Lecture 2: Introduction (continued) Learning Machines

Isabelle Guyon
guyoni@inf.ethz.ch

Class Organization

- **Office hours:** Tuesday 10:00AM-12:00AM, CAB E 45 :
- **Homework:** turn in the homework tuesday afternoon.
- **Exercise class:** bring your laptop.
- **List of questions:** answers available on web page.
- **Choose your paper:** http://clopinet.com/isabelle/Projects/ETH/

  Email to: guyoni@inf.ethz.ch

Papers Left

- ***Combining support vector machines with different feature selection strategies*** by Yi-Wei Chen and Chih-Jen Lin (rank top 3, authors of LibSVM.)
- ***Piecewise Linear Regularized Solution Paths*** Saharon Rosset and Ji Zhu (students of Hastie, new algorithm advance.)
- ***Bayesian SVM for feature weighting and selection*** Wei Chu et al.
  (bridges the gap between risk-minimization and bayesian techniques.)

Grading

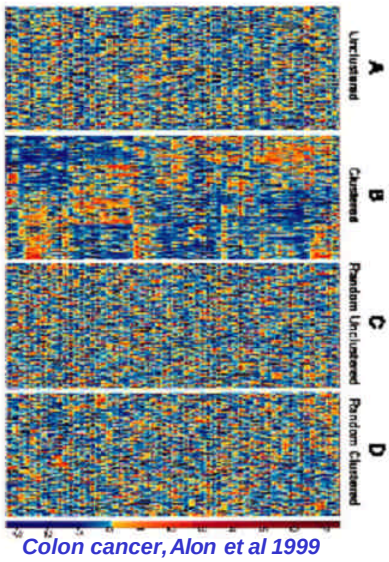
- **Challenge submissions** (10 points):
  - for each dataset, a baseline model is provided having a baseline performance BER_0 and number of features n_0
  - Earn 1 point per dataset for a valid submission having {$BER<BER_0$, any n} or {$BER \leq BER_0$ and $n<n_0$}
  - Earn 2 points per dataset for a submission outperforming the best challenge entry.
- **Paper presentation** (5 points)
- **Final exam,** poster + questions (15 points)
  (contents=5; presentation=5; questions: 5)

Grade = max(6, num_points/4);
Pass: Grade ³ 4

Feature Selection Basics

Textbook “Introduction”

Learning problem



Colon cancer, Alon et al 1999

Data matrix: X

m lines = patterns (data points, examples): samples, patients, documents, images, …

n columns = features: (attributes, input variables): genes, proteins, words, pixels, …

Unsupervised learning

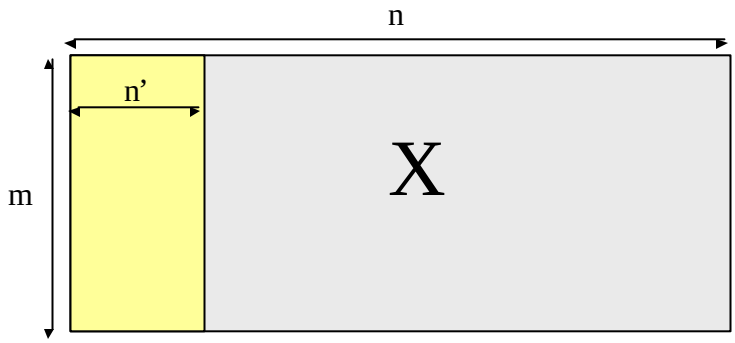
Is there structure in data?

Supervised learning

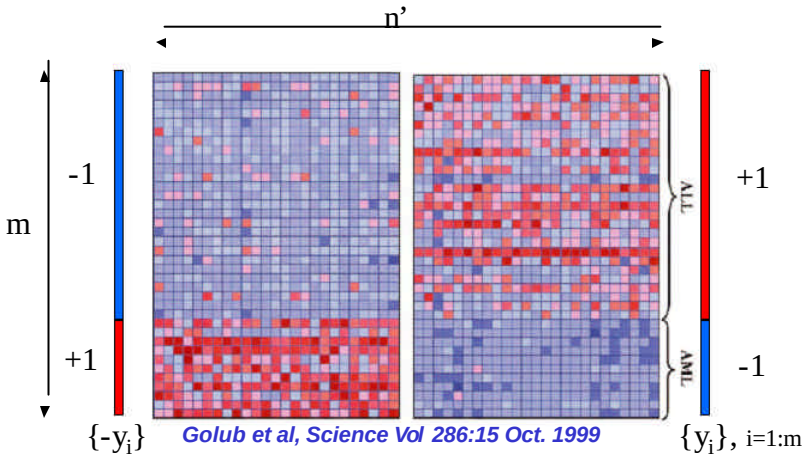
Predict an outcome.

Feature Selection

- **Thousands to millions of low level features**: select the most relevant one to build **better, faster, and easier to understand** learning machines.



Leukemia Diagnosis

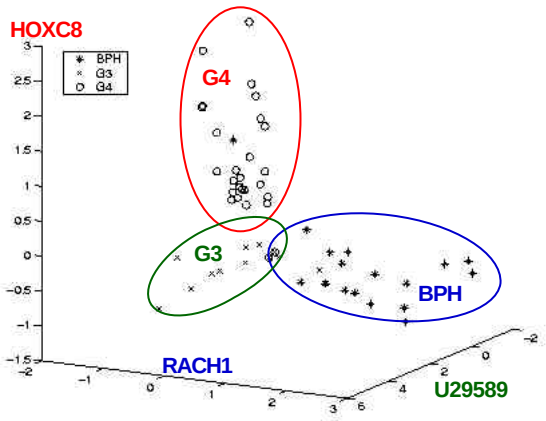


Golub et al, Science Vol 286:15 Oct. 1999

$w_j = \Sigma_i x_{ij} y_i$ $\quad$ $f(\mathbf{x}) = \Sigma_j w_j x_j$

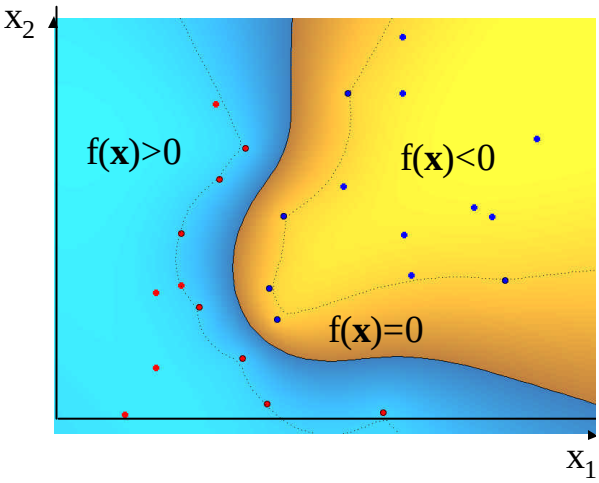
(with "proper normalization" of the data matrix)

Prostate Cancer Genes



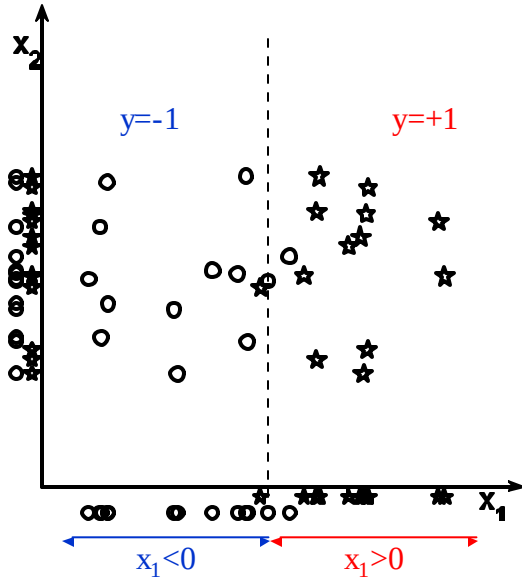
Elisseeff-Weston, 2001

Scatter Plots

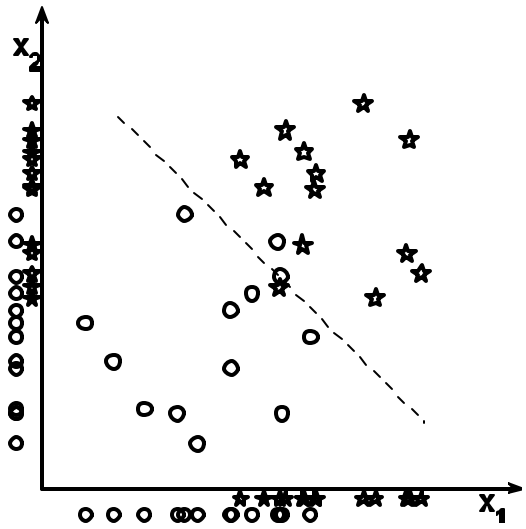


$\mathbf{x}=[x_1,x_2]$

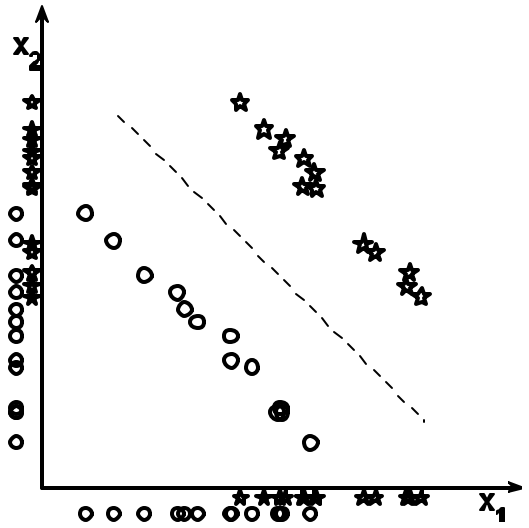
One Informative Feature



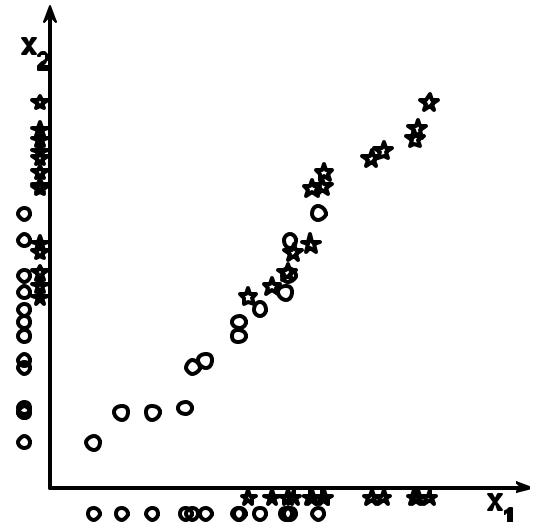
Rotation



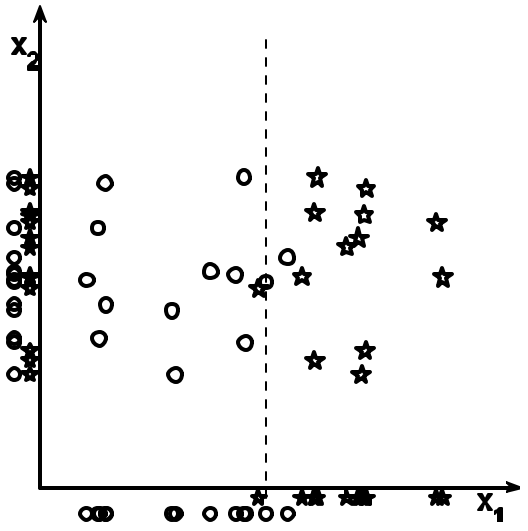
An Informative Pair



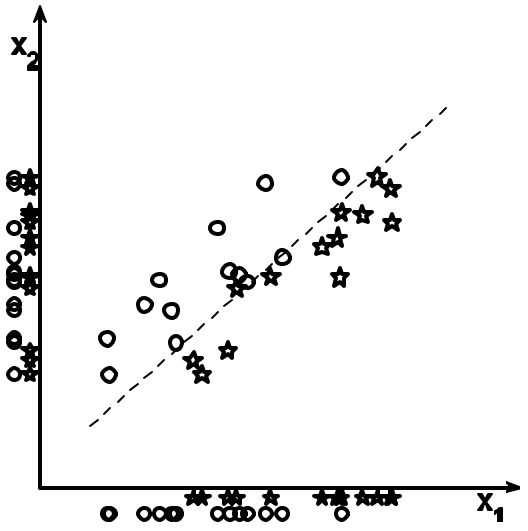
Two no better than one



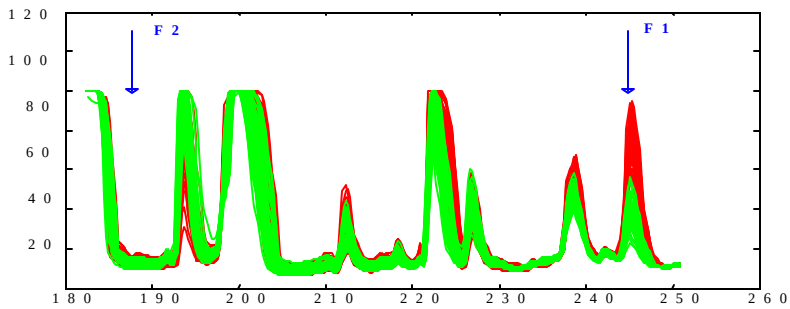
Back to first example



One "Useless" Feature

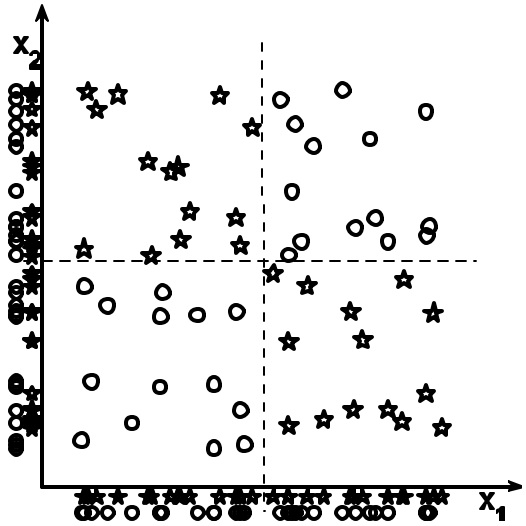


Mass-Spectrometry



F1: The peak of interest
F2: The best local estimate of the baseline.

Two "Useless" Features



Summary

- A lot can be achieved by looking at the separating power of single features.

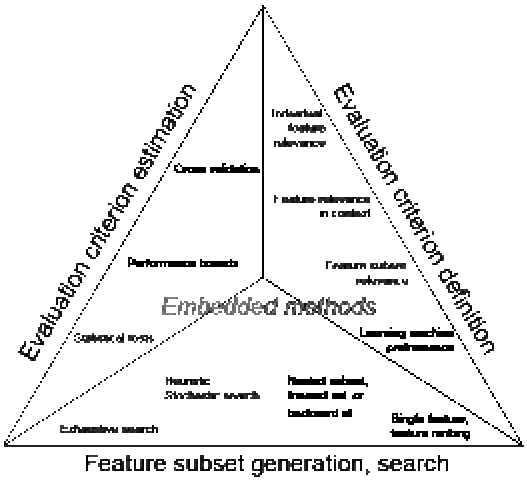
BUT

- Some problems are inherently multi-dimensional, so subsets of features should sometimes be considered.

Nomenclature

- **Univariate method**: considers one variable (feature) at a time.
- **Multivariate method:** considers subsets of variables (features) together.
- **Filter method:** ranks features or feature subsets independently of the predictor (classifier).
- **Wrapper method:** uses a classifier to assess features or feature subsets.

Three Ingredients



Machine Learning

Textbook “Chapter 1”

Some Learning Machines

- Linear discriminant
- Kernel methods
- Neural networks

Conventions

• $X = \{x_{ij}\}$	training data matrix, i=1:m, j=1:n
$\mathbf{x}_i = \{x_j\}$	matrix line, training pattern i
$\mathbf{x}$	test pattern, dim n
• y_i	target value of pattern i
y	target value of test pattern
• $\mathbf{w}$	weight vector, dim n
• a	weight vector, dim m

Linear Discriminant

- $f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x} + b = \sum_{j=1:n} w_j x_j + b$

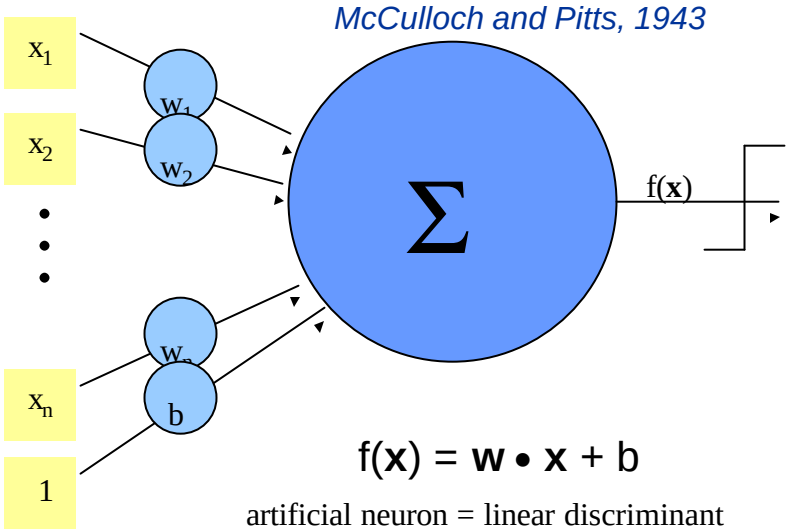
  <u>Linearity in the parameters</u>, NOT in the input components.

- $f(\mathbf{x}) = \mathbf{w} \bullet F(\mathbf{x}) + b = \sum_j w_j \phi_j(\mathbf{x}) + b$ (Perceptron)

- $f(\mathbf{x}) = \sum_{i=1:m} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b$ (Kernel method)

Artificial Neuron

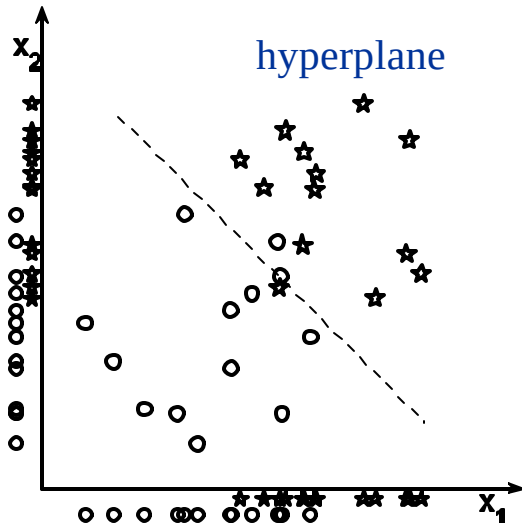
McCulloch and Pitts, 1943



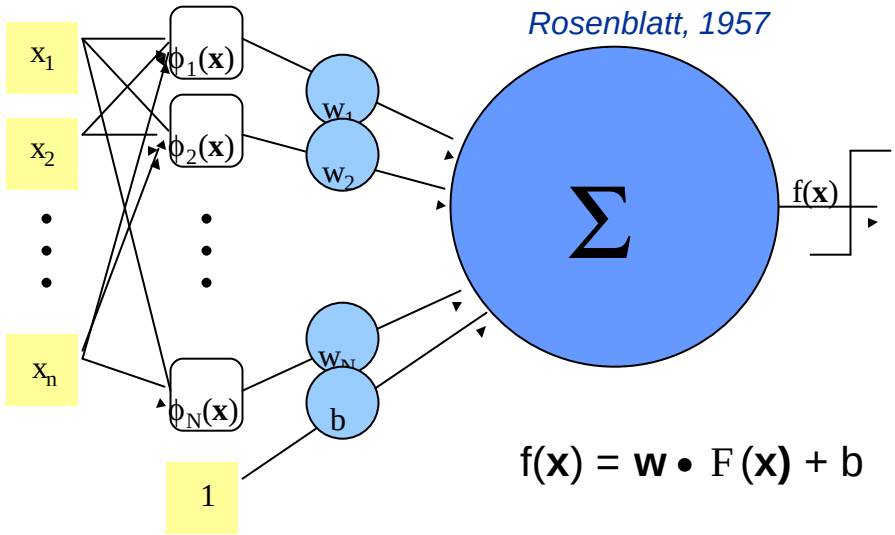
$f(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x} + b$

artificial neuron = linear discriminant

Linear Decision Boundary



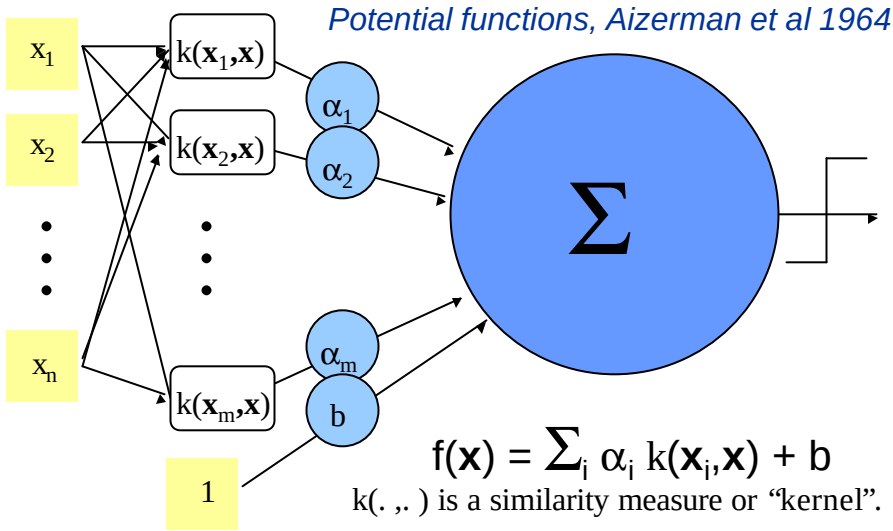
Perceptron



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

Kernel Method

Potential functions, Aizerman et al 1964



$$f(\mathbf{x}) = \sum_i \alpha_i \, k(\mathbf{x}_i,\mathbf{x}) + b$$

k(. ,.) is a similarity measure or "kernel".

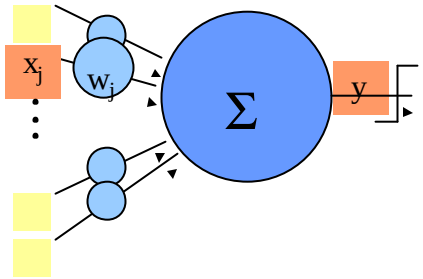
Hebb's Rule

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

$$w_j = \sum_i y_i \, x_{ij}$$

$$\mathbf{w} = \sum_i y_i \, \mathbf{x}_i$$

$$= m+ \; m+ - m- \; m-$$



If m+=m-, classify to the nearest centroid.

Link to "Naïve Bayes".

Kernel "Trick" (Hebb)

- Hebb's rule for the Perceptron:

$$\mathbf{w} = \sum_i y_i \, F(\mathbf{x}_i)$$

$$f(\mathbf{x}) = \mathbf{w} \bullet F(\mathbf{x}) = \sum_i y_i \, F(\mathbf{x}_i) \bullet F(\mathbf{x})$$

- Define a dot product:

$$k(\mathbf{x}_i, \mathbf{x}) = F(\mathbf{x}_i) \bullet F(\mathbf{x})$$

$$f(\mathbf{x}) = \sum_i y_i \, k(\mathbf{x}_i, \mathbf{x})$$

Kernel "Trick" (general)

- $f(\mathbf{x}) = \sum_i \alpha_i \, k(\mathbf{x}_i, \mathbf{x})$
- $k(\mathbf{x}_i, \mathbf{x}) = F(\mathbf{x}_i) \bullet F(\mathbf{x})$

$\Updownarrow$ *Dual forms*

- $f(\mathbf{x}) = \mathbf{w} \bullet F(\mathbf{x})$
- $\mathbf{w} = \sum_i \alpha_i \, F(\mathbf{x}_i)$

What is a Kernel?

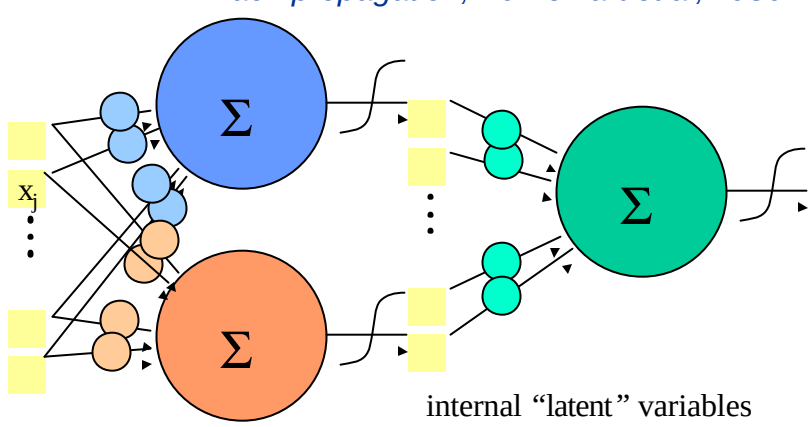
A kernel is a dot product in *some* feature space: $k(\mathbf{s}, \mathbf{t}) = F(\mathbf{s}) \bullet F(\mathbf{t})$

- <u>Examples:</u>
- $k(\mathbf{s}, \mathbf{t}) = \exp(-||\mathbf{s}-\mathbf{t}||^2/\sigma^2)$ Gaussian kernel
- $k(\mathbf{s}, \mathbf{t}) = 1/||\mathbf{s}-\mathbf{t}||$ Potential function
- $k(\mathbf{s}, \mathbf{t}) = (\mathbf{s} \bullet \mathbf{t})^q$ Polynomial kernel

$$\underbrace{([s_1,s_2]\bullet[t_1,t_2])^2}_{k(\mathbf{s},\,\mathbf{t})} = \underbrace{[s_1^2,s_2^2,\sqrt{2}s_1s_2]}_{F(\mathbf{s})} .\underbrace{[t_1^2,t_2^2,\sqrt{2}t_1t_2]}_{F(\mathbf{t})}$$

Multi-Layer Perceptron

Back-propagation, Rumelhart et al, 1986

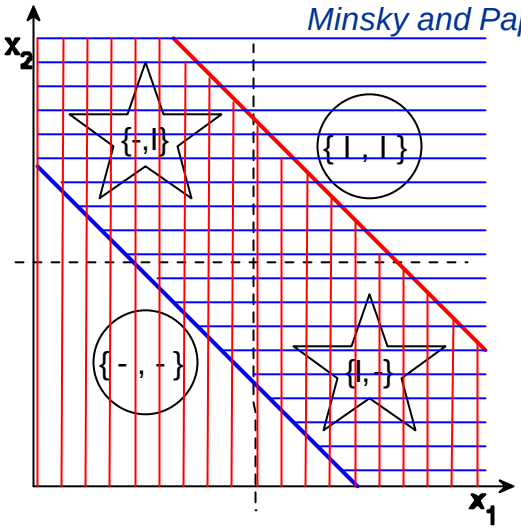


internal "latent" variables

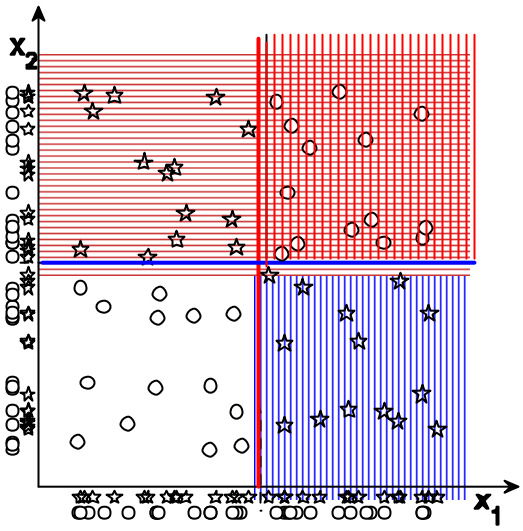
"hidden units"

XOR Problem

Minsky and Papert, 1969



Chessboard Problem



How to Train?

- Define a risk functional R[f(**x**,**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.)

Summary

- With linear threshold units ("neurons") we can build:
  - Linear discriminant
  - Kernel methods
  - Neural networks
- The architectural hyper-parameters may include:
  - The choice of basis functions ϕ (features)
  - The kernel
  - The number of hidden units.

Risk Minimization

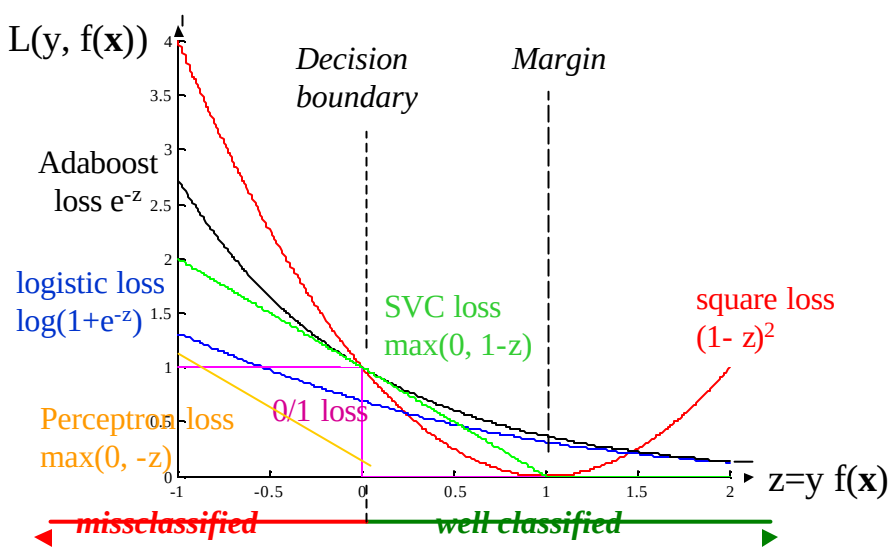
- **Learning problem:** find the best function f(**x**; a) minimizing the risk functional

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

- **Examples are given**:

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

Loss Functions

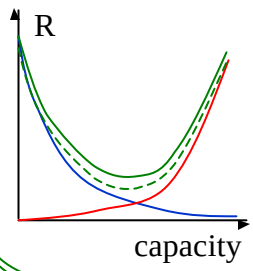


Approximations of R[f]

$R[f] = \int L(f(\mathbf{x};\ a), y)\, dP(\mathbf{x}, y)$

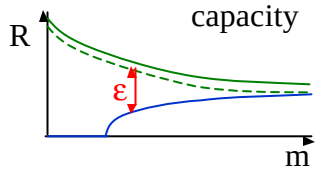
- Empirical risk: $R_{emp}[f] = \Sigma_i\, L(f(\mathbf{x}_i;\ a), y_i)$
- Guaranteed risk:

  $\text{Proba}(\ R[f] > \overbrace{R_{emp}[f] + \varepsilon}^{R_{gua}[f]}\ ) < \delta$



- Penalized/regularized risk:

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



MAP @ RRM

- Maximum A Posteriori (MAP):

  $f = \text{argmax}\ P(f|D)$

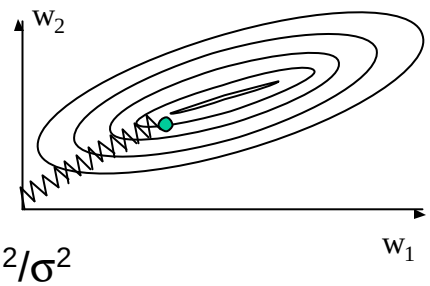
  $= \text{argmax}\ P(D|f)\ P(f)$

  $= \text{argmin}\ \underbrace{-\log P(D|f)}_{\substack{\text{Negative log likelihood} \\ = \text{Empirical risk } R[f]}}\ \underbrace{-\log P(f)}_{\substack{\text{Negative log prior} \\ = \text{Regularizer } \Omega[f]}}$

- Regularized Risk Minimization (RRM):

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

Example: Gaussian Prior



- Linear model:

  $f(\mathbf{x}) = \mathbf{w}.\mathbf{x}$
- Gaussian prior:

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

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

Summary

- Model complexity affects generalization.
- Model complexity may be monitored by the ridge of “weight decay” hyper-parameter.
- Weight decay=Gaussian prior in weight space.

Practical Work

Homework 2:

1) Download the datasets (see homework 1). Download the latest version of CLOP (see homework 1).
2) Download the sample code from:
   http://clopinet.com/isabelle/Projects/ETH/homework2.zip
   Run the sample code main.m.
3) Modify the baseline model (try to obtain better performances or fewer features)
4) Email the result zip file of the results to guyoni@inf.ethz.ch with subject "Homework2" no later than:
   Tuesday November 8th.

Homework 2: Gisette

- **Handwritten digits.**
- Goal: get familiar with the data and result formats. Make a first submission.
- Easiest LM for Gisette: naive and svc.
- Best preprocessing: normalize.
- Easiest feature selection method: s2n.
- Many training examples (6000). Unsuitable for kridge unless subsampling is used.
- Many features (5000). Select features before running neural or rf.

Hyper-parameters

- http://clopinet.com/isabelle/Projects/modelselect/MFAQ.html
- Kernel methods: kridge and svc:

  $k(\mathbf{x}, \mathbf{y}) = (\text{coef0} + \mathbf{x} \bullet \mathbf{y})^{\text{degree}} \exp(-\text{gamma}\ ||\mathbf{x} - \mathbf{y}||^2)$

  $k_{ij} = k(x_i, x_j)$

  $k_{ii} \leftarrow k_{ii} + \text{shrinkage}$
- Naïve Bayes: naive: none
- Neural network: neural

  units, shrinkage, maxiter
- Random Forest: rf (windows only)

  mtry

Best Challenge Performances

Without the validation labels:

- Yi-Wei Chen BER=1.37% feat= 18.26%
- Saharon Rosset BER=1.34 % feat=30%
- Radford Neal BER=1.29% feat=100%

With the validation labels:

- Yi-Wei Chen BER=1.35% feat=18.32%
- Thomas Navin Lal BER=1.31% feat=34%
- Radford Neal BER=1.26% feat=100%
- GhostMiner BER=1.31% feat=100%

Baseline Model

baselineGisette (BER=1.8%, feat=20%)

```
my_classif=svc({'coef0=1',
  'degree=3', 'gamma=0',
  'shrinkage=1'});

my_model=chain({normalize,
  s2n('f_max=1000'), my_classif});
```