's Razor

- Principle proposed by William of Ockham in the fourteenth century: "**Pluralitas non est ponenda sine neccesitate**".
- Of two theories providing similarly good predictions, prefer *the simplest one*.
- Shave off unnecessary parameters of your models.

The Power of Amnesia

- The human **brain** is made out of billions of cells or Neurons, which are highly interconnected by synapses.
- Exposure to enriched environments with extra sensory and social stimulation enhances the **connectivity** of the synapses, but children and adolescents can lose them up to 20 million per day.

Artificial Neurons

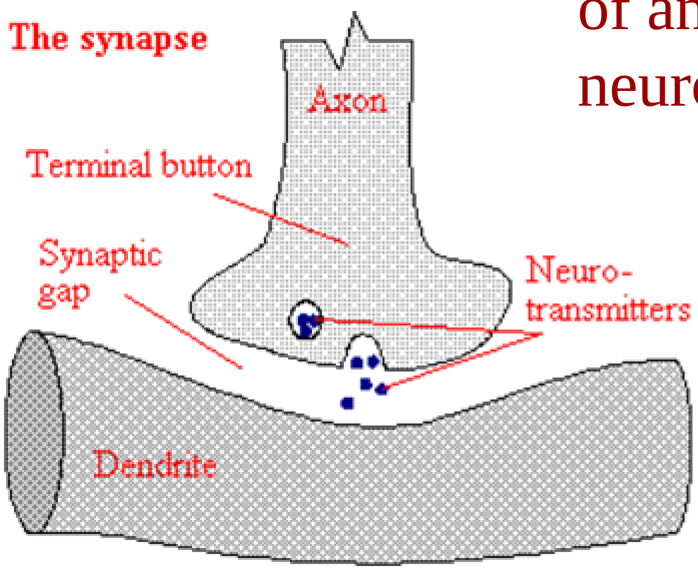


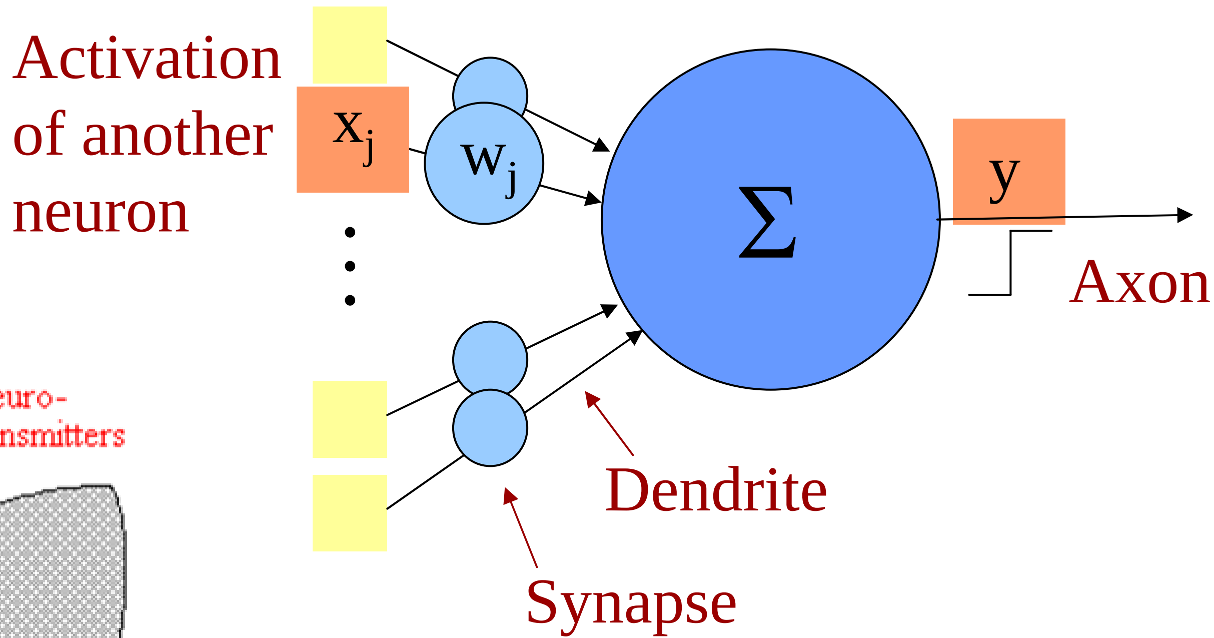
McCulloch and Pitts, 1943

$$\boxed{f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x} + b}$$

Hebb's Rule

$$w_j \leftarrow w_j + y_i\, x_{ij}$$





Link to "Naïve Bayes"

Weight Decay

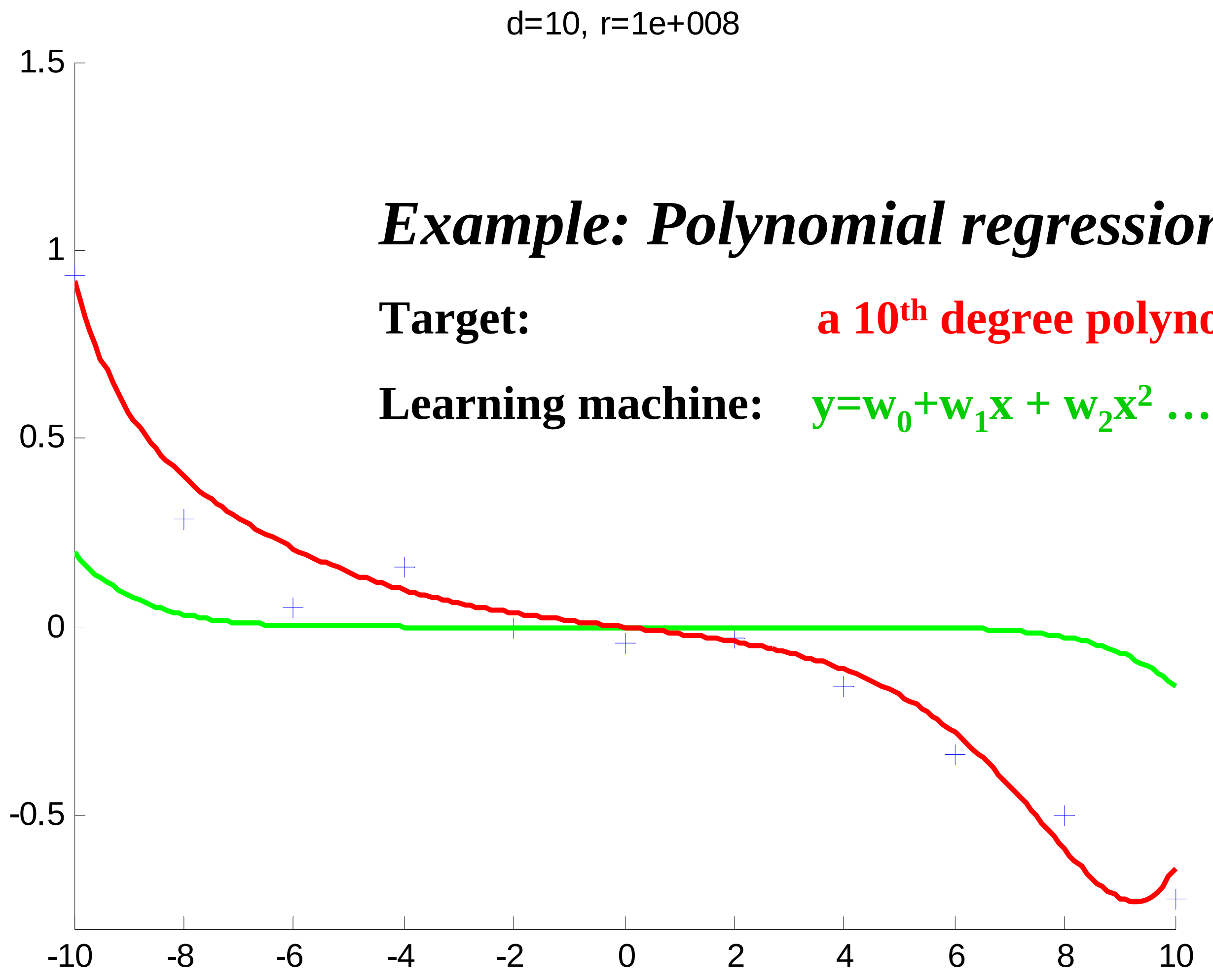
$w_j \leftarrow w_j + y_i \, x_{ij}$ Hebb's rule

$$\boxed{w_j \leftarrow (1-\gamma)\, w_j + y_i \, x_{ij}}$$

Weigh decay

$\gamma \in [0, 1]$, decay parameter

Overfitting Avoidance



Example: Polynomial regression

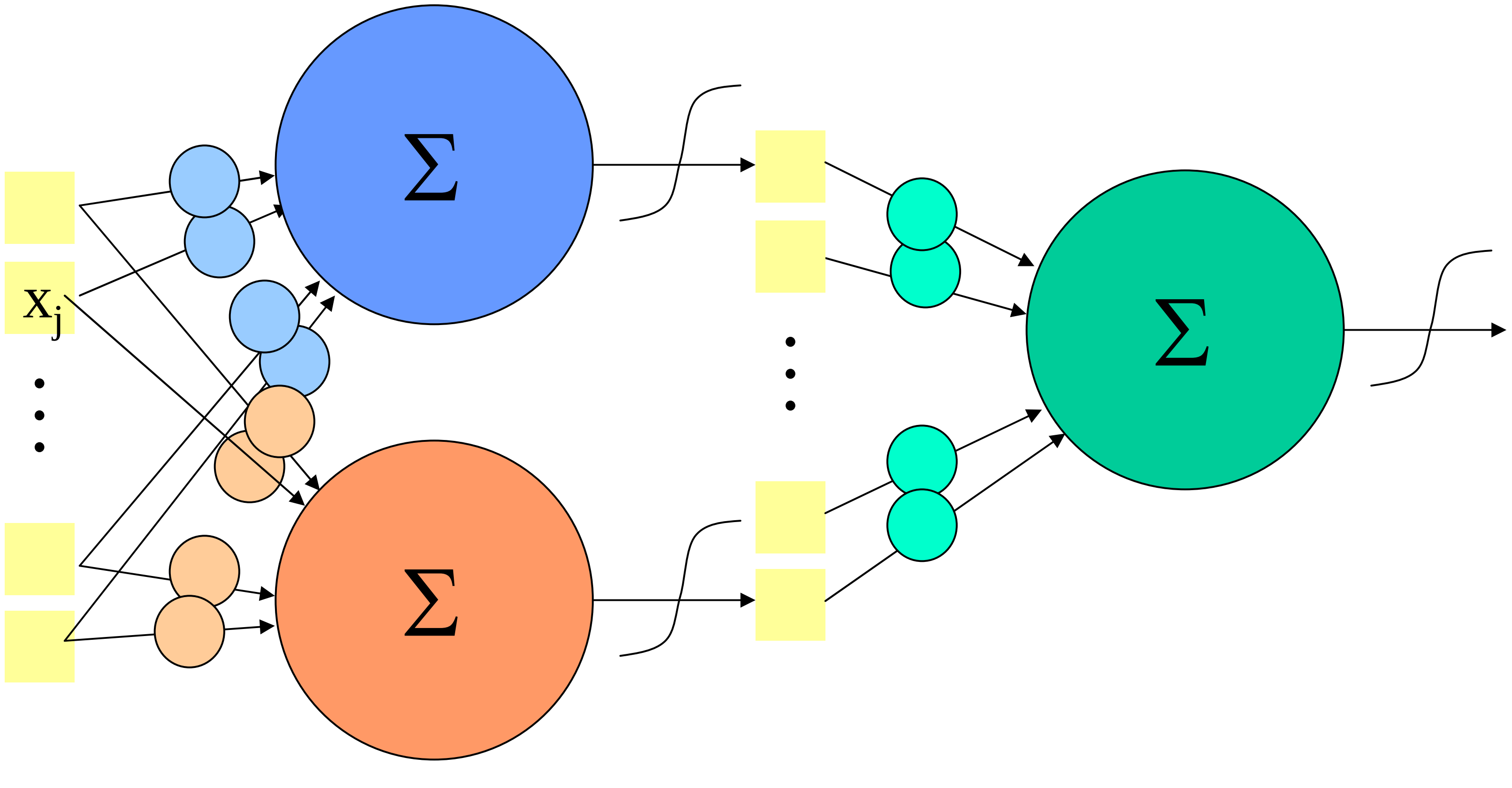
Target: **a 10^{th} degree polynomial + noise**

Learning machine: $\mathbf{y=w_0+w_1x + w_2x^2 \ldots + w_{10}x^{10}}$

Weight Decay for MLP

Replace: $w_j \leftarrow w_j + \text{back\_prop}(j)$

by: $w_j \leftarrow (1-\gamma)\, w_j + \text{back\_prop}(j)$



Theoretical Foundations

- Structural Risk Minimization
- Bayesian priors
- Minimum Description Length
- Bayes/variance tradeoff

Risk Minimization

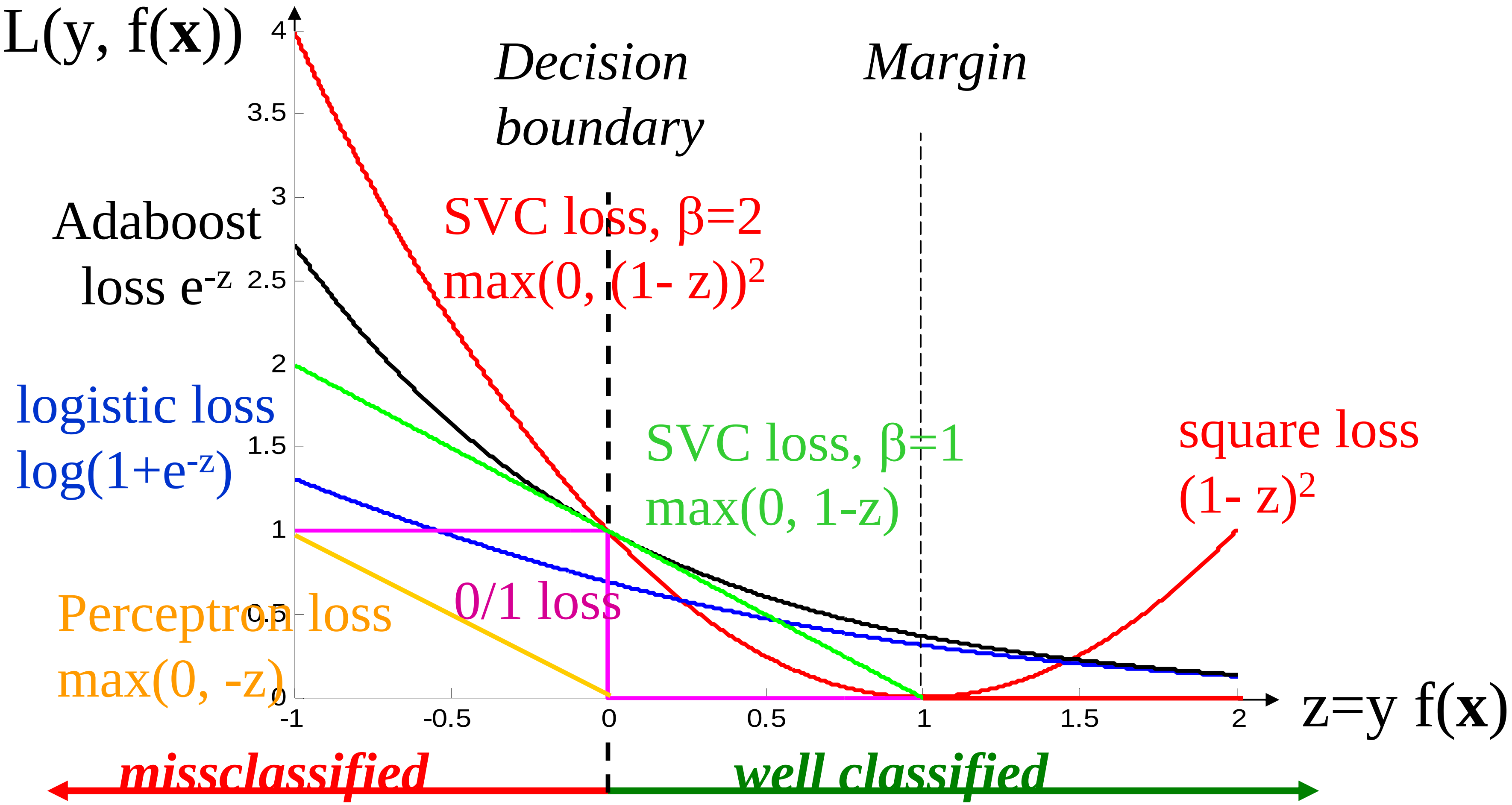
- **Learning problem:** find the best function $f(\mathbf{x};\ \mathbf{w})$ minimizing a risk functional

$$R[f] = \int \underbrace{L(f(\mathbf{x};\ \mathbf{w}),\ y)}_{\text{loss function}}\ d\underbrace{P(\mathbf{x},\ y)}_{\text{unknown distribution}}$$

- **Examples are given**:

$(\mathbf{x}_1,\ y_1),\ (\mathbf{x}_2,\ y_2),\ \ldots\ (\mathbf{x}_m,\ y_m)$

Loss Functions



Approximations of R[f]

- **Empirical risk**: $R_{train}[f] = (1/n)\ \Sigma_{\mathbf{i=1:m}}\ L(f(\mathbf{x_i};\ \mathbf{w}),\ y_\mathbf{i})$

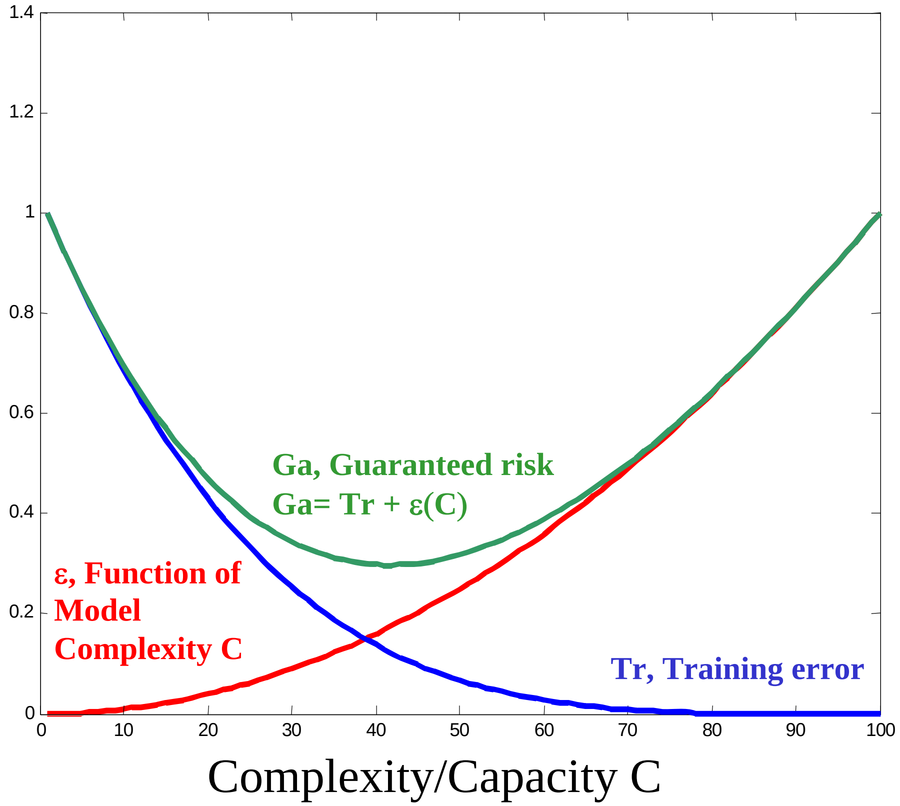
  - 0/1 loss $\mathbf{1}(F(\mathbf{x}_i)\neq y_i)$: $R_{train}[f]$ = error rate
  - square loss $(f(\mathbf{x}_i)-y_i)^2$: $R_{train}[f]$ = mean square error

- **Guaranteed risk**:

With *high* probability $(1-\delta)$, $R[f] \leq R_{gua}[f]$

$$\boxed{R_{gua}[f] = R_{train}[f] + \varepsilon(\delta, C)}$$

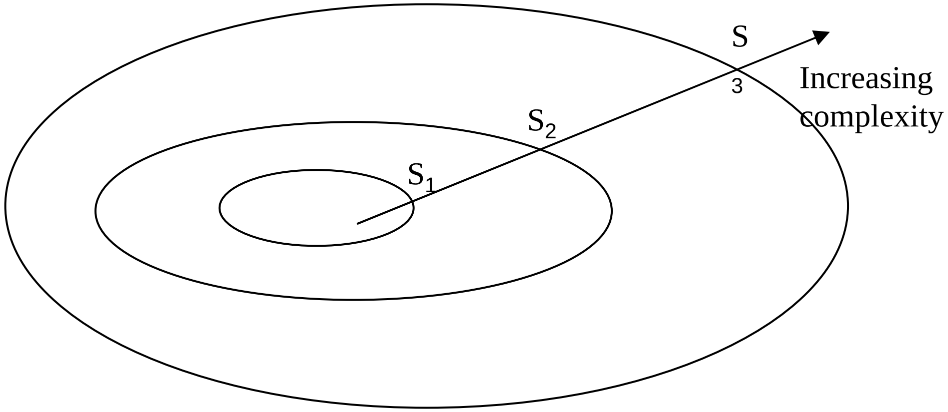
Structural Risk Minimization



Vapnik, 1974

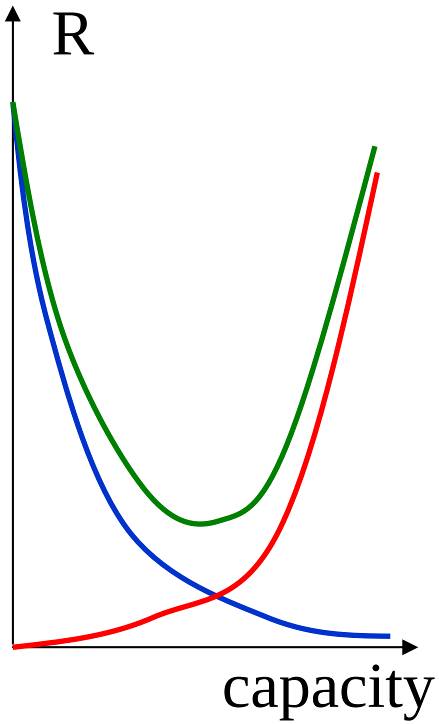
Nested subsets of models, increasing complexity/capacity:

$S_1 \subset S_2 \subset \ldots S_N$



SRM Example

$$S_1 \subset S_2 \subset \ldots S_N$$



- **Rank with** $||\mathbf{w}||^2 = \Sigma_i w_i^2$

  $S_k = \{ \mathbf{w} \mid ||\mathbf{w}||^2 < \omega_k^2 \}, \omega_1 < \omega_2 < \ldots < \omega_k$

- **Minimization under constraint:**

  $\min R_{train}[f] \qquad \text{s.t.} \ ||\mathbf{w}||^2 < \omega_k^2$

- **Lagrangian:**

$$R_{reg}[f,\gamma] = R_{train}[f] + \gamma \, ||\mathbf{w}||^2$$

Gradient Descent

$R_{reg}[f] = R_{emp}[f] + \lambda \, ||\mathbf{w}||^2$ **SRM/regularization**

$w_j \leftarrow w_j - \eta \, \partial R_{reg}/\partial w_j$

$w_j \leftarrow w_j - \eta \, R_{emp}/\partial w_j - 2 \, \eta \, \lambda \, w_j$

$w_j \leftarrow (1- \gamma) \, w_j - \eta \, R_{emp}/\partial w_j$ **Weight decay**

Multiple Structures

- **Shrinkage (weight decay, ridge regression, SVM):**

  $S_k = \{ \mathbf{w} \mid \|\mathbf{w}\|_2 < \omega_k \}$, $\omega_1 < \omega_2 < \ldots < \omega_k$

  $\gamma_1 > \gamma_2 > \gamma_3 > \ldots > \gamma_k$ *(γ is the ridge)*

- **Feature selection:**

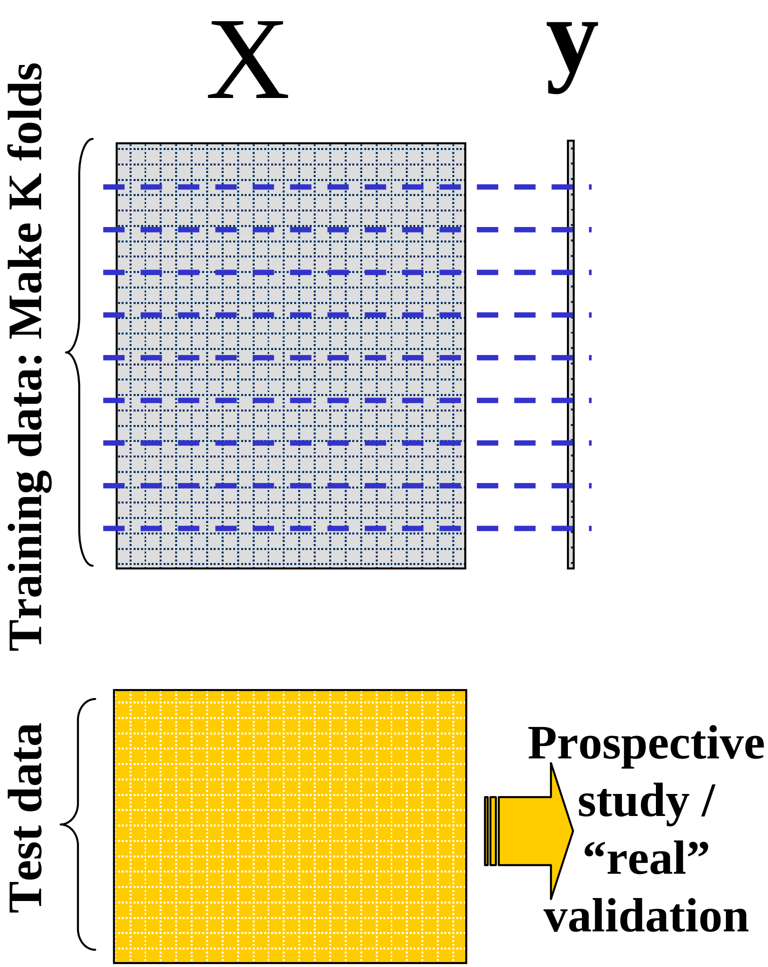
  $S_k = \{ \mathbf{w} \mid \|\mathbf{w}\|_0 < \sigma_k \}$,

  $\sigma_1 < \sigma_2 < \ldots < \sigma_k$ *(σ is the number of features)*

- **Data compression:**

  $\kappa_1 < \kappa_2 < \ldots < \kappa_k$ *(κ may be the number of clusters)*

Hyper-parameter Selection



- **Learning = adjusting:**

  parameters ($\mathbf{w}$ vector).

  hyper-parameters (γ, σ, κ).

- ***Cross-validation* with K-folds:**

For various values of γ, σ, κ:

- Adjust $\mathbf{w}$ on a fraction (K-1)/K of training examples *e.g.* 9/10th.
- Test on 1/K remaining examples *e.g.* 1/10th.
- Rotate examples and average test results (CV error).
- Select γ, σ, κ to minimize CV error.
- Re-compute $\mathbf{w}$ on **all** training examples using optimal γ, σ, κ.

Bayesian MAP $\simeq$ SRM

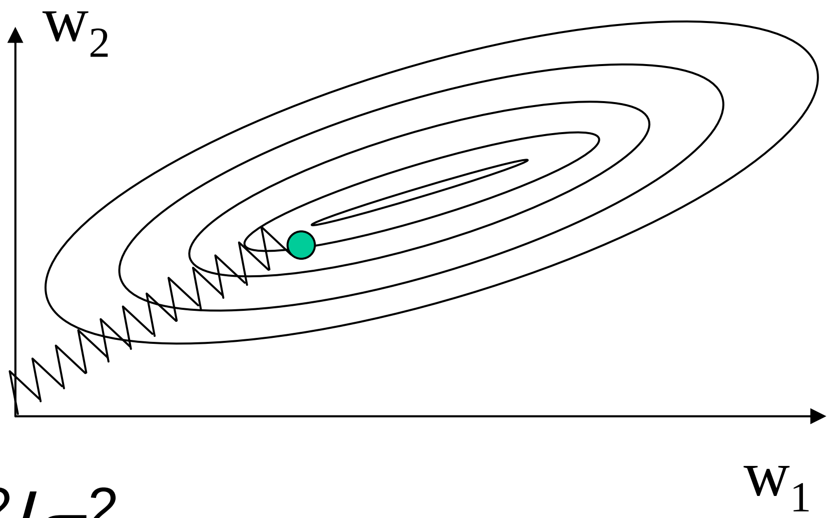
- Maximum A Posteriori (MAP):

$$
\begin{aligned}
f &= \text{argmax } P(f|D) \\
&= \text{argmax } P(D|f)\,P(f) \\
&= \text{argmin } \underbrace{-\log P(D|f)}_{\substack{\text{Negative log likelihood} \\ = \text{Empirical risk } R_{emp}[f]}} \quad \underbrace{-\log P(f)}_{\substack{\text{Negative log prior} \\ = \text{Regularizer } \Omega[f]}}
\end{aligned}
$$

- Structural Risk Minimization (SRM):

$$f = \text{argmin } R_{emp}[f] + \Omega[f]$$

Example: Gaussian Prior



- Linear model:

  $f(\mathbf{x}) = \mathbf{w}.\mathbf{x}$

- Gaussian prior:

  $P(f) = \exp -||\mathbf{w}||^2/\sigma^2$

- Regularizer:

  $\Omega[f] = -\log P(f) = \lambda\ ||\mathbf{w}||^2$

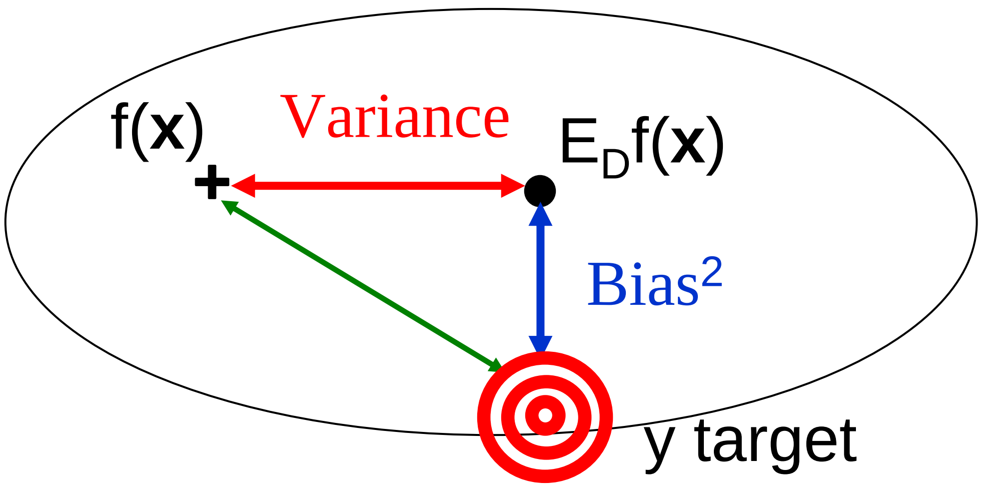
Minimum Description Length

- MDL: minimize the length of the "message".
- Two part code: transmit the model and the residual.
- $f = \text{argmin} \underbrace{-\log_2 P(D|f)}_{\substack{\text{Residual: length of} \\ \text{the shortest code} \\ \text{to encode the data} \\ \text{given the model}}} \underbrace{-\log_2 P(f)}_{\substack{\text{Length of the} \\ \text{shortest code to} \\ \text{encode the model} \\ \text{(model complexity)}}}$

Bias-variance tradeoff

- f trained on a training set D of size m (m fixed)
- For the square loss:

$$\underbrace{E_D[f(\mathbf{x})-y]^2}_{\text{Expected value of the loss over datasets D of the same size}} = \underbrace{[E_D f(\mathbf{x})-y]^2}_{\text{Bias}^2} + \underbrace{E_D[f(\mathbf{x})-E_D f(\mathbf{x})]^2}_{\text{Variance}}$$



The Effect of SRM

Reduces the variance…

…at the expense of introducing some bias.

Ensemble Methods

- Variance can also be reduced with committee machines.
- The committee members “vote” to make the final decision.
- Committee members are built e.g. with data subsamples.
- Each committee member should have a low bias (no use of ridge/weight decay).

Summary

- Weight decay is a powerful means of overfitting avoidance ($||\mathbf{w}||^2$ regularizer).
- It has several theoretical justifications: SRM, Bayesian prior, MDL.
- It controls variance in the learning machine family, but introduces bias.
- Variance can also be controlled with ensemble methods.

Want to Learn More?

- **Statistical Learning Theory,** *V. Vapnik.* Theoretical book. Reference book on generatization, VC dimension, Structural Risk Minimization, SVMs, ISBN : 0471030031.
- **Structural risk minimization for character recognition,** *I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S.A. Solla.*
  In J. E. Moody et al., editor, Advances in Neural Information Processing Systems 4 (NIPS 91), pages 471--479, San Mateo CA, Morgan Kaufmann, 1992. http://clopinet.com/isabelle/Papers/srm.ps.Z
- **Kernel Ridge Regression Tutorial,** *I. Guyon.*
  http://clopinet.com/isabelle/Projects/ETH/KernelRidge.pdf
- **Feature Extraction: Foundations and Applications.** *I. Guyon et al, Eds.* Book for practitioners with datasets of NIPS 2003 challenge, tutorials, best performing methods, Matlab code, teaching material.
  http://clopinet.com/fextract-book