Supervised machine learning
An introduction

Romain Raveaux

romain.raveaux@univ-tours.fr

Maître de conférences

Université de Tours

Laboratoire d'informatique (LIFAT)

Equipe RFAI

Outline

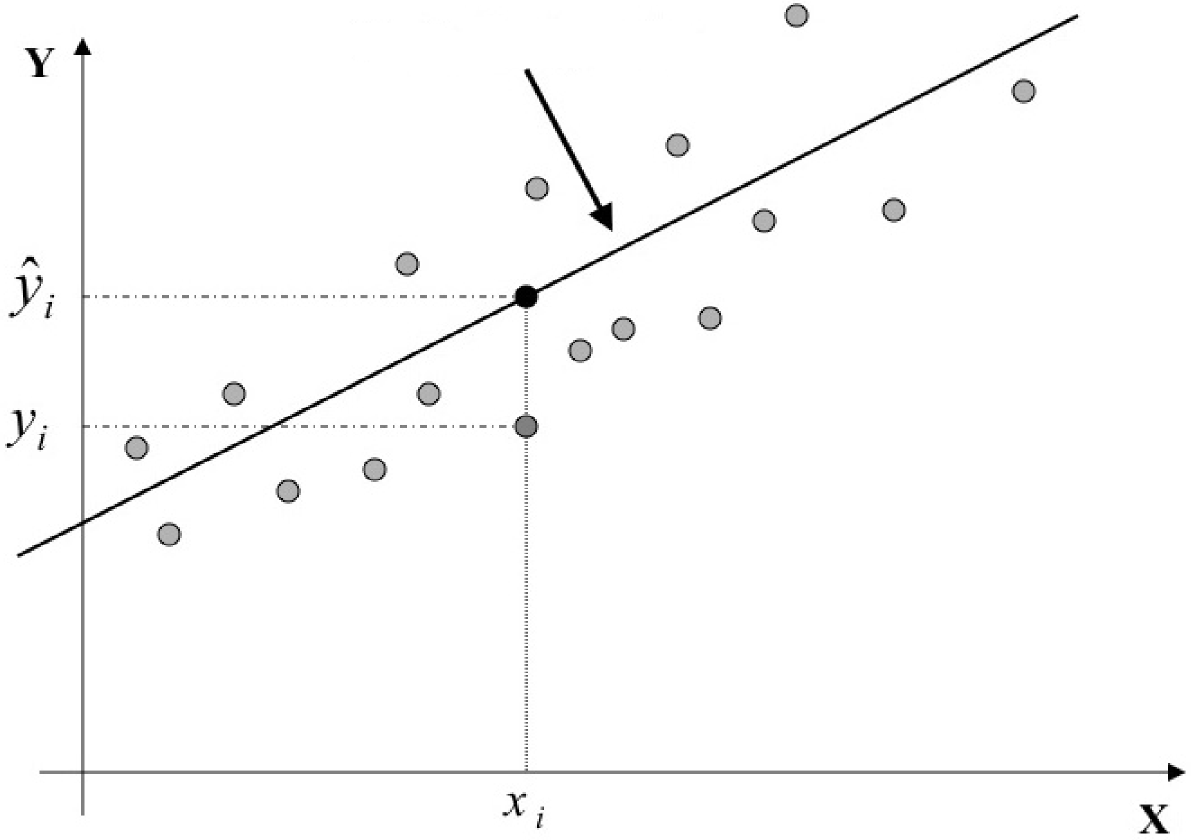
1. Recall
    - Supervised/Re-inforcement/Unsupervised
    - Generative/Discriminative models
    - Fitting probability models
    - Fitting discriminative models
    - Fitting generative models

Recall

- Supervised learning
  - task of learning a function: f
    - that maps an input to an output : $f\colon\ X \rightarrow Y$
    - based on example input-output pairs : $\{x_i, y_i\}^N{}_{i=1}$; $x_i \in X\ et\ y_i \in Y$
- Reinforcement learning $f\ = f_1 \circ f_{\dots} \circ f_k \colon X \rightarrow \bar{Y}$
  - differs from supervised learning
    - f is composed of sub functions (in X) to output $\bar{y}_i$,
    - Incomplete feedback ($\bar{y}_i$ is not completely given $\overline{y_i} \subset y_i \in Y$)
- Unsupervised learning
  - task of learning a function: f
    - that maps an input to an output : $f\colon\ X \rightarrow Z$
    - based on input examples : $\{x_i\}^N{}_{i=1}$; $x_i \in X$
    - No labels (y_i) are given

Recall : Supervised learning

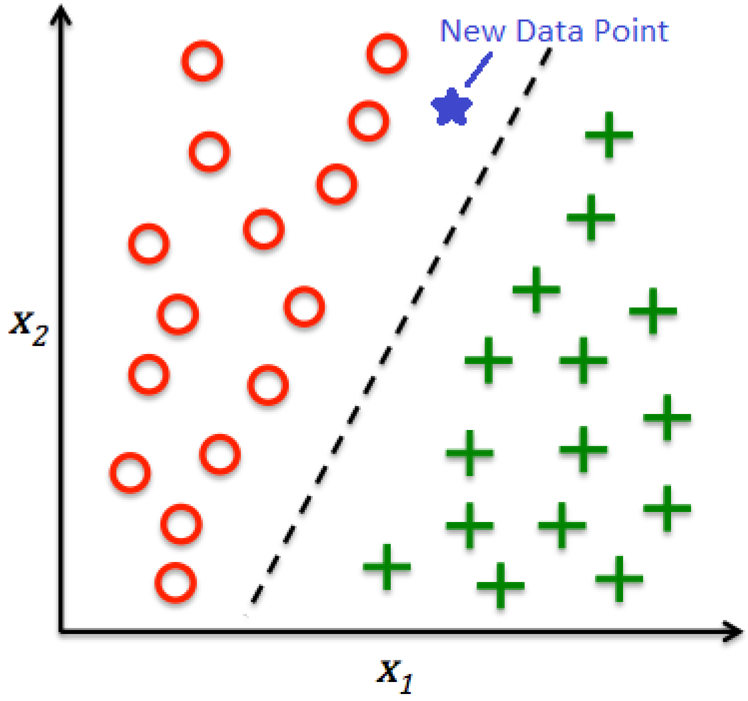
Regression



Credit : https://openclassrooms.com/fr/courses/4525266

Recall : Supervised learning

Classification



[Credit : https://towardsdatascience.com/supervised-learning-basics-of-classification-and-main-algorithms-c16b06806cd3](https://towardsdatascience.com/supervised-learning-basics-of-classification-and-main-algorithms-c16b06806cd3)

Recall : Supervised learning

- Regression : $f: X \rightarrow Y \; and \; y \in \mathbb{R}$
- Classification : $f: X \rightarrow Y \; and \; y \in \{a, b, c, d\}$
- Structured classification : $f: X \rightarrow Y \; and \; y \in \{a, b, c, d\}^{N \times M}$
- Structured Regression : $f: X \rightarrow Y \; and \; y \in \mathbb{R}^{N \times M}$
- Structured Regression : $f: X \rightarrow Y \; and \; y \in \boldsymbol{G} \; (graph \; space)$
- Structured learning when the output is complex:
  - Segmentation mask for instance

Recall : Probability = rational way to deal with uncertainty

Pr(X,Y) = joint probability

Pr(Y|X) = conditional probability

Pr(X) = Marginal probability

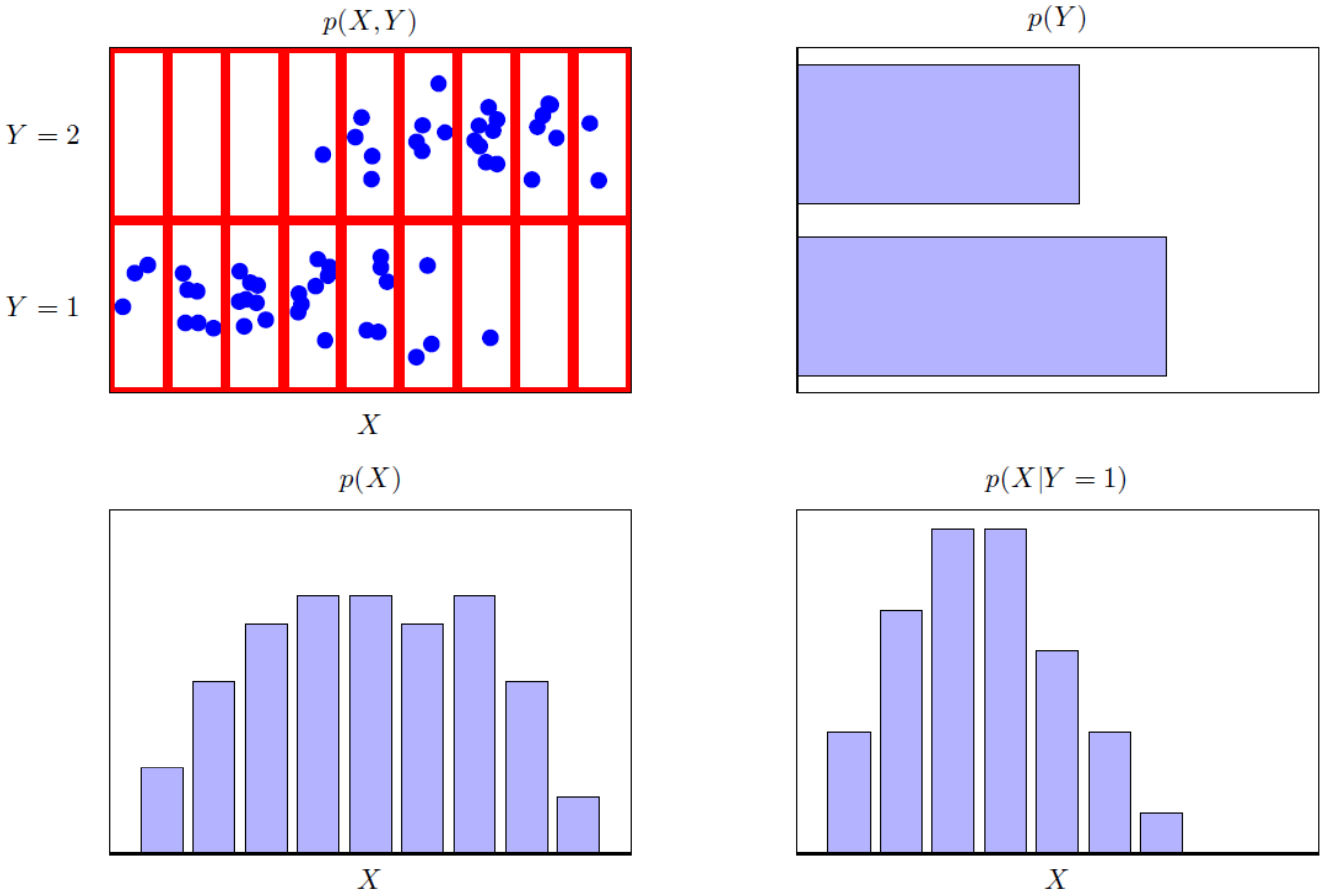
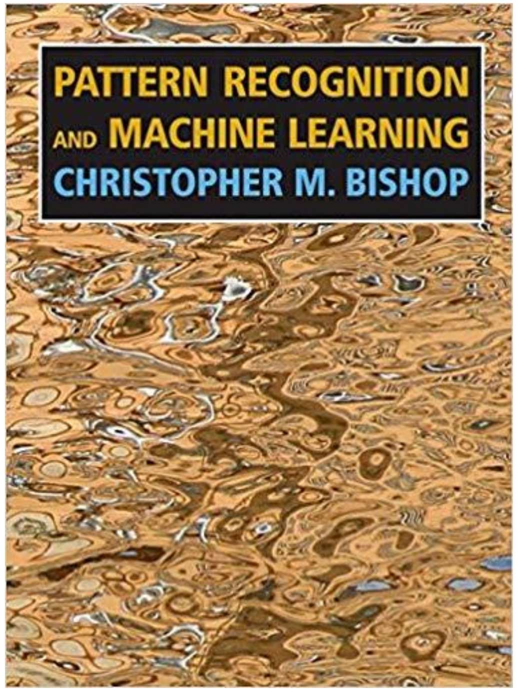


Image taken from



Recall : The Rules of Probability

sum rule $$p(X) = \sum_Y p(X, Y)$$ → called Marginalization

product rule $$p(X, Y) = p(Y|X)p(X).$$

Recall : Bayes' rule : provide a quantification of uncertainty

$$p(X,Y) = p(Y,X).$$ → Symmetry

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

$$p(X) = \sum_{Y} p(X|Y)p(Y)$$

Recall : Supervised learning

- **Discriminative and Generative models**

Models relating the data x to the target y fall into one of two categories. We either:

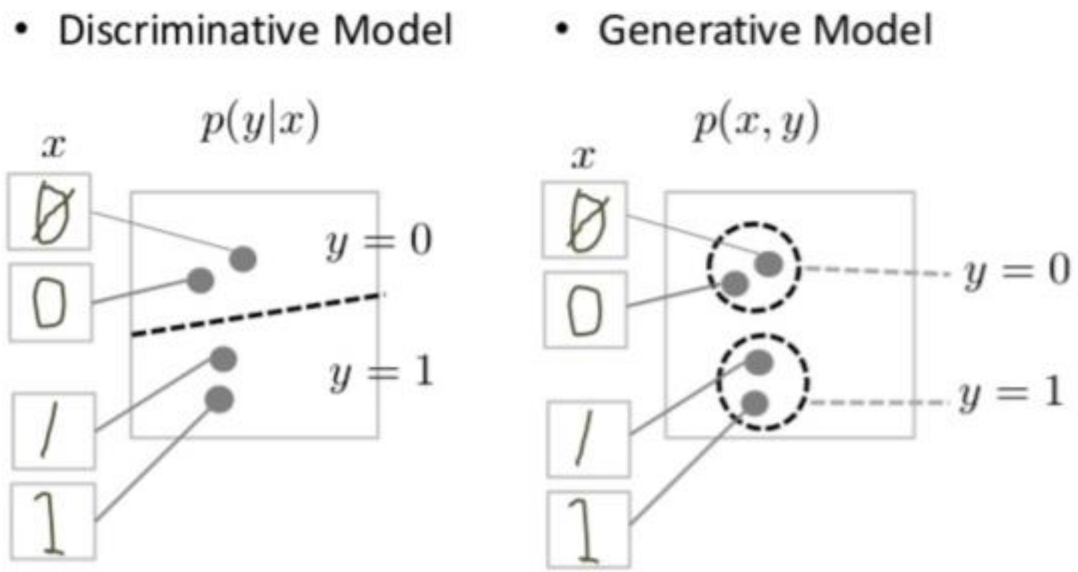
1. model the contingency of the target state on the data $Pr(y|x)$ or
2. model the contingency of the data on the world state $Pr(x|y)$.

The first type of model is termed discriminative. The second is termed generative;

The target y can be a class label ("cat") if we are dealing with a classification problem.

or just $\Pr(x)$: if there are no output labels [this is a generative model]

Recall : Supervised learning



Credit:
https://developers.google.com/machine-learning/gan/images/generative_v_discriminative.png

Recall: Supervised learning : What can we learn?

We want to find these distributions:

- $Pr(y|x)$ or
- $Pr(x|y)$

Where can we act?

1. On the parameters of the distributions.
2. Let's call them W

Parametrized distributions are :

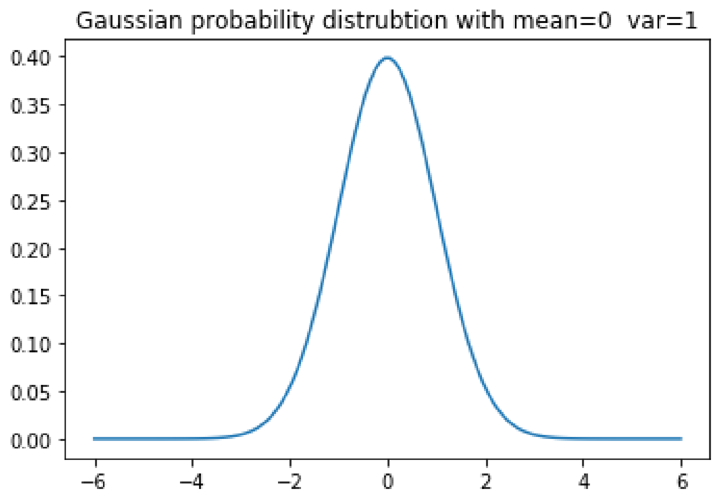
- $Pr(y|x,W)$ or
- $Pr(x|y,W)$

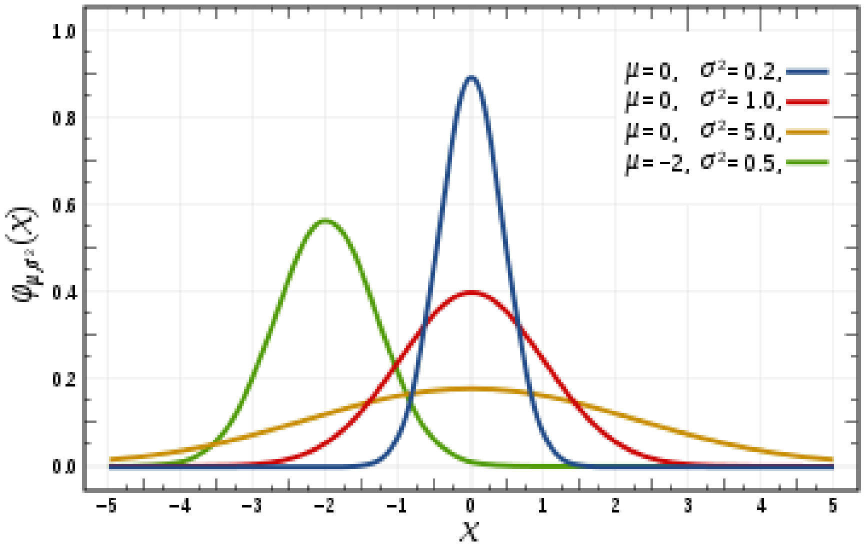
Recall: Supervised learning : fitting probability models

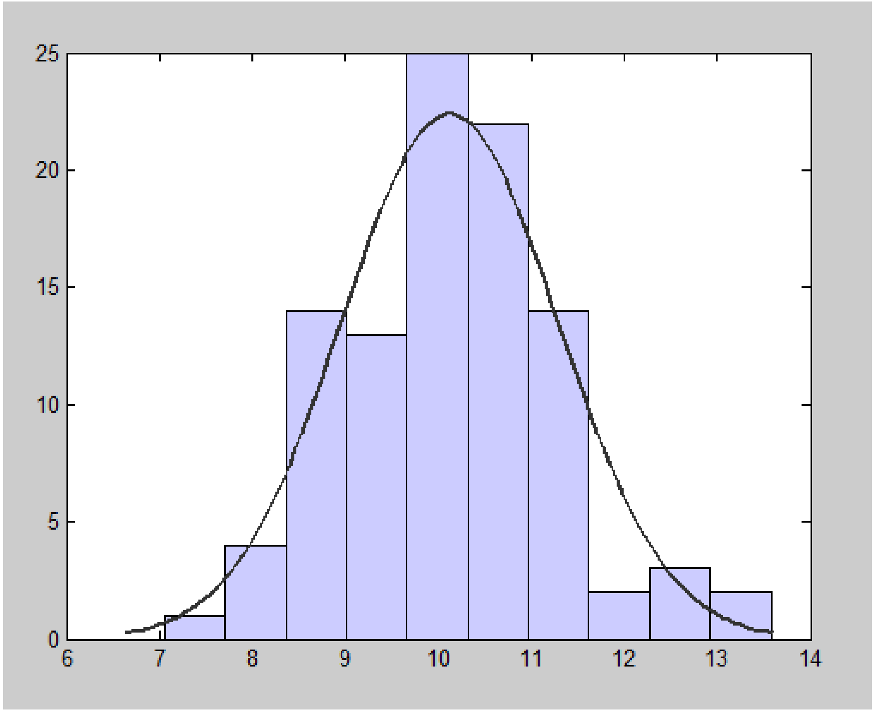
We want to find the parameters (W) such that the probability distribution fits the data

Recall: Supervised learning : fitting probability models

$$Pr(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{\frac{-(x-\mu))^2}{2\sigma^2}\} = \mathcal{N}(x|\mu,\sigma^2)$$







Recall: Supervised learning : fitting probability models

- We find the parameters to fit the data
- Three main ways :
  - 1°) Maximum likelihood

$$W^* = arg \max_W \left[Pr(x_1, \cdots, x_M | W)\right]$$

$$W^* = arg \max_W \left[\prod_{i=1}^{M} Pr(x_i | W)\right]$$

- **Assuming each data point was drawn independently from the distribution (i.i.d)**

Recall: Supervised learning : fitting probability models

- Three ways :
  - 2°) Maximum a posteriori

$$W^* = arg\max_W \left[Pr(W|x_1, \cdots, x_M) \right]$$

$$W^* = arg\max_W \left[\frac{Pr(x_1, \cdots, x_M|W).Pr(W)}{Pr(x_1, \cdots, x_M)} \right]$$

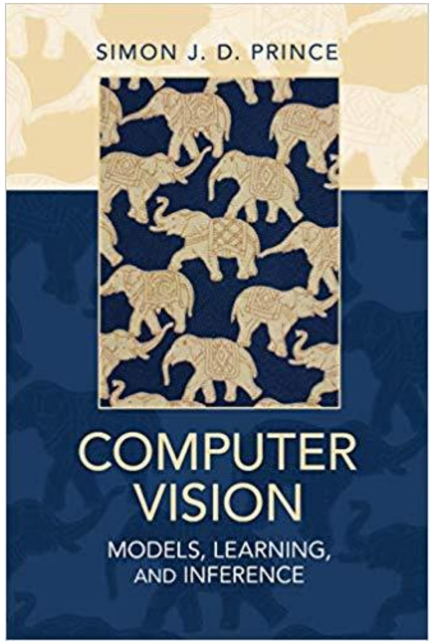
$$W^* = arg\max_W \left[\frac{\prod_{i=1}^{M} Pr(x_i|W).Pr(W)}{Pr(x_1, \cdots, x_M)} \right]$$

$$W^* = arg\max_W \left[\prod_{i=1}^{M} Pr(x_i|W).Pr(W) \right]$$

- Assuming each data point was drawn independently from the distribution (i.i.d)

Recall: Supervised learning : fitting probability models

- Three ways :
  - 3°) The Bayesian approach $Pr(W|x_1, \cdots, x_M)$
  - Beyond the scope of this lecture



A nice book

Recall : Supervised learning : fitting probability models : On a Gaussian

- Fitting by Maximum of likelihood:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2$$

  - Overfitting phenomenon of the maximum likelihood
    - http://romain.raveaux.free.fr/document/Overfittingbiaisedandunbiaisedvariance.html

- Fitting by Maximum a posteriori: W=[mean,variance]
  - Well it depends on the prior (Pr(W))
  - See conjugate distribution
    - the result is proportional to a new distribution which has the same form as the conjugate.
  - Non informative prior

Recall: Learning a discriminative model

$$\mathcal{D} = \{x_i, y_i\}_{i=1}^{M}$$

- A discriminative model : Pr(y|x)
- A discriminative model with its parameters: Pr(y|x,W)
- Maximum of likelihood: That's what your favorite standard neural network does

$$W^* = arg\max_{W} \left[\prod_{i=1}^{M} Pr(y_i|x_i;W) \right]$$

Link to : **Minimizing the cross entropy : a nice trip from Maximum likelihood to Kullback–Leibler divergence**
http://romain.raveaux.free.fr/document/CrossEntropy.html

Recall Learning a discriminative model

- A discriminative model : Pr(y|x)
- A discriminative model with its parameters: Pr(y|x,W)
- Maximum a posteriori (MAP)
  - Your favorite neural network can do that if you precise some strcuture on Pr(W)

$$\mathcal{D} = \{x_i, y_i\}_{i=1}^{M}$$

$$W^* = arg\max_W \left[Pr(W|y_1, \cdots, y_M; x_1, \cdots, x_M) \right]$$

$$W^* = arg\max_W \left[Pr(W|\mathcal{D}) \right]$$

$$W^* = arg\max_W \left[\frac{Pr(\mathcal{D}|W).Pr(W)}{Pr(\mathcal{D})} \right]$$

$$W^* = arg\max_W \left[Pr(\mathcal{D}|W).Pr(W) \right]$$

$$W^* = arg\max_W \left[Pr(y_1, \cdots, y_M|x_1, \cdots, x_M; W).Pr(W) \right]$$

$$W^* = arg\max_W \left[\prod_{i=1}^{M} Pr(y_i|x_i; W).Pr(W) \right]$$

Link to : **Minimizing the least squares error with quadratic regularization**
http://romain.raveaux.free.fr/document/LeastSquaresError.html

Recall: Learning a generative model

- A generative model : Pr(x|y)
- A generative model with its parameters: Pr(x|y,W)
- Maximum of likelihood
  - That's what a Generative Adverserial Network (GAN) does
- Can be used for classification:
  - See Naive Bayes classifier

$$W^* = arg\max_W \left[\prod_{i=1}^{M} Pr(x_i|y_i;W)\right]$$

http://romain.raveaux.free.fr/document/NaiveBayesClassifier.html

Recall: Learning a generative model

- A generative model : Pr(x|y)
- A generative model with its parameters: Pr(x|y,W)
- Maximum of a posteriori

$$W^* = arg\max_{W} \left[\prod_{i=1}^{M} Pr(x_i|y_i;W).Pr(W) \right]$$

Recall : summary

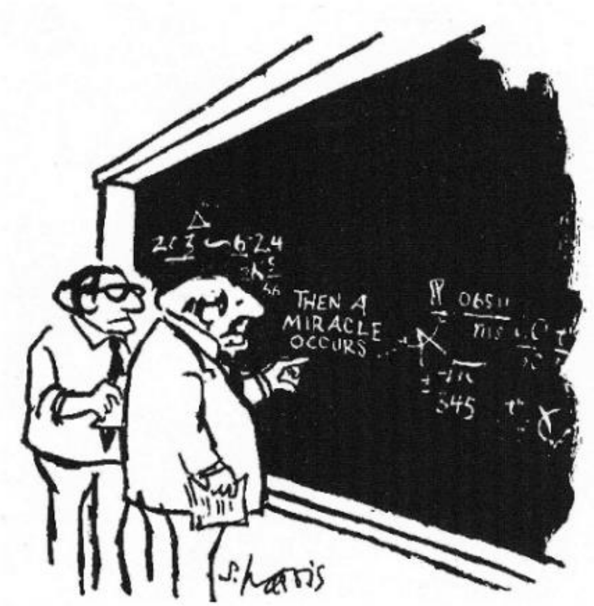
Why is everything an optimization problem?

Why all the formulas?

Why not simply teach algorithms?

Because...

- we want to separate between:
  - what is our ideal goal?
    = **objective function**
  - (how) do we achieve it?
    = **optimization method**
- defining a goal helps in **understanding** the problem
- mathematical formulation allows **re-using existing algorithms** (developed for different tasks)



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."