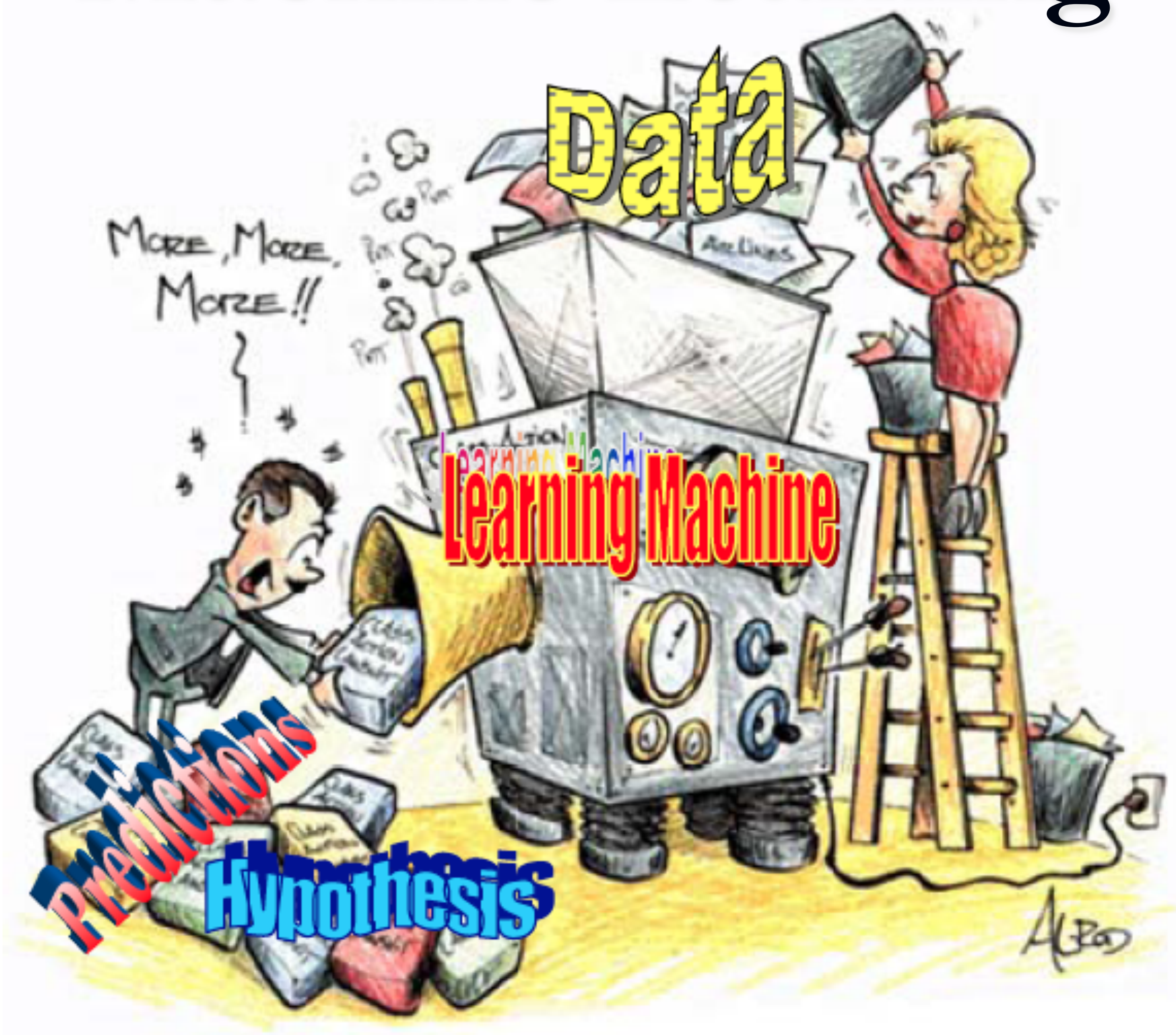
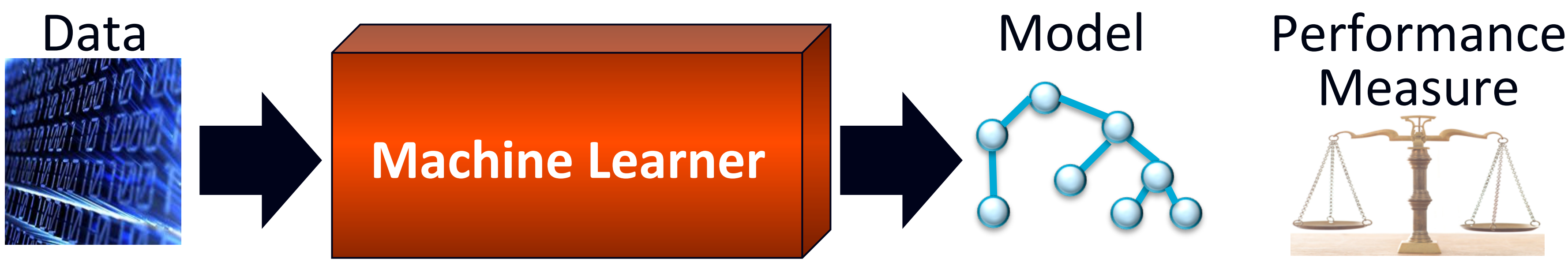
Machine Learning



Machine Learning in a Nutshell

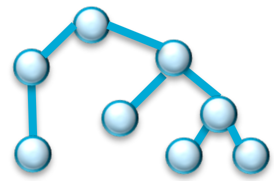


Machine Learning in a Nutshell

Data

Machine Learner

Model



Performance Measure

Data with attributes

ID	A1	Reflex	RefLow	RefHigh	Label
1	5.6	Normal	3.4	7	No
2	5.5	Normal	2.4	5.7	No
3	5.3	Normal	2.4	5.7	Yes
4	5.3	Elevated	2.4	5.7	No
5	6.3	Normal	3.4	7	No
6	3.3	Normal	2.4	5.7	Yes
7	5.1	Decreased	2.4	5.7	Yes
8	4.2	Normal	2.4	5.7	Yes
…	…	…	…	…	…

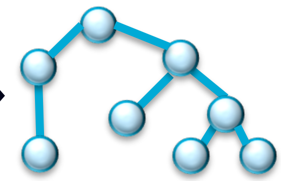
Instance $x_i \in \mathcal{X}$
with label $y_i \in \mathcal{Y}$

Machine Learning in a Nutshell

Data

Machine Learner

Model



Performance Measure

Data with attributes

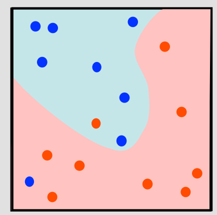
ID	A1	Reflex	RefLow	RefHigh	Label
1	5.6	Normal	3.4	7	No
2	5.5	Normal	2.4	5.7	No
3	5.3	Normal	2.4	5.7	Yes
4	5.3	Elevated	2.4	5.7	No
5	6.3	Normal	3.4	7	No
6	3.3	Normal	2.4	5.7	Yes
7	5.1	Decreased	2.4	5.7	Yes
8	4.2	Normal	2.4	5.7	Yes
…	…	…	…	…	…

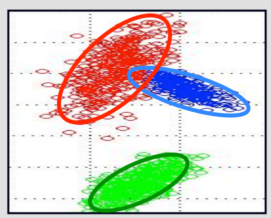
Instance $x_i \in \mathcal{X}$
with label $y_i \in \mathcal{Y}$

Model $f : \mathcal{X} \mapsto \mathcal{Y}$

Logistic regression

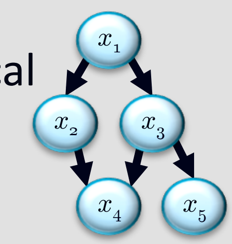
Support vector machines





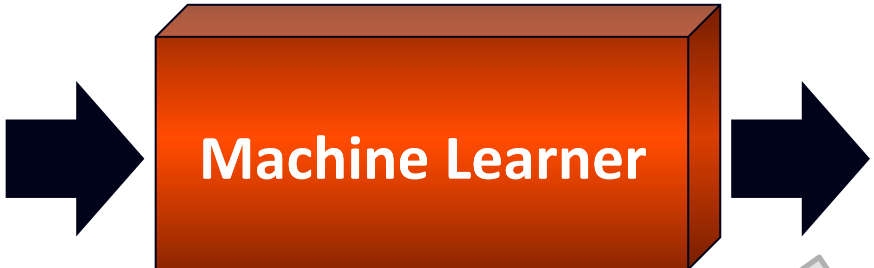
Mixture Models

Hierarchical Bayesian Networks

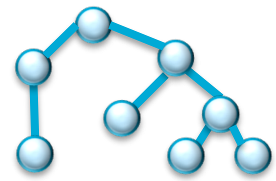


Machine Learning in a Nutshell

Data



Model



Performance Measure

Data with attributes

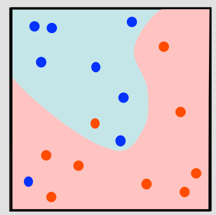
ID	A1	Reflex	RefLow	RefHigh	Label
1	5.6	Normal	3.4	7	No
2	5.5	Normal	2.4	5.7	No
3	5.3	Normal	2.4	5.7	Yes
4	5.3	Elevated	2.4	5.7	No
5	6.3	Normal	3.4	7	No
6	3.3	Normal	2.4	5.7	Yes
7	5.1	Decreased	2.4	5.7	Yes
8	4.2	Normal	2.4	5.7	Yes
…	…	…	…	…	…

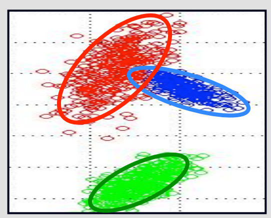
Instance $x_i \in \mathcal{X}$
with label $y_i \in \mathcal{Y}$

Model $f : \mathcal{X} \mapsto \mathcal{Y}$

Logistic regression

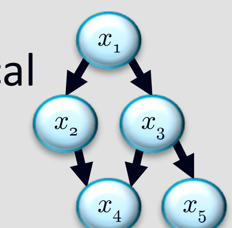
Support vector machines





Mixture Models

Hierarchical Bayesian Networks

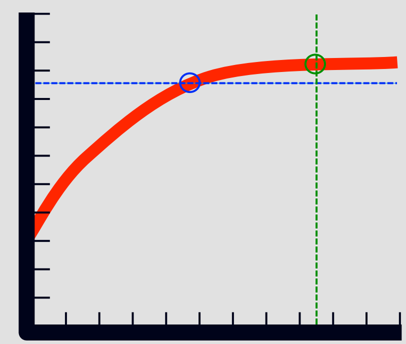


Evaluation

Measure predicted labels vs actual labels on test data

Learning Curve

Performance

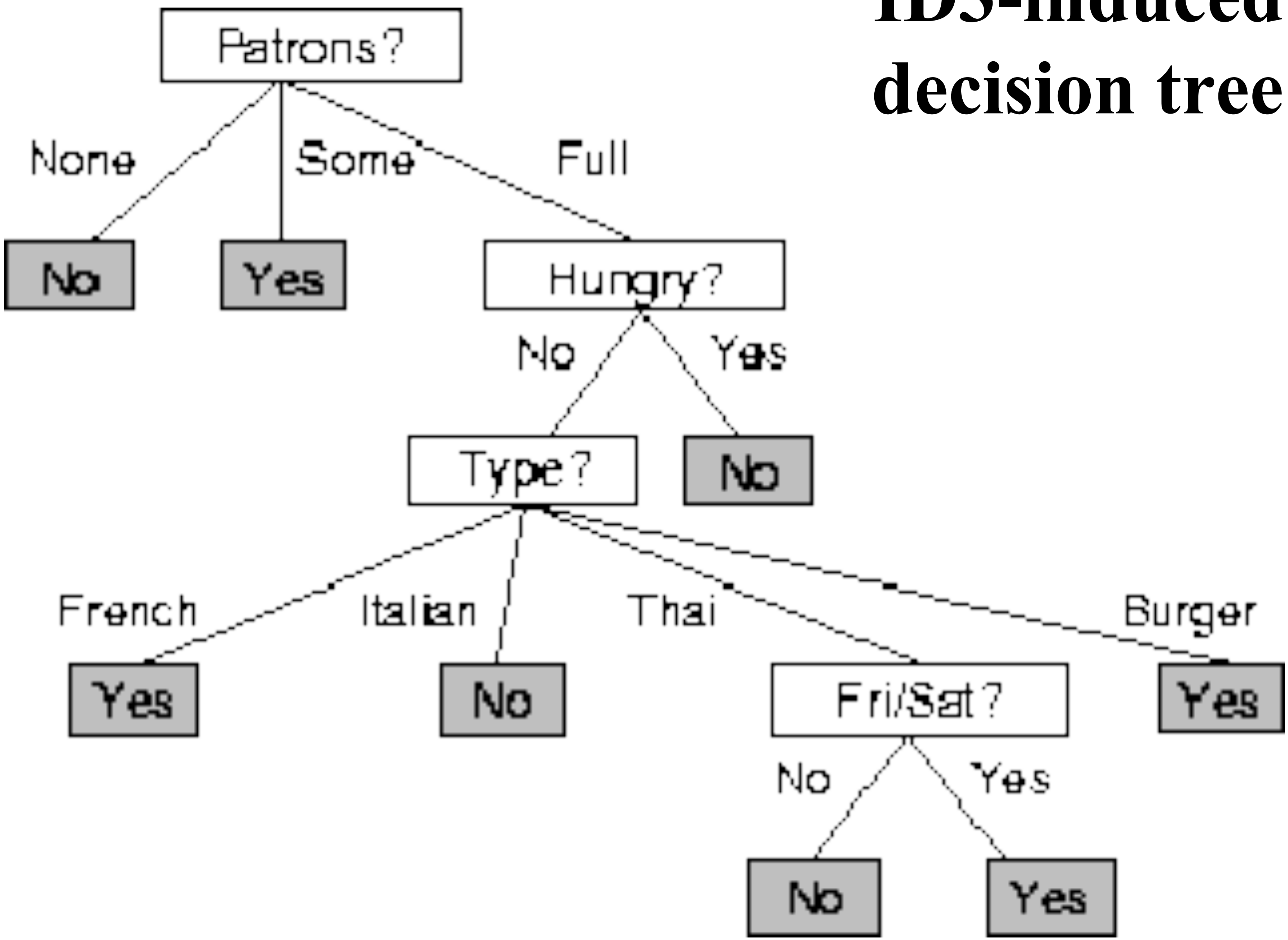


Training Examples

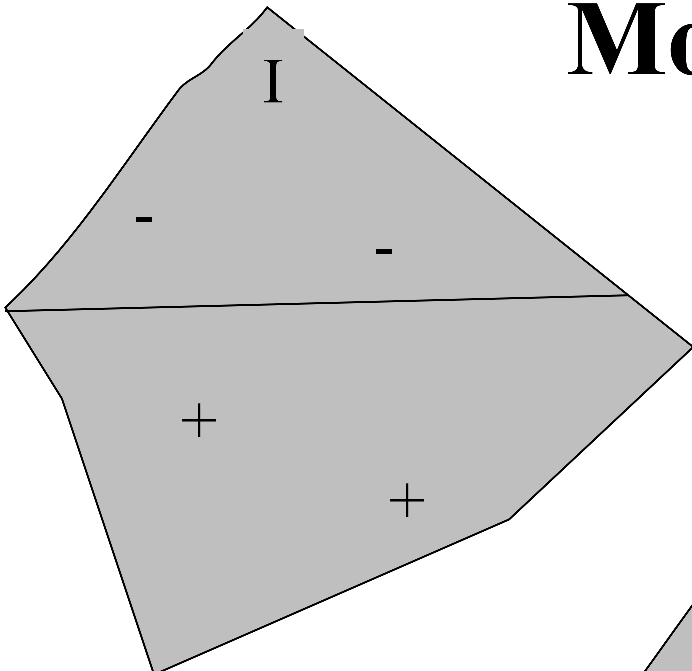
A training set

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

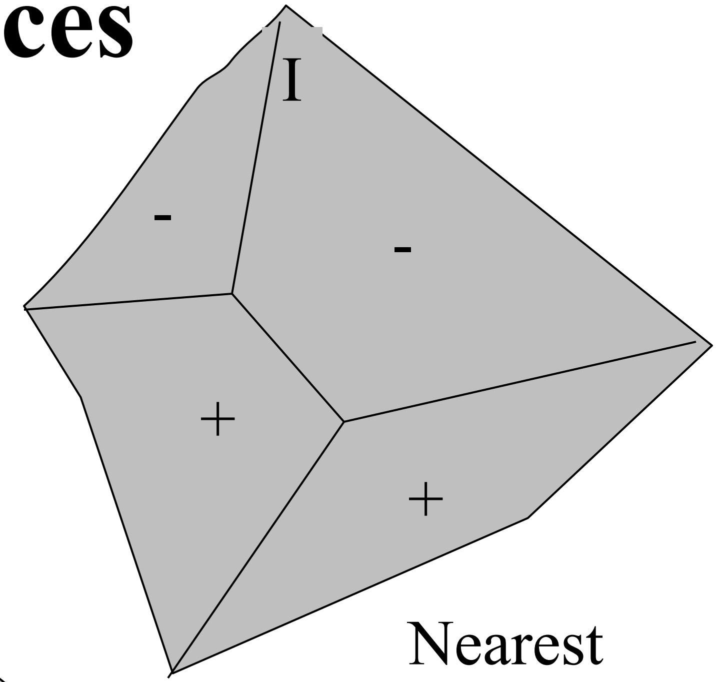
ID3-induced decision tree



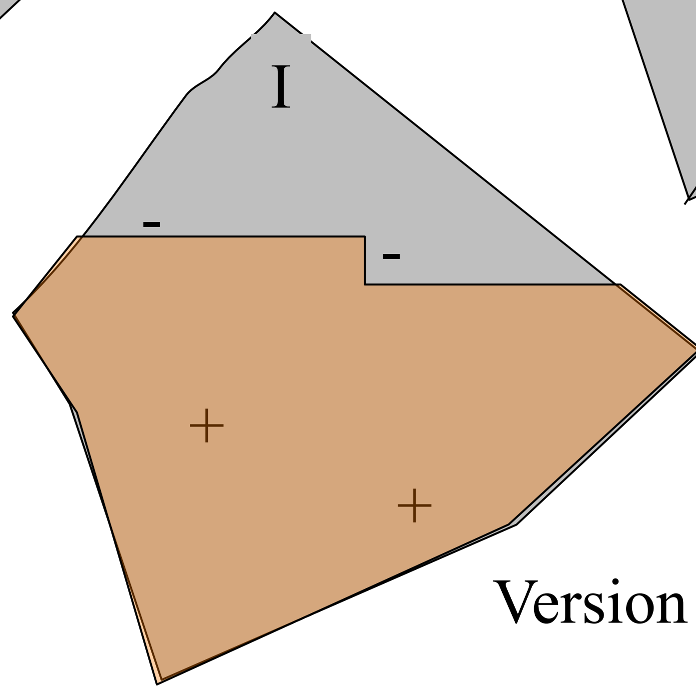
Model spaces



Decision tree

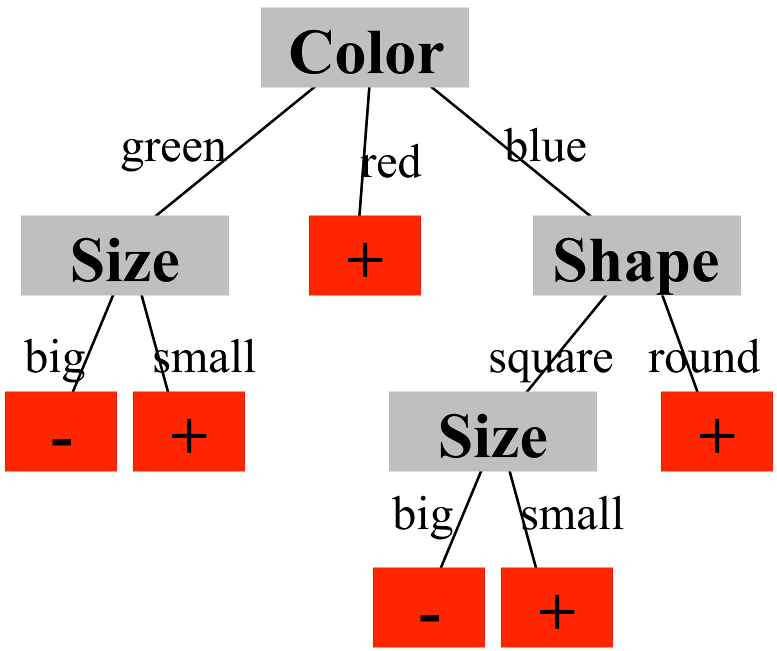


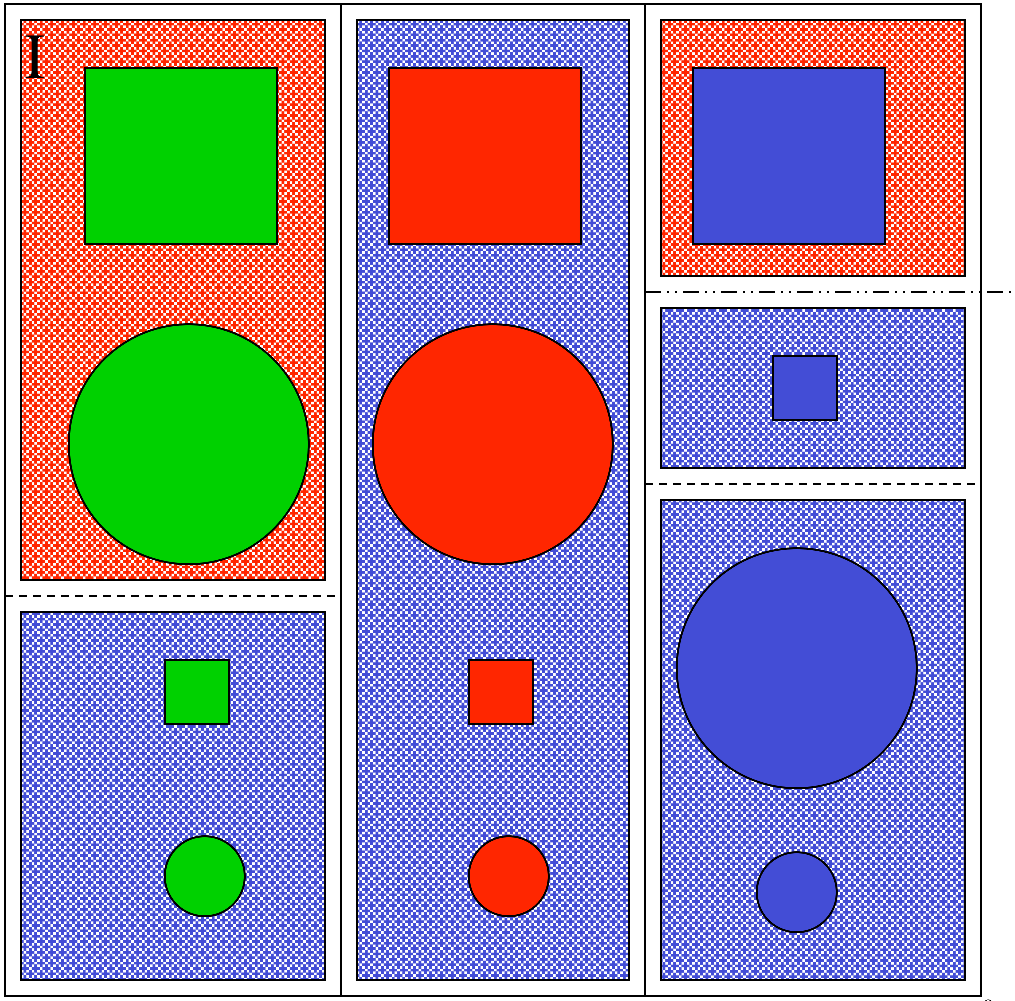
Nearest neighbor



Version space

Decision tree-induced partition – example





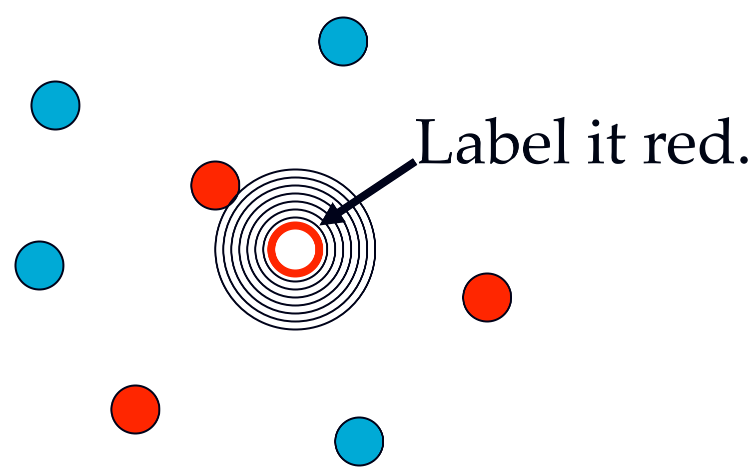
k-Nearest Neighbor Instance-Based Learning

Some material adapted from slides by Andrew Moore, CMU.

Visit http://www.autonlab.org/tutorials/ for Andrew's repository of Data Mining tutorials.

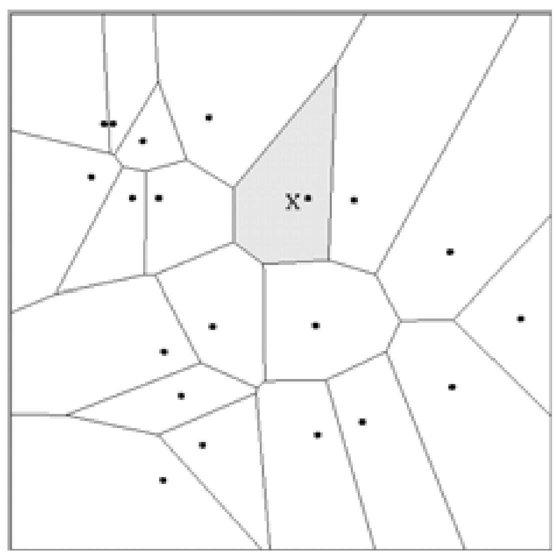
1-Nearest Neighbor

- One of the simplest of all machine learning classifiers
- Simple idea: label a new point the same as the closest known point



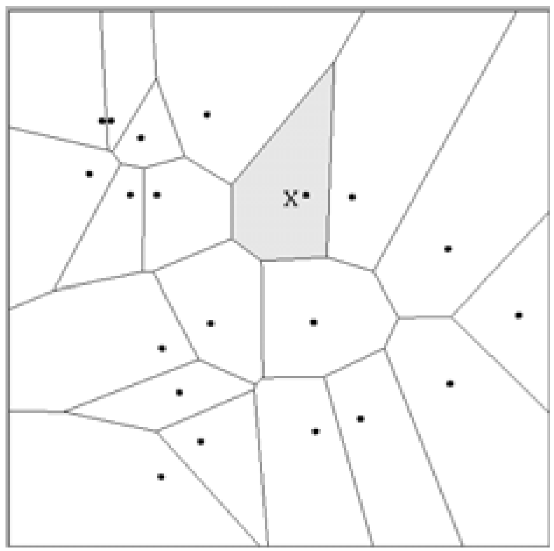
1-Nearest Neighbor

- A type of instance-based learning
  - Also known as “memory-based” learning
- Forms a Voronoi tessellation of the instance space

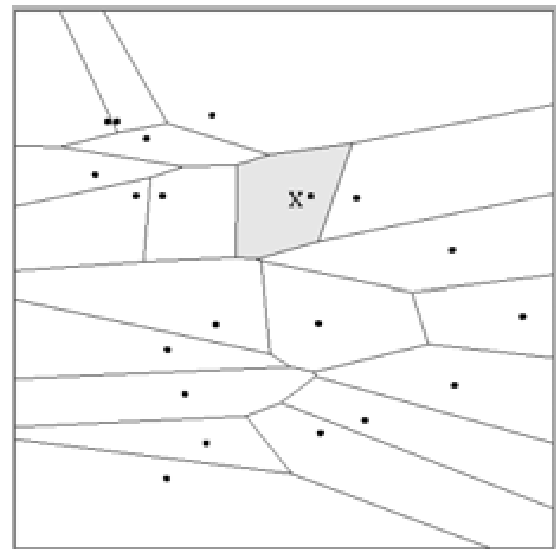


Distance Metrics

- Different metrics can change the decision surface



$\text{Dist}(\mathbf{a},\mathbf{b}) = (a_1 - b_1)^2 + (a_2 - b_2)^2$



$\text{Dist}(\mathbf{a},\mathbf{b}) = (a_1 - b_1)^2 + (3a_2 - 3b_2)^2$

- Standard Euclidean distance metric:
  - Two-dimensional: $\text{Dist}(a,b) = \text{sqrt}((a_1 - b_1)^2 + (a_2 - b_2)^2)$
  - Multivariate: $\text{Dist}(a,b) = \text{sqrt}(\sum (a_i - b_i)^2)$

Adapted from "Instance-Based Learning" lecture slides by Andrew Moore, CMU.

Four Aspects of an Instance-Based Learner:

1. A distance metric
2. How many nearby neighbors to look at?
3. A weighting function (optional)
4. How to fit with the local points?

Adapted from "Instance-Based Learning" lecture slides by Andrew Moore, CMU.

1-NN's Four Aspects as an Instance-Based Learner:

1. A distance metric
   - *Euclidian*
2. How many nearby neighbors to look at?
   - *One*
3. A weighting function (optional)
   - *Unused*
4. How to fit with the local points?
   - *Just predict the same output as the nearest neighbor.*

Adapted from "Instance-Based Learning" lecture slides by Andrew Moore, CMU.

Zen Gardens

Mystery of renowned zen garden revealed [CNN Article]
Thursday, September 26, 2002 Posted: 10:11 AM EDT (1411 GMT)

LONDON (Reuters) -- For centuries visitors to the renowned Ryoanji Temple garden in Kyoto, Japan have been entranced and mystified by the simple arrangement of rocks.

The five sparse clusters on a rectangle of raked gravel are said to be pleasing to the eyes of the hundreds of thousands of tourists who visit the garden each year.

Scientists in Japan said on Wednesday they now believe they have discovered its mysterious appeal.

"We have uncovered the implicit structure of the Ryoanji garden's visual ground and have shown that it includes an abstract, minimalist depiction of natural scenery," said Gert Van Tonder of Kyoto University.

The researchers discovered that the empty space of the garden evokes a hidden image of a branching tree that is sensed by the unconscious mind.

"We believe that the unconscious perception of this pattern contributes to the enigmatic appeal of the garden," Van Tonder added.

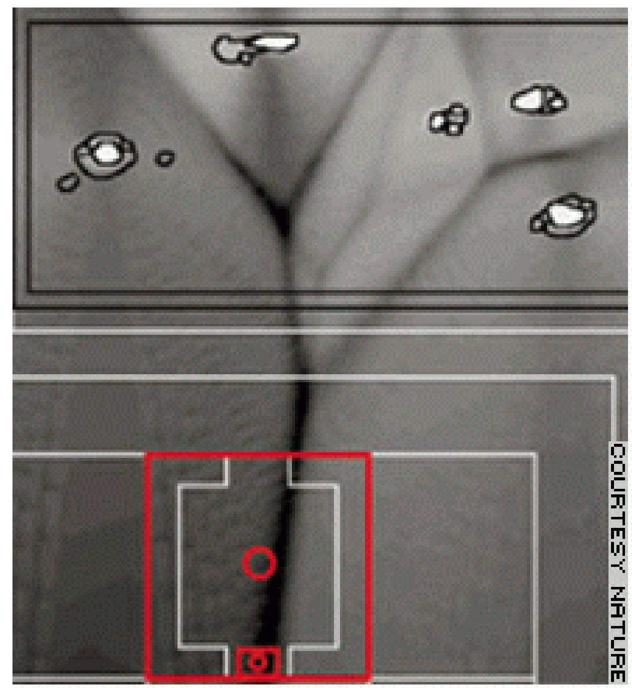
He and his colleagues believe that whoever created the garden during the Muromachi era between 1333-1573 knew exactly what they were doing and placed the rocks around the tree image.

By using a concept called medial-axis transformation, the scientists showed that the hidden branched tree converges on the main area from which the garden is viewed.

The trunk leads to the prime viewing site in the ancient temple that once overlooked the garden. It is thought that abstract art may have a similar impact.

"There is a growing realisation that scientific analysis can reveal unexpected structural features hidden in controversial abstract paintings," Van Tonder said

Ryoanji Temple garden in Kyoto

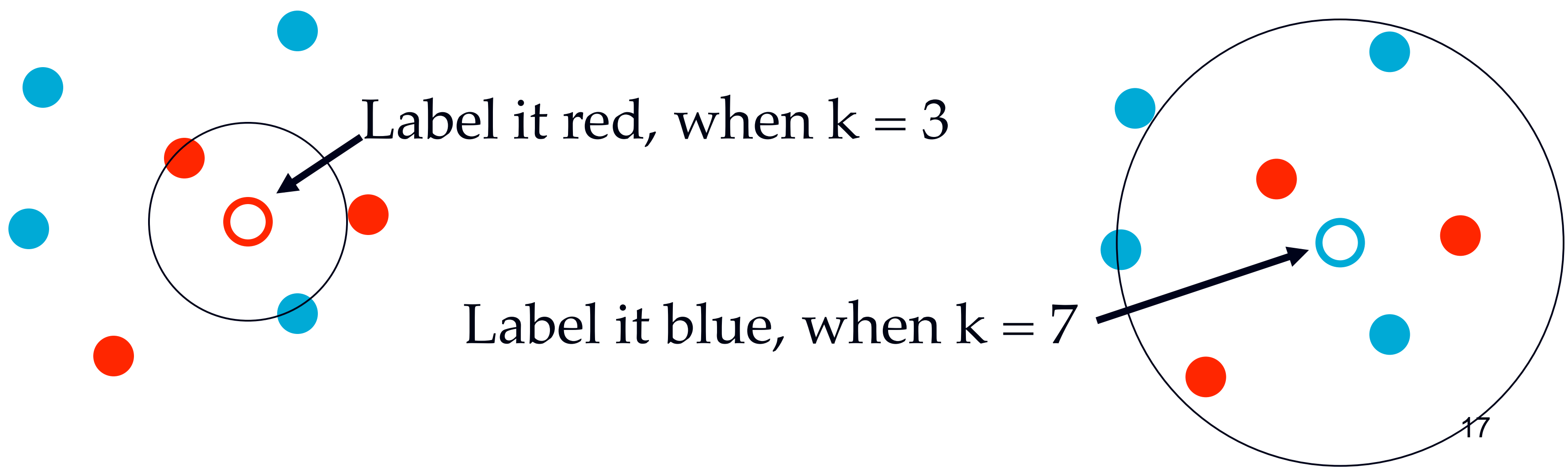


Layout shows the rock clusters (top) and the preferred viewing spot of the garden from the main hall (the circle in the middle of the square).

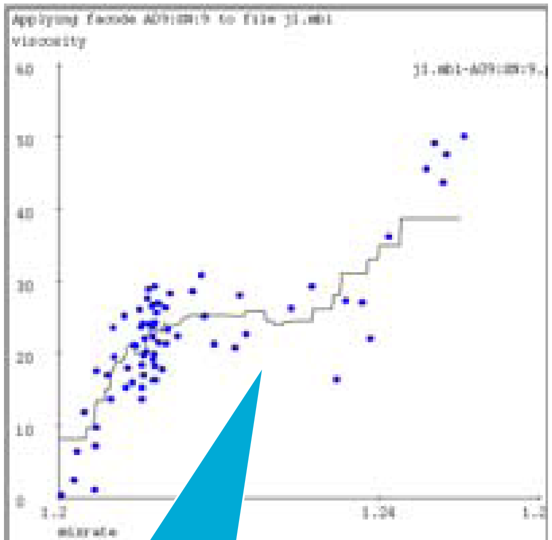
Adapted from "Instance-Based Learning" lecture slides by Andrew Moore, CMU.

k – Nearest Neighbor

- Generalizes 1-NN to smooth away noise in the labels
- A new point is now assigned the most frequent label of its *k* nearest neighbors

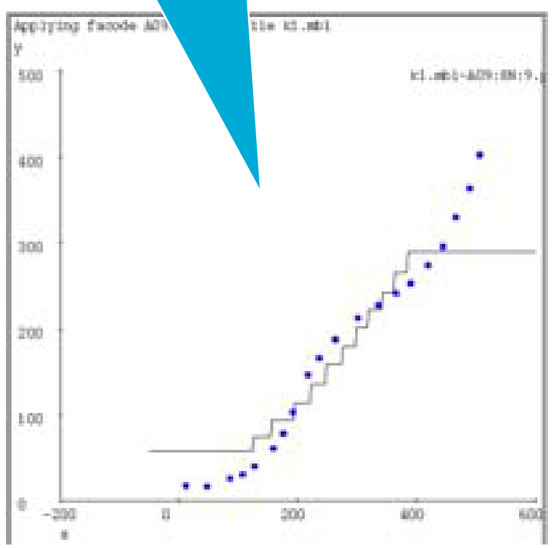


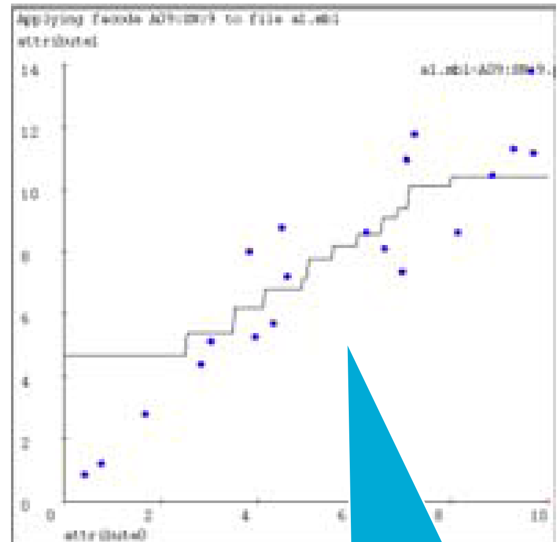
k-Nearest Neighbor (k = 9)



A magnificent job of noise smoothing. Three cheers for 9-nearest-neighbor. But the lack of gradients and the jerkiness isn't good.

Appalling behavior! Loses all the detail that 1-nearest neighbor would give. The tails are horrible!





Fits much less of the noise, captures trends. But still, frankly, pathetic compared with linear regression.

Adapted from "Instance-Based Learning" lecture slides by Andrew Moore, CMU.

The Naïve Bayes Classifier

Some material adapted from slides by
Tom Mitchell, CMU.

The Naïve Bayes Classifier

- Recall Bayes rule:

$$P(Y_i \mid X_j) = \frac{P(Y_i)P(X_j \mid Y_i)}{P(X_j)}$$

- Which is short for:

$$P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i)P(X = x_j \mid Y = y_i)}{P(X = x_j)}$$

- We can re-write this as:

$$P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i)P(X = x_j \mid Y = y_i)}{\sum_k P(X = x_j \mid Y = y_k)P(Y = y_k)}$$

Deriving Naïve Bayes

- Idea: use the training data to directly estimate:

$$P(X \mid Y) \quad \text{and} \quad P(Y)$$

- Then, we can use these values to estimate $P(Y \mid X_{new})$ using Bayes rule.

- Recall that representing the full joint probability $P(X_1, X_2, \ldots, X_n \mid Y)$ is not practical.

Deriving Naïve Bayes

- However, if we make the assumption that the attributes are independent, estimation is easy!

$$P(X_1,\ldots,X_n \mid Y) = \prod_i P(X_i \mid Y)$$

- In other words, we assume all attributes are conditionally independent given Y.
  - Often this assumption is violated in practice, but more on that later…

Deriving Naïve Bayes

- Let $X = \langle X_1, \ldots, X_n \rangle$ and label Y be discrete.

- Then, we can estimate $P(X_i \mid Y_i)$ and $P(Y_i)$ directly from the training data by counting!

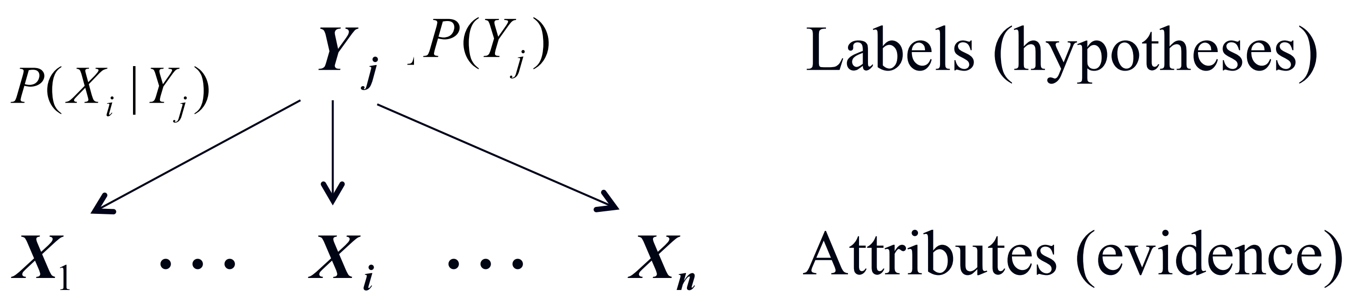
Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

The Naïve Bayes Classifier

- Now we have:

$$P(Y = y_j \mid X_1, \ldots, X_n) = \frac{P(Y = y_j) \prod_i P(X_i \mid Y = y_j)}{\sum_k P(Y = y_k) \prod_i P(X_i \mid Y = y_k)}$$

which is just a one-level Bayesian Network



- To classify a new point X_{new}:

$$Y_{new} \leftarrow \arg\max_{y_k} P(Y = y_k) \prod_i P(X_i \mid Y = y_k)$$

The Naïve Bayes Algorithm

- For each value y_k
  - Estimate $P(Y = y_k)$ from the data.
  - For each value x_{ij} of each attribute X_i
    - Estimate $P(X_i=x_{ij} \mid Y = y_k)$
- Classify a new point via:

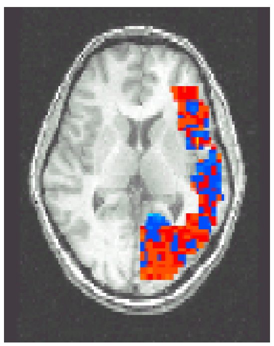
$$Y_{new} \leftarrow \arg\max_{y_k} P(Y = y_k) \prod_i P(X_i \mid Y = y_k)$$

- In practice, the independence assumption doesn't often hold true, but Naïve Bayes performs very well despite it.

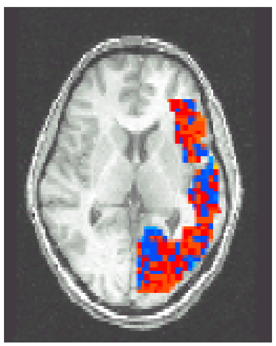
Naïve Bayes Applications

- Text classification
  - Which e-mails are spam?
  - Which e-mails are meeting notices?
  - Which author wrote a document?
- Classifying mental states

Learning P(BrainActivity | WordCategory)



People Words



Animal Words

Pairwise Classification
Accuracy: 85%

Polynomial Curve Fitting

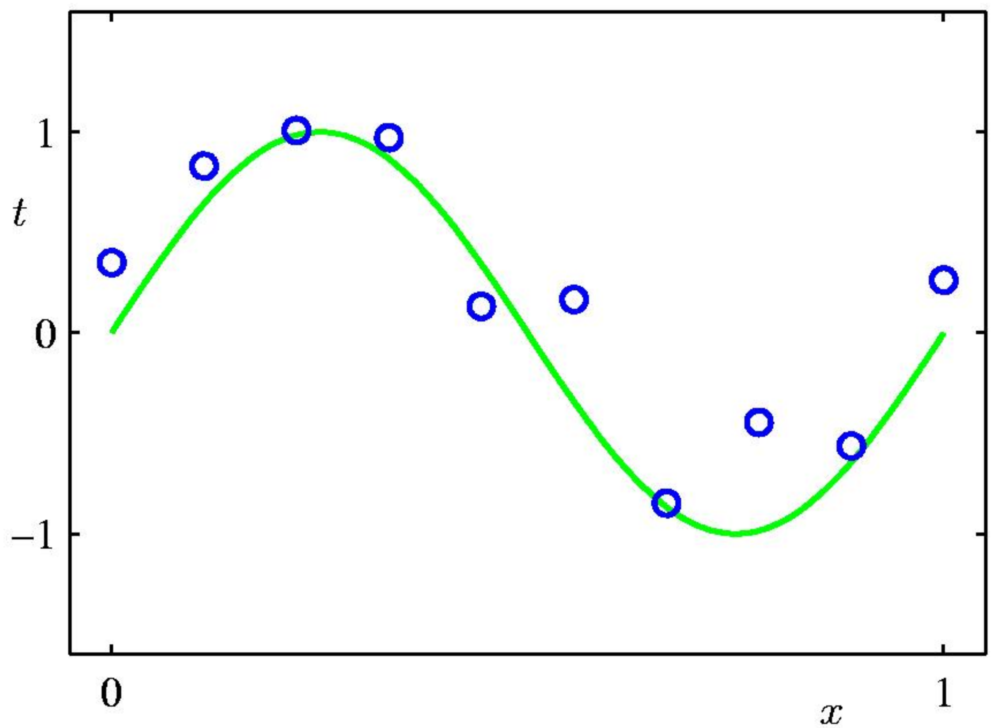
Slides adapted from

Pattern Recognition and

Machine Learning

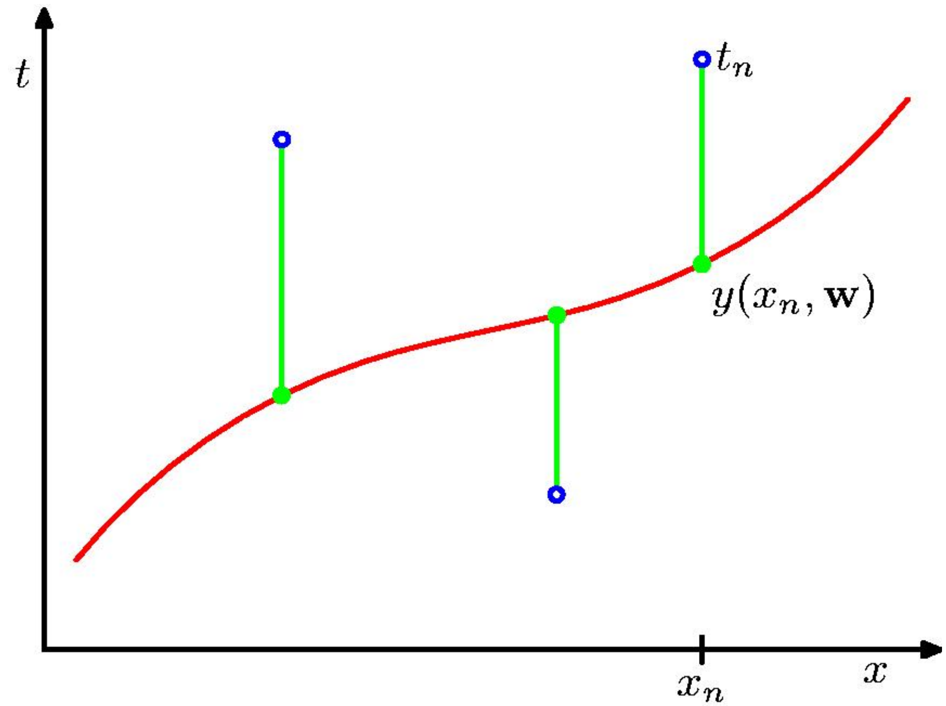
by Christopher Bishop

Polynomial Curve Fitting



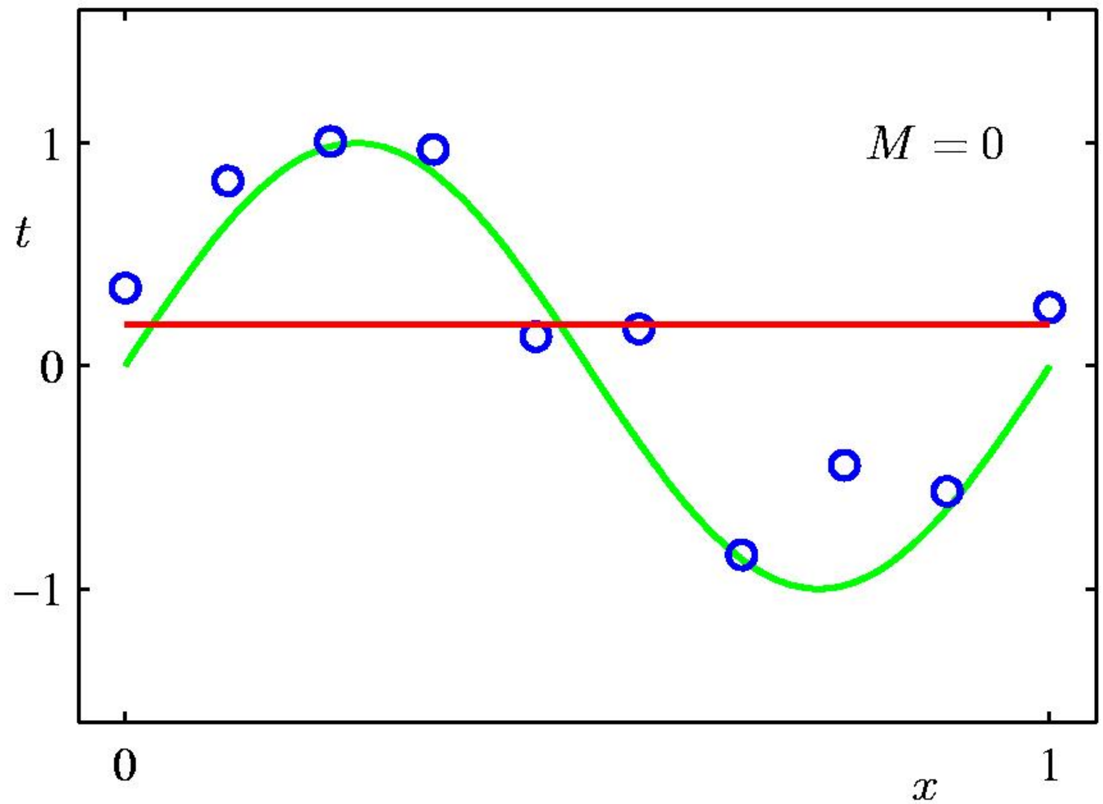
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function

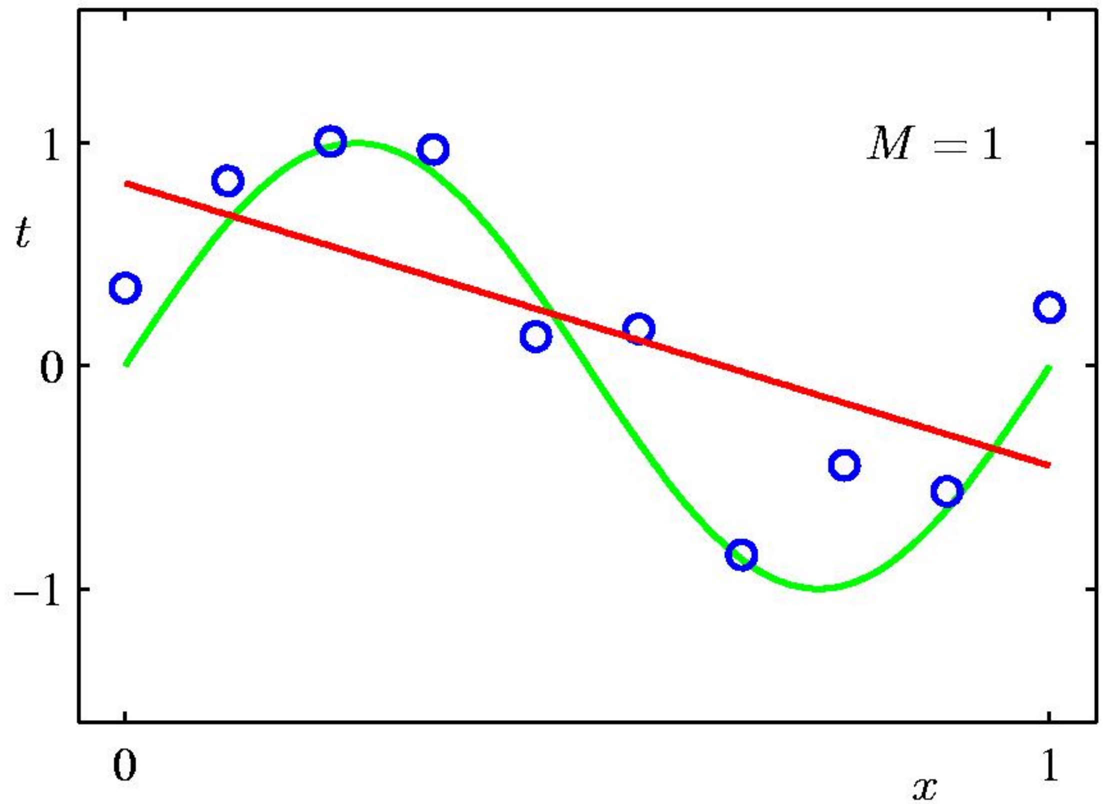


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

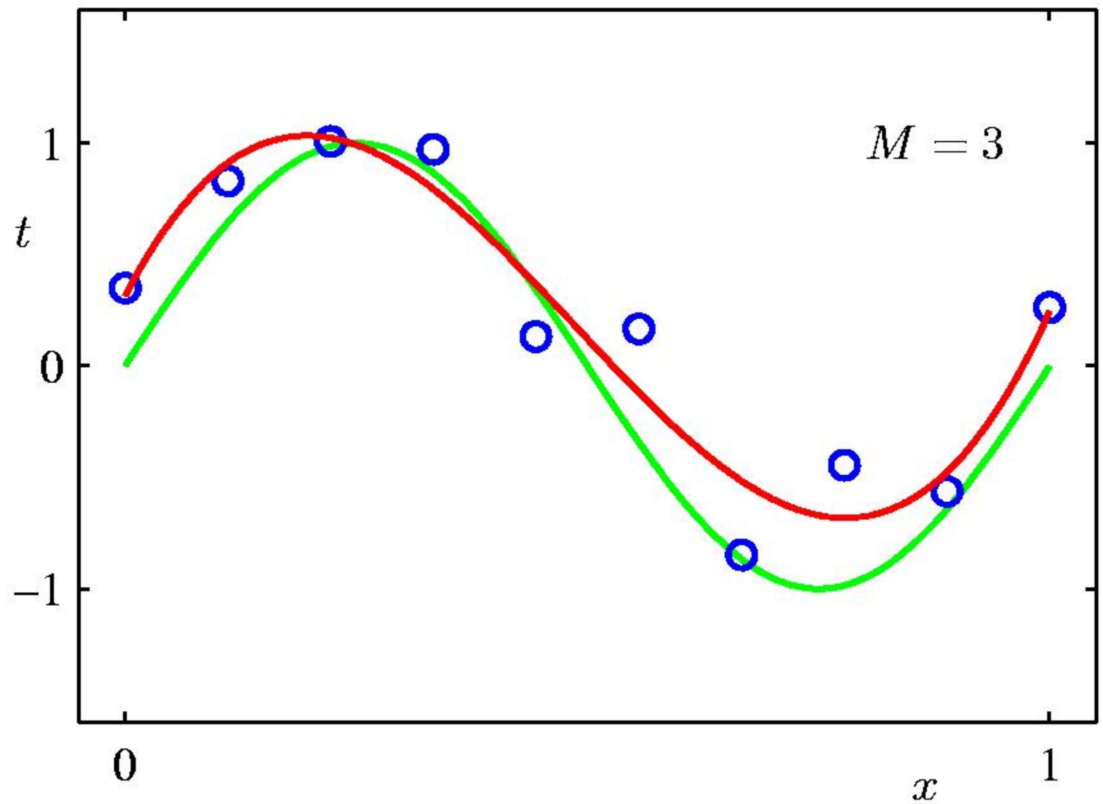
0^{th} Order Polynomial



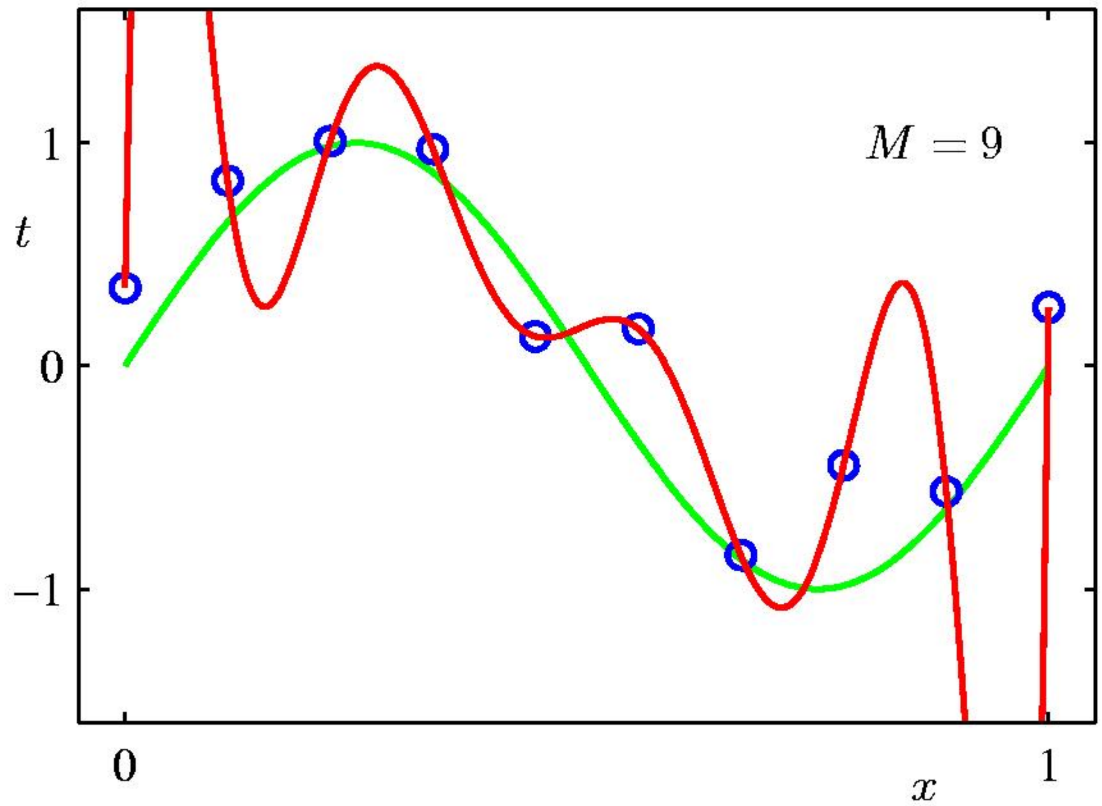
1st Order Polynomial



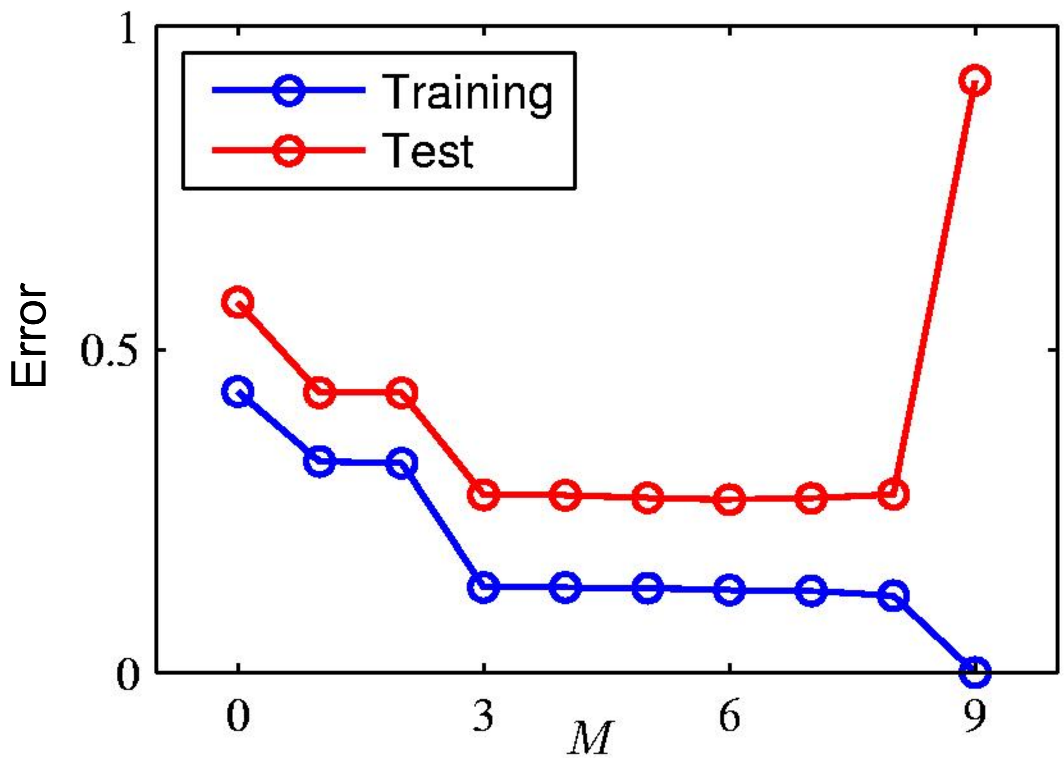
3rd Order Polynomial



9th Order Polynomial



Over-fitting

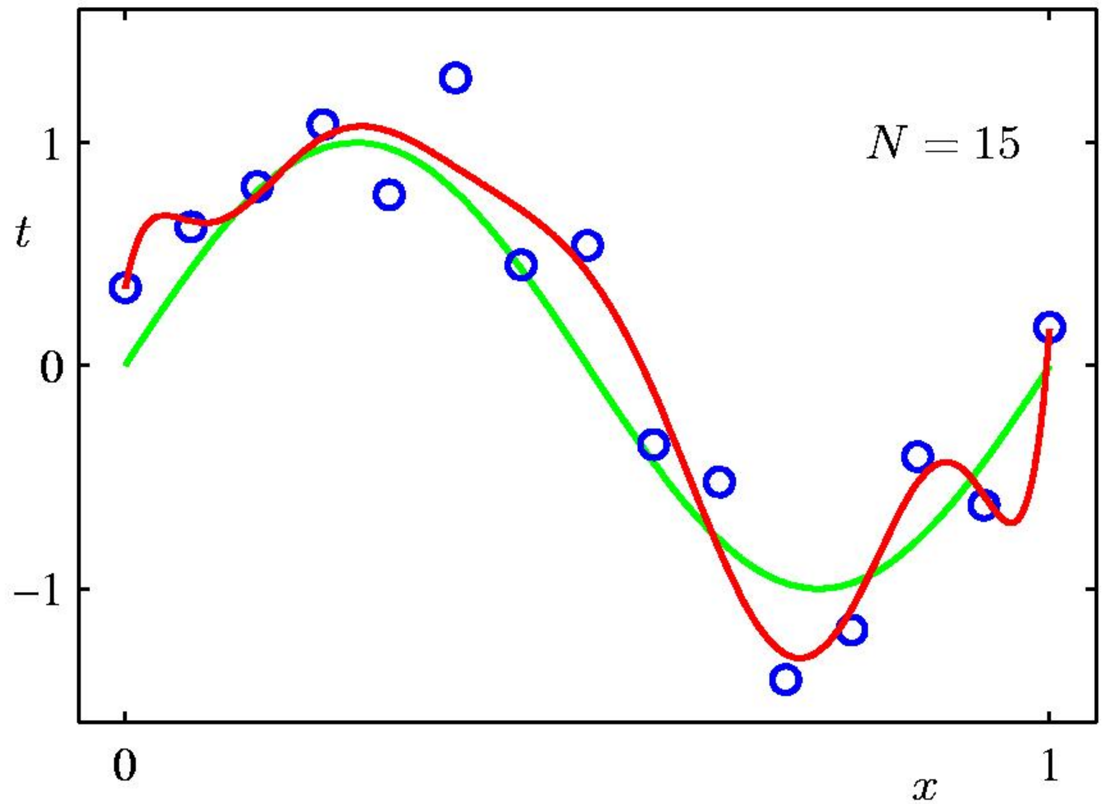


Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

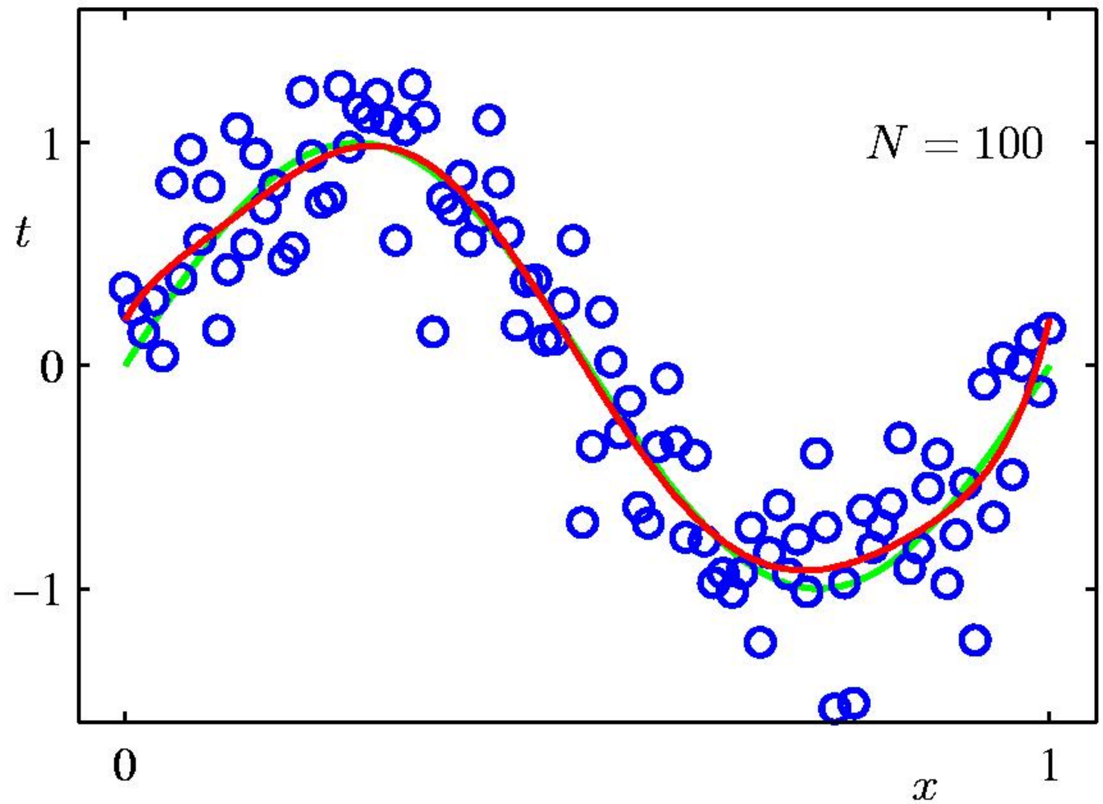
Data Set Size: $N = 15$

9th Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial



Regularization

Penalize large coefficient values

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\left(\|\mathbf{w}\|_2\right)^2$$

L_2 Norm

$$\|\mathbf{w}\|_2 = \sqrt{\sum_i \mathrm{w}_i^2}$$

Measures the "complexity" of w

Regularization

λ regularization parameter
higher λ → more regularization

Penalize large coefficient values

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\left(\|\mathbf{w}\|_2\right)^2$$

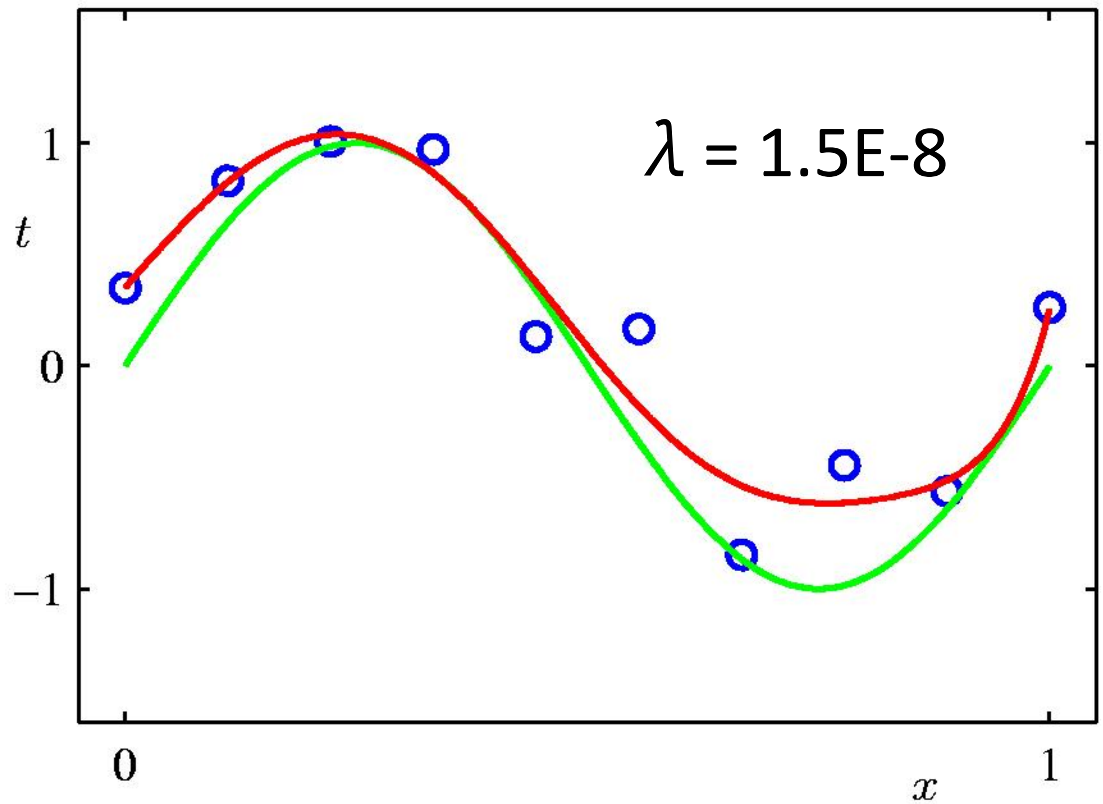
$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\sum_{i}\mathrm{w}_i^2$$

L_2 Norm

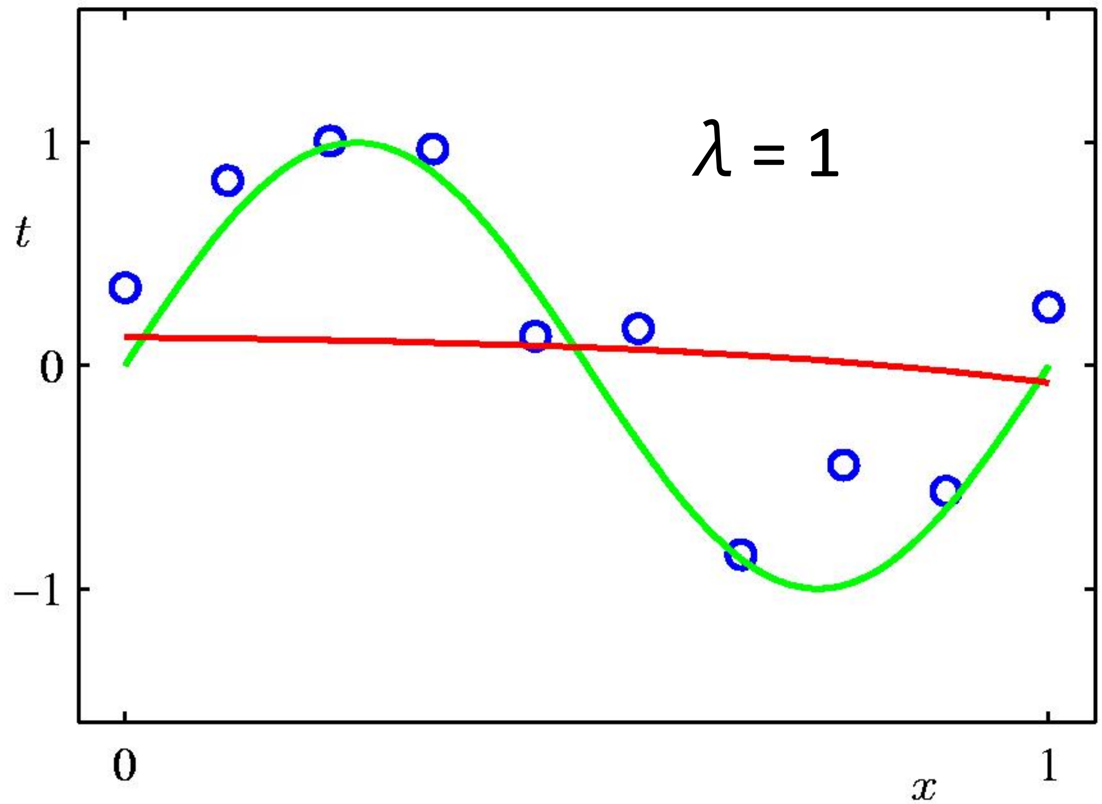
$$\|\mathbf{w}\|_2 = \sqrt{\sum_i \mathrm{w}_i^2}$$

Measures the "complexity" of w

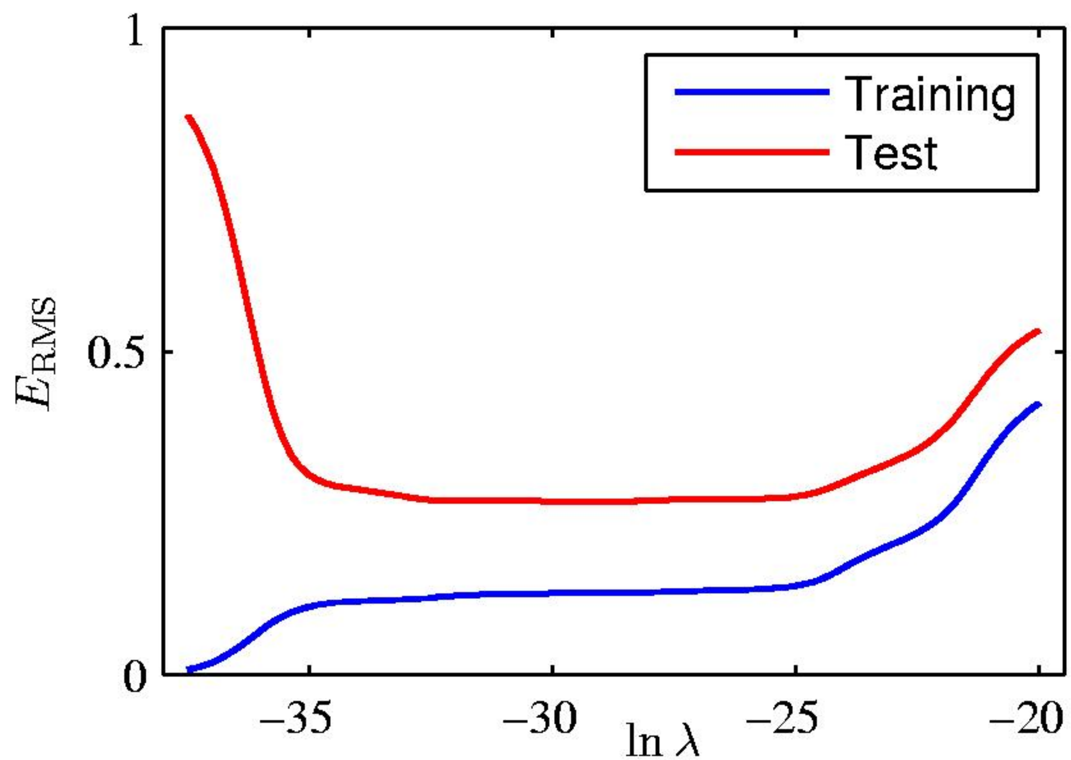
Regularization: λ = 1.5E-8



Regularization: $\lambda = 1$



Regularization: E_{RMS} vs. $\ln \lambda$



Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Learning via Gradient Descent

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\Big(y(x_n,\mathbf{w}) - t_n\Big)^2 + \frac{\lambda}{2}\sum_j \mathrm{w}_j^2$$

$$\nabla_j \tilde{E}(\mathbf{w}) = \sum_{n=1}^{N} x_n^j \Big(y(x_n,\mathbf{w}) - t_n\Big) + \lambda \mathrm{w}_j$$

Choose $\mathbf{w}$ randomly, where $\mathrm{w}_j \sim N(0,\sigma^2)$

Repeat until $\mathbf{w}$ converges (i.e., $||\mathbf{w} - \mathbf{w}_{old}|| < \varepsilon$)

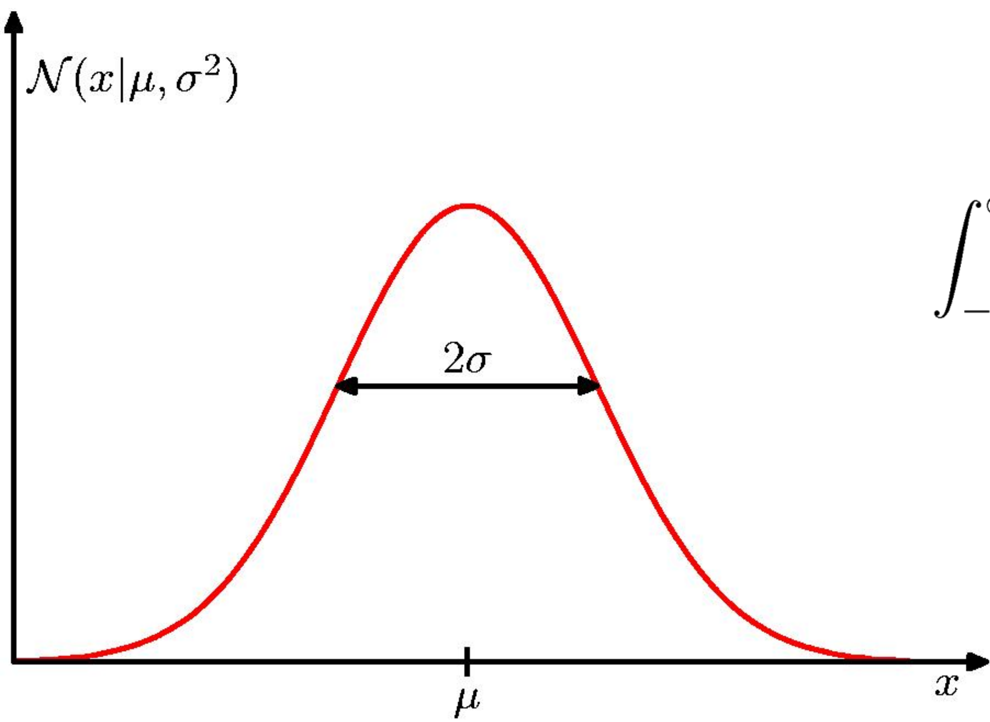
$\mathbf{w}_{old} = \mathbf{w}$

For j = 0 ... M:

$$\mathrm{w}_j = \mathrm{w}_j - \alpha \nabla_j \tilde{E}(\mathbf{w})$$

The Gaussian Distribution

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

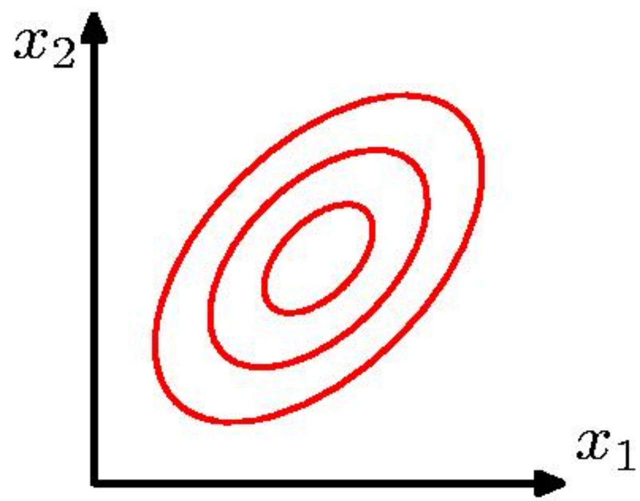


$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

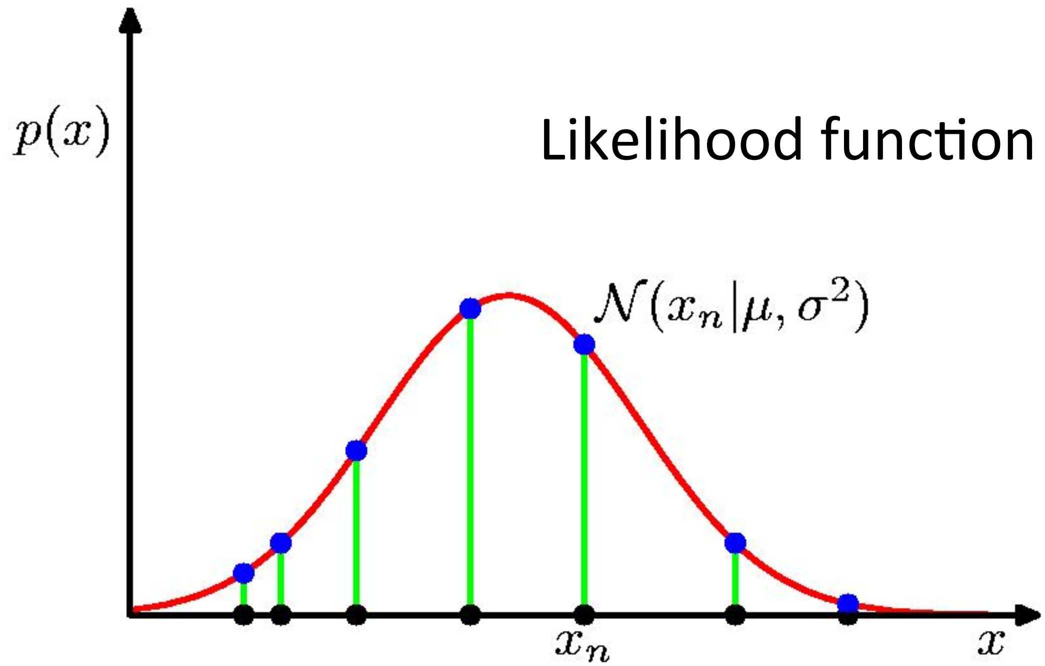
$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

The Multivariate Gaussian

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\}$$



Gaussian Parameter Estimation



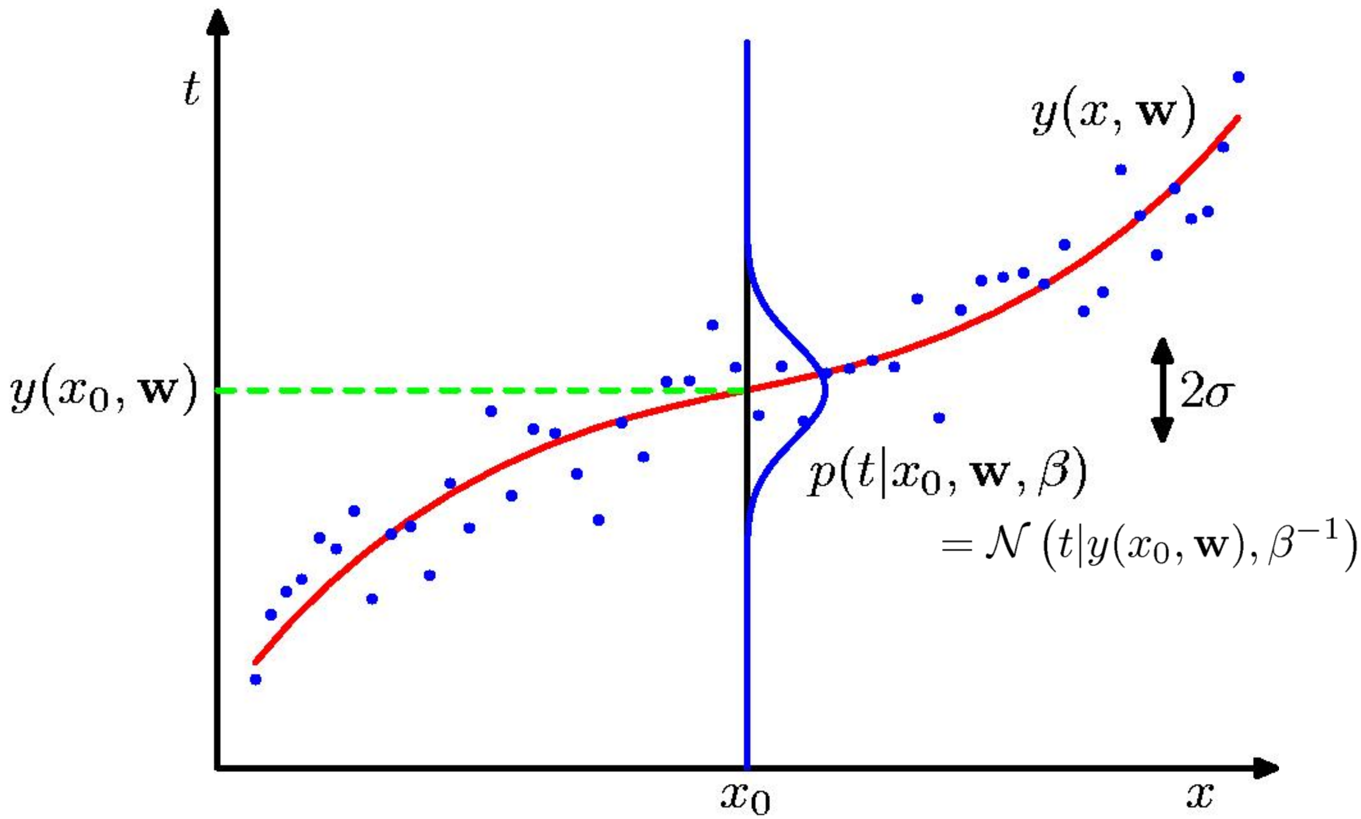
$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu, \sigma^2\right)$$

Maximum (Log) Likelihood

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{\mathrm{ML}})^2$$

Curve Fitting Re-visited



Maximum Likelihood

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n | y(x_n, \mathbf{w}), \beta^{-1}\right)$$

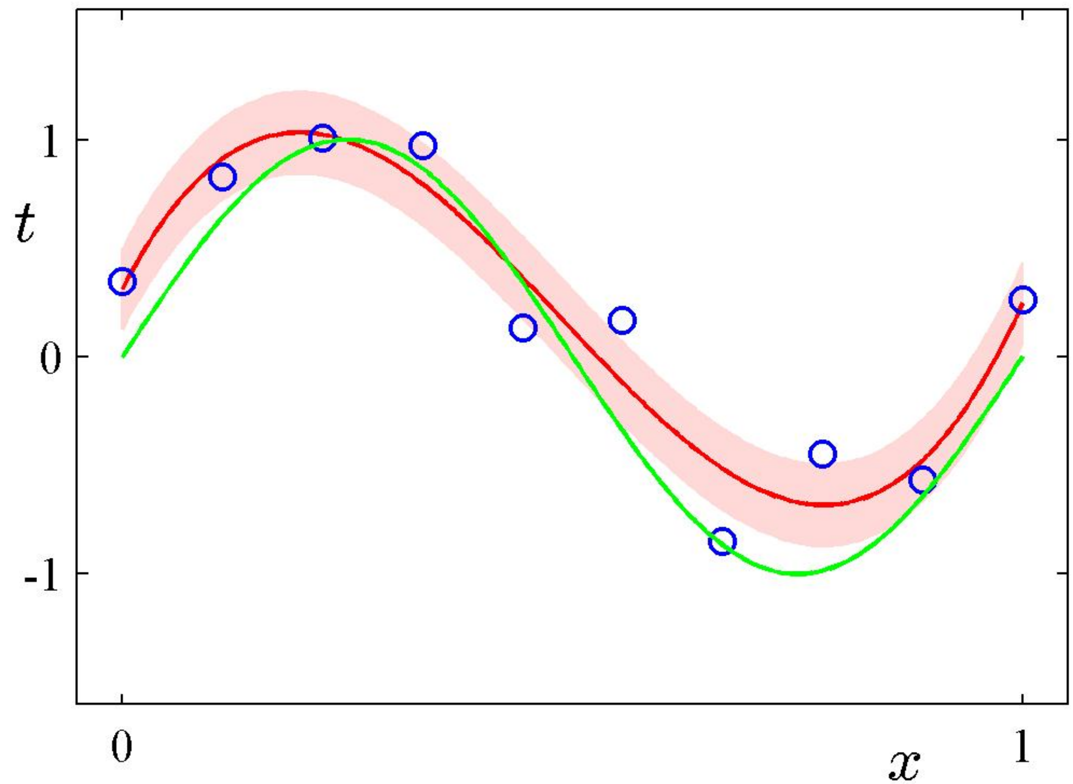
$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \underbrace{-\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2}_{\beta E(\mathbf{w})} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

Determine $\mathbf{w}_{\mathrm{ML}}$ by minimizing sum-of-squares error, $E(\mathbf{w})$.

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}_{\mathrm{ML}}) - t_n\}^2$$

Predictive Distribution

$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right)$$



MAP: A Step towards Bayes

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

$$\beta\widetilde{E}(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Determine $\mathbf{w}_{\mathrm{MAP}}$ by minimizing regularized sum-of-squares error, $\widetilde{E}(\mathbf{w})$.

Bayesian Curve Fitting

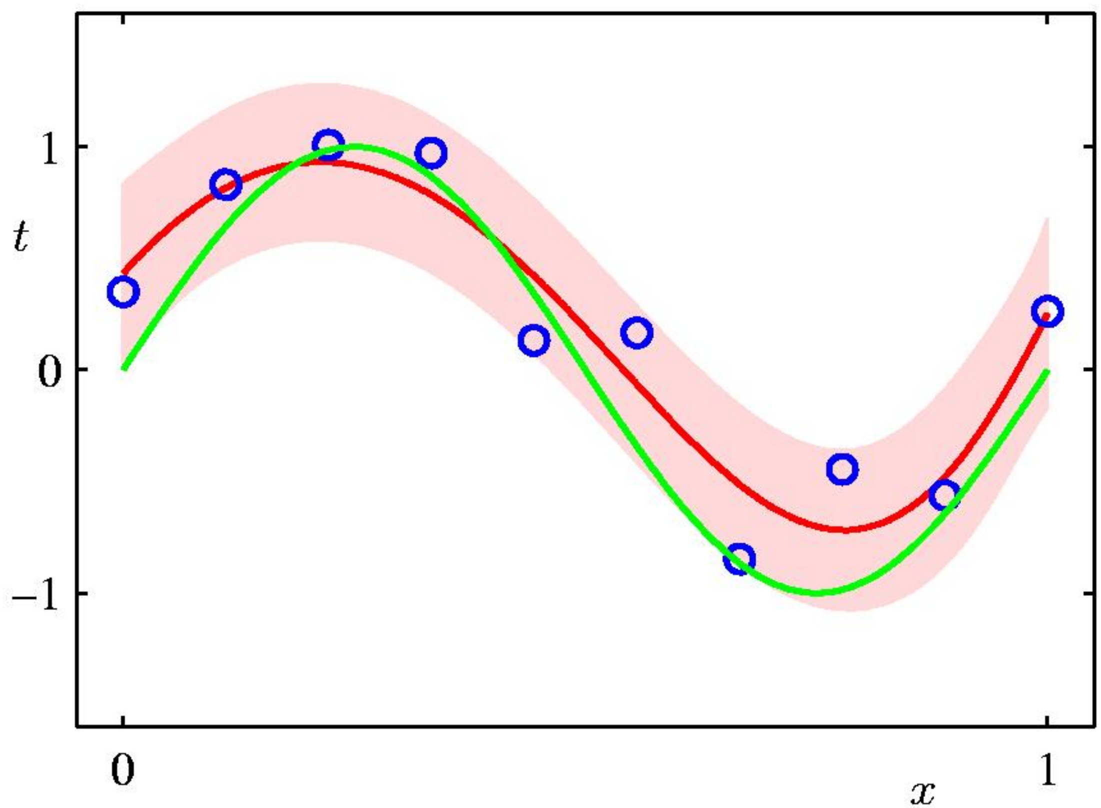
$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w})p(\mathbf{w}|\mathbf{x}, \mathbf{t})\,\mathrm{d}\mathbf{w} = \mathcal{N}\left(t|m(x), s^2(x)\right)$$

$$m(x) = \beta\boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)t_n \qquad s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\boldsymbol{\phi}(x)$$

$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^{\mathrm{T}} \qquad \boldsymbol{\phi}(x_n) = \left(x_n^0, \ldots, x_n^M\right)^{\mathrm{T}}$$

Bayesian Predictive Distribution

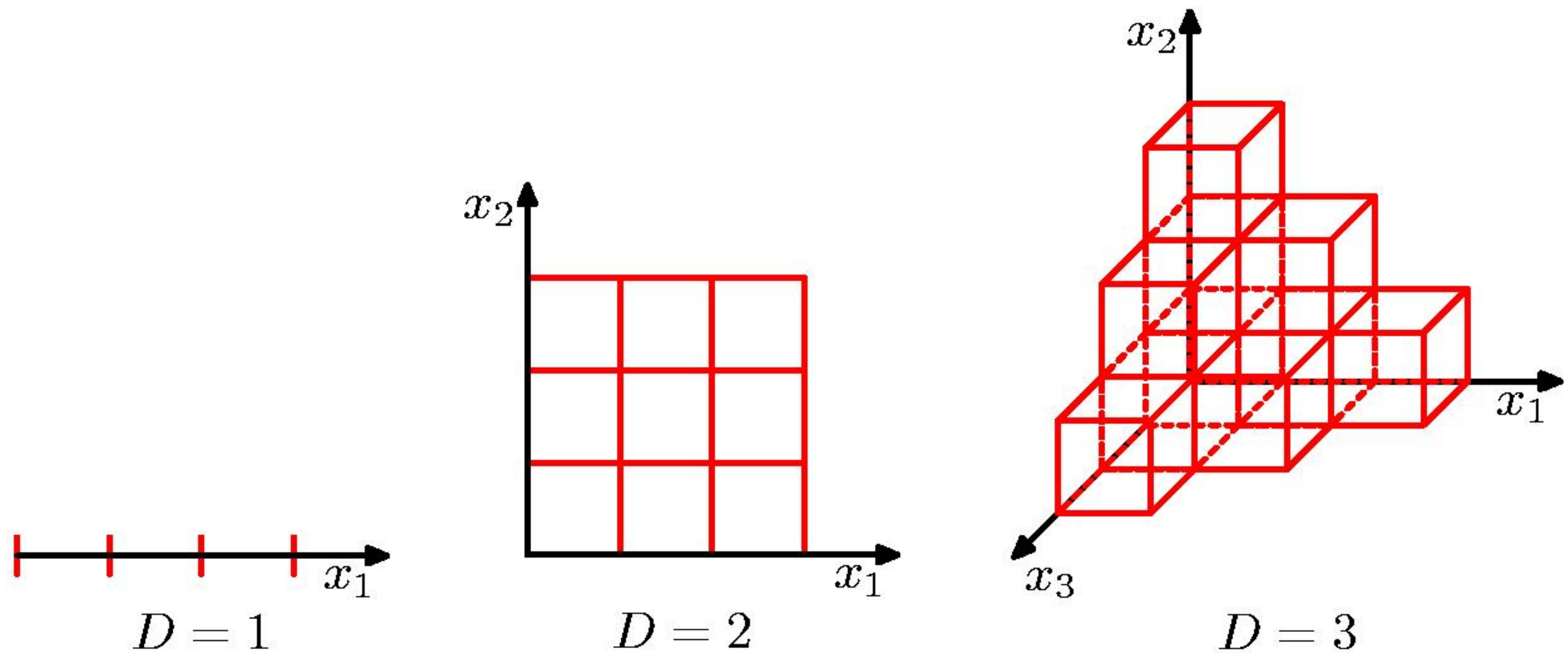
$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right)$$



Model Selection

Cross-Validation

Curse of Dimensionality



Curse of Dimensionality

Polynomial curve fitting, M $= 3$

$$y(\mathbf{x},\mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

Gaussian Densities in higher dimensions