Optimization Methods for Machine Learning (OMML)

Laura Palagi
Department of Computer, Control, and Management Engineering
Antonio Ruberti

SAPIENZA
UNIVERSITÀ DI ROMA

4th lecture
October 2, 2018

Beyond the ERM principle

With probability $1-\eta$ being $\eta \in (0,1)$.

$$R(\propto) \leq R_{emp} + C_{VC}(\eta, l, h\ )$$

- Vapnik Chervonenkis (VC) theory
  - the term $C_{VC}(\eta, l, h\ )$ is called VC confidence depends on
    - l the number of samples in the training set
    - h a parameter
  - the parameter h is called VC dimension

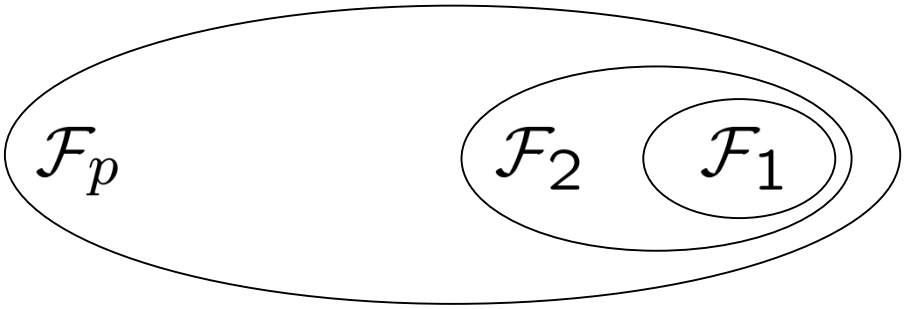
    and it is a measure of complexity and measures the expressive power, richness or flexibility of the class of functions.

Structural Risk Minimization principle

Consider the nested classes of functions

$$h_1 \leq h_2 \leq h_3 \leq \cdots \leq h_p$$

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \cdots \subseteq F_p$$



For each class F_j with VC dimension h_j

- Find the optimal solution

$$\alpha^{j*} = \arg\min_{\alpha} R_{emp}(f_\alpha^j) \qquad f_\alpha^j \in \mathcal{F}^j$$

- Find the value of the upper bound

$$U^{j*} = U(\alpha^{j*}, f_\alpha^j) = R_{emp}(\alpha^{j*}) + C_{VC}(h^j, \ell, \eta)$$

Choose the class of functions which minimizes the upper bound

$$\mathcal{F}^* = \arg\min_{\mathcal{F}^j} U^{j*}$$

VC dimension

The VC dimension h is equal to the maximum number of vectors x_i that can be shattered, namely that can be separated using this set of functions $\{f_\alpha\}$ into two different classes when labeled as ±1 in all the 2^h possible ways

Theorem: The VC dimension of the class of function hyperplanes in n dimensions is n+1.

Features selection

The VC-dimension *h* may depend on the dimension of the input x, e.g. for the class of linear functions $h = n + 1$. Hence reducing the dimension n, namely reducing the number of features of the training data, may indirectly decreases h and in turn limits the VC-confidence term.

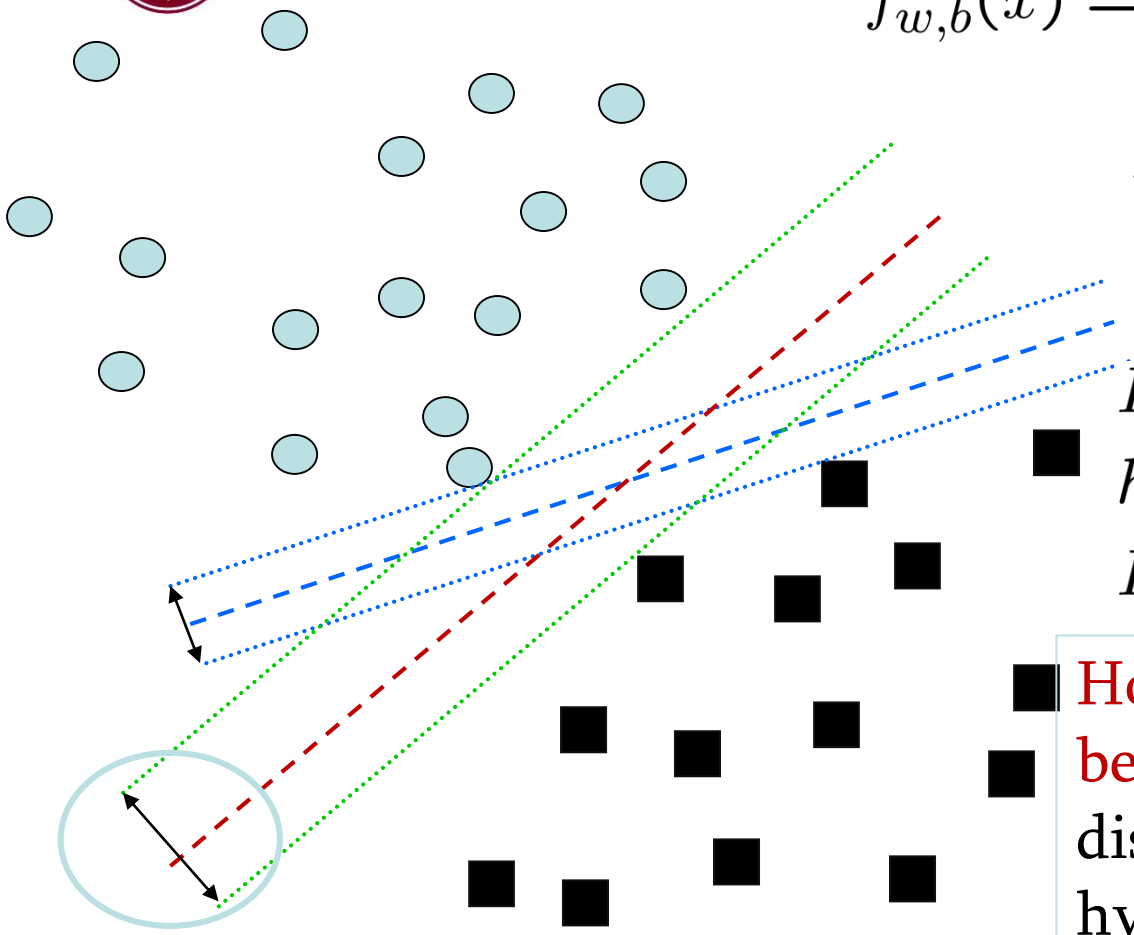
On the other hand reducing *n* may lead to an increase of the empirical risk.

Feature selection is a hot topic in ML

Computation of the VC dimension

The VC dimension of class of function "hyperplanes" is n+1

$$f_{w,b}(x) = sign\{w^T x + b\}$$



In this case we get the same value of the bound for both the functions in the class

$$R_{emp}(f_\alpha) = 0$$

$$h = 3$$

$$R(f_\alpha) \leq 0 + C_{VC}(3, \ell, \eta)$$

However the red one seems better: it "maximizes" the distance among the hyperplane and points in the two sets

Which one of the two is the best ?

Intuition: hyperplane with margin

- A hyperplane that passes too close to the training examples will be sensitive to noise and less likely to generalize well for new data

- Instead, it seems reasonable to expect that a hyperplane that is farthest from all training examples will have better generalization capabilities

Hyperplane with margin

Idea: restrict the choice within the class of linear classification may improve the VC dimension

Linear Classification with tolerance gap (≈margin)

Consider the hyperplane $w^T x + b = 0$. Let $\varrho(x^i; w, b) = \frac{w^T x + b}{\|w\|}$ be the «algebraic distance»[1] of a point x^i from the hyperplane.

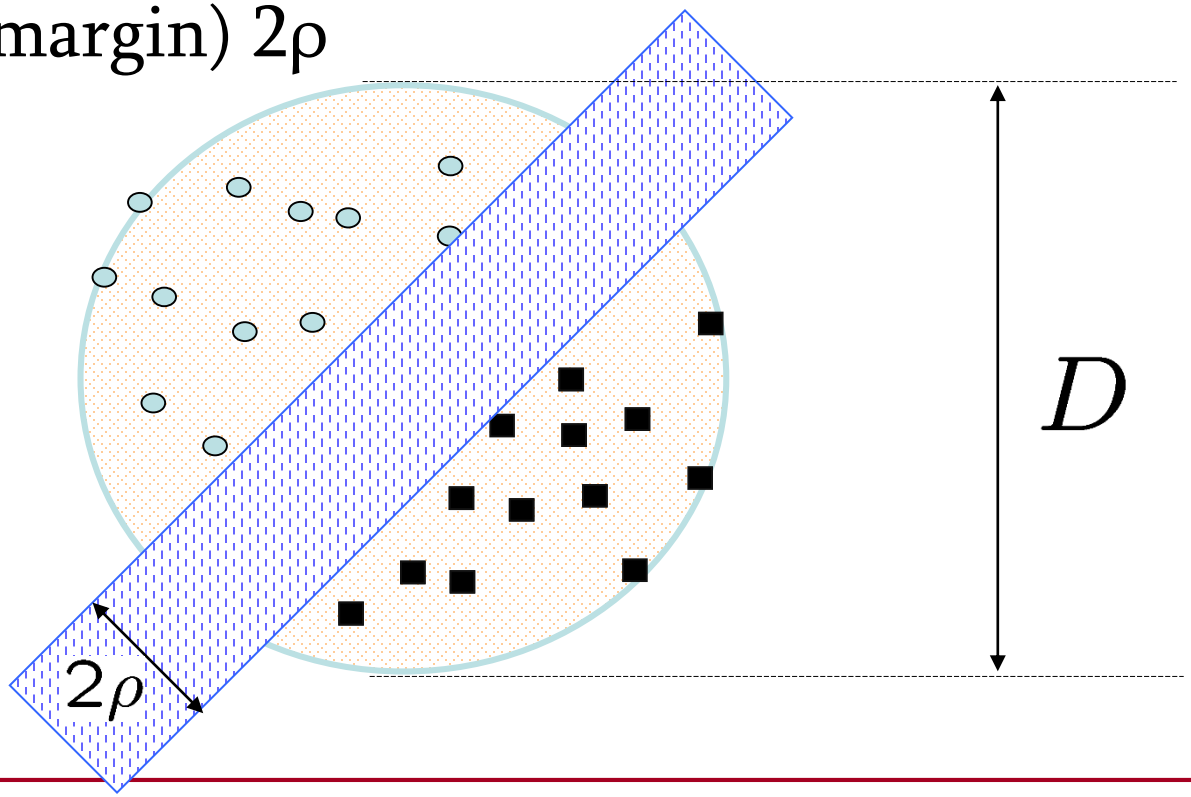
$$\text{Assign } y^i = \begin{cases} 1, & if\ \varrho(x^i; w, b) \geq \rho \\ -1, & if\ \varrho(x^i; w, b) \leq -\rho \end{cases}$$

[1] reading that the distance of a point x^i from the hyperplane is $|\varrho(x^i; w, b)|$

Hyperplane with tolerance gap

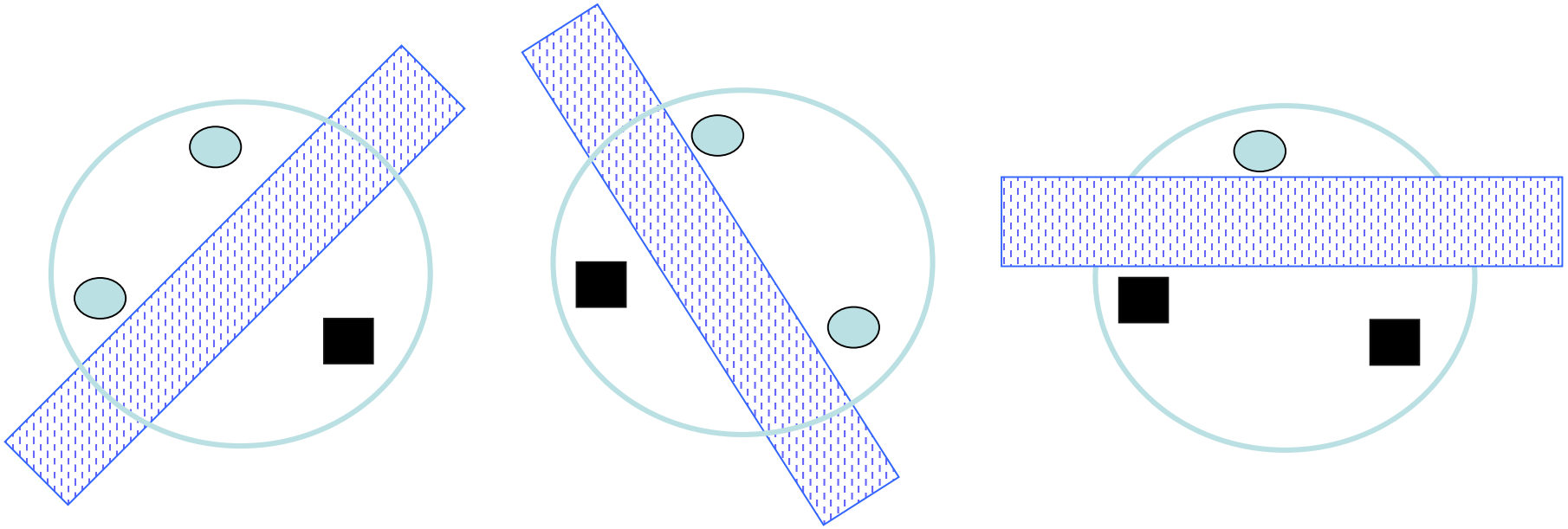
Assume that data x^i stay in a sphere with diameter D=2R, namely $||x^i|| \leq R$

Hyperplane with tolerance gap classifies vectors within a sphere of diameter D and outside the "strip" of tolerance (margin) 2ρ



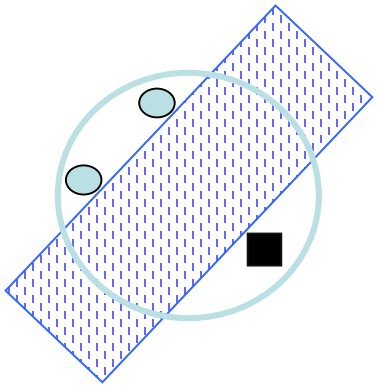
Hyperplane with tolerance gap

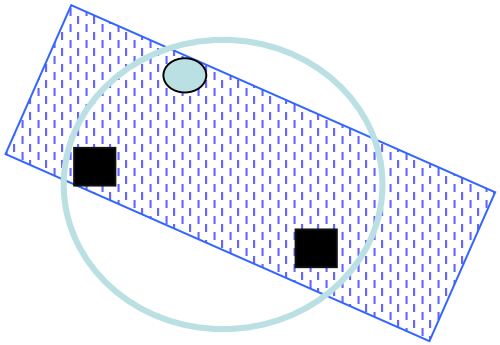
If ρ is small with respect to D, it is still possible to shatter 3 points in R^2



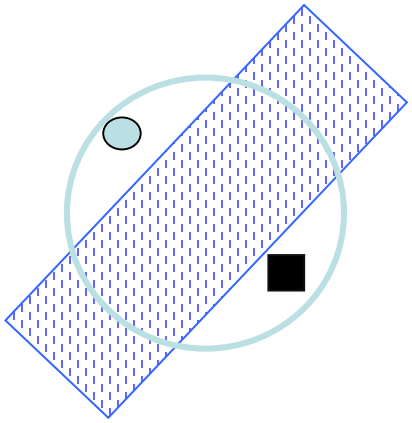
Hyperplane with tolerance gap

If ρ grows with respect to D, the number of points that can be shattered decreases



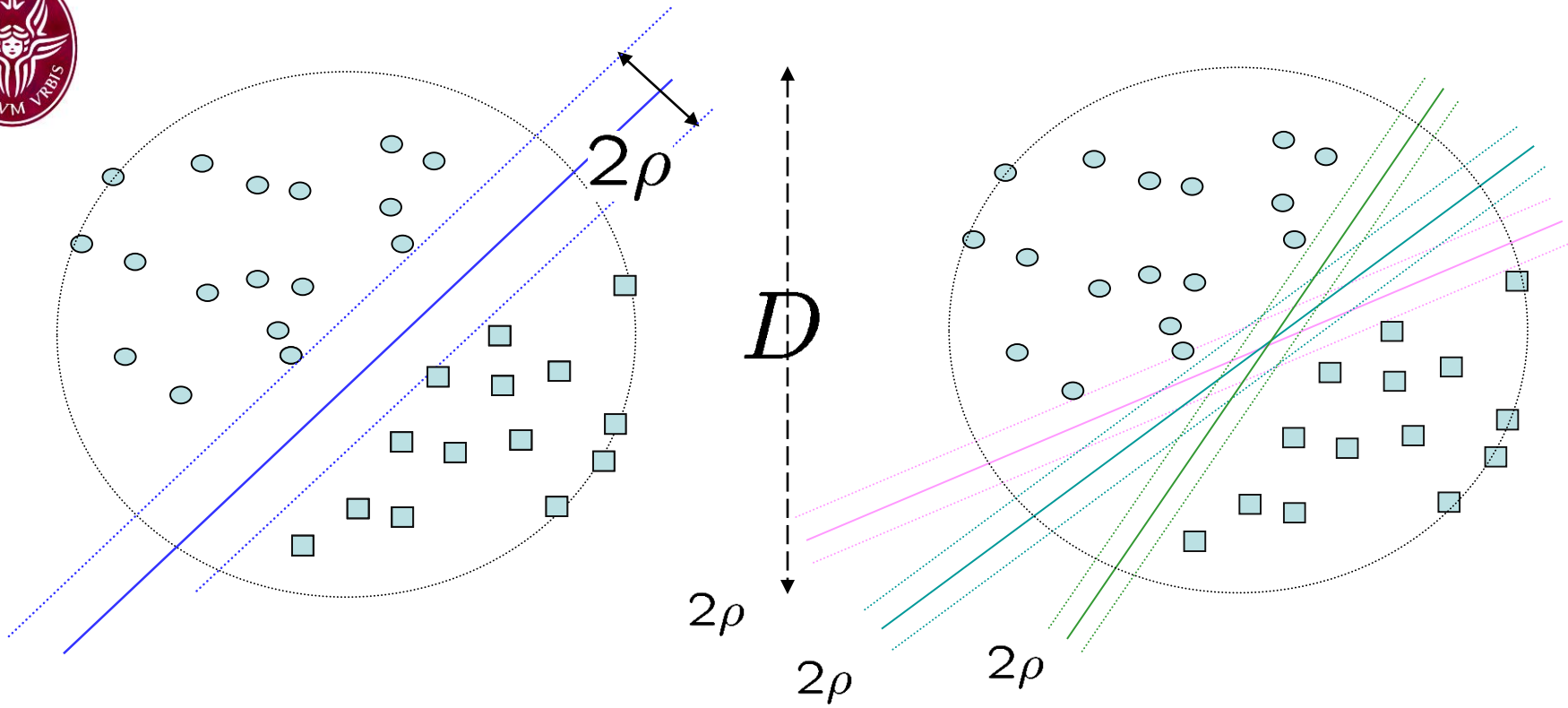


It is no more possible to shatter 3 points with larger ρ



It is still possible to shatter 2 points

Margin and VC confidence



The highest the margin ρ, the lowest the VC dimension h

Margin and VC confidence

The highest the margin ρ, the lowest the VC dimension h

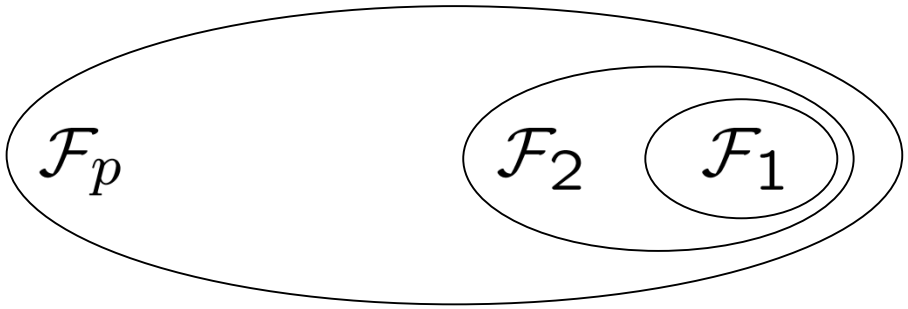
$$h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, n\right\} + 1 \leq n + 1$$

h hyperplane with tolerance gap ≤ h hyperplane

$$\rho_1 \geq \rho_2 \geq \rho_3 \geq \cdots \geq \rho_p$$

$$h_1 \leq h_2 \leq h_3 \leq \cdots \leq h_p$$

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \cdots \subseteq F_p$$



Structural risk minimization
hyperplane with tolerance gap

$$\mathcal{F}_i = \{f : R^n \to R : f(x) = w^T x + b,\ \ \rho(w,b) \geq \rho_i\},$$
$$\rho_1 \geq \rho_2 \geq \ldots \geq \rho_p$$

- Minimizing the upper bound on the risk
  - minimizing the empirical risk
  - maximizing the margin (hence minimizing the VC confidence)
- for each function in the class F_j
  - obtain h_j
  - minimize the empirical risk R^j_{emp}
  - compute the bound U^j on the risk
- Chose the class j with minimum bound

Methods in SRM paradigm

$$\min R_{emp} + C_{VC}(\eta, l, h)$$

Two main approaches

- "Fix" the Empirical risk to a given value and minimize the VC confidence

Support Vector Machines

- Fixed the architecture (i.e. the complexity and VC confidence) and minimize the Empirical risk

Deep Neural Network

Methods in SRM paradigm

In both cases do not forget the two "contrasting" terms in the objective function

Support Vector Machines

- Maximize the margin and control the growth of the empirical risk. Use Kernel trick to allow more complex separating surface

Deep Neural Network

- Fix the architecture include safeguards rules when minimize the ER, e.g.
  - add a regularization term
  - use early stopping rules

The overall learning problem

Both approaches need to set up the values of some hyper-parameters that control the trade off between the two terms in the true objective function

Support Vector Machines

Deep Neural Network

Use of sophisticated trial & errors procedure which includes a two-fold use of the target data to quantify also the expected performance

Learning process

Target Data may have a twofold role

- Training set: the data used for defining the optimization problem and finding the parameters
- Validation set: the data used to check the predictive performance and to adjust the hyper-parameters

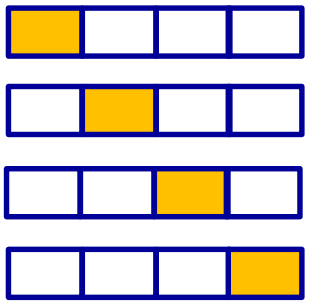
- Test set: data used ins the 2nd phase for checking the accuracy

SAPIENZA
UNIVERSITÀ DI ROMA

K-fold cross validation

If data is plentiful, then one can use some of the available data as training set and a second set of independent data, called *validation set*, to check the predictive performance.

In order to build good models, we wish to use as much of the available data as possible for training. However, if the validation set is small, it will give a relatively noisy estimate of predictive performance. One solution is to use *k-fold cross validation*.



Example of 4-fold cross validation

The available data are partitioned into k groups. Then $k - 1$ of the groups are used as training set and the remaining group as validation. This procedure is then repeated for all k possible choices for the held-out group. The performance scores from the k runs are then averaged.

How Optimization enters ?

In both cases we need to optimize something

Support Vector Machines	Constrained optimization

- The optimization problem is linear constrained with a convex quadratic function

Deep Neural Network	Unconstrained optimization

- The optimization problem is unconstrained highly nonconvex

The Learning model

Quoting:

Pedro Domingos (Communications of the ACM Volume 55 Issue 10, 2012)

A Few Useful Things to Know about Machine Learning

"Suppose you have an application that you think machine learning might be good for. The first problem facing you is the bewildering variety of learning algorithms available.

Which one to use ?

There are literally thousands available, and hundreds more are published each year.

The key to not getting lost in this huge space is to realize that it consists of combinations of just three components.

LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

Optimization

Quoting: Pedro Domingos, *A Few Useful Things to Know about Machine Learning*

"Finally, we need a method to search among the classifiers in the language for the highest-scoring one.

The choice of optimization technique is key to the efficiency of the learner, and also helps determine the classifier produced if the evaluation function has more than one optimum.

It is common for new learners to start out using off-the-shelf optimizers, which are later replaced by custom-designed ones."

Quoting: Pedro Domingos, *A Few Useful Things to Know about Machine Learning*

Table 1. The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Decomposition methods