
MACHINE LEARNING

CHAPTER 1: PRELIMINARY

Learning Objectives

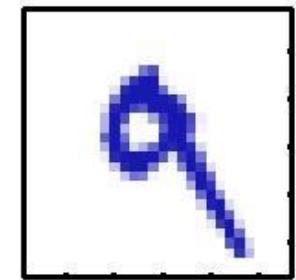
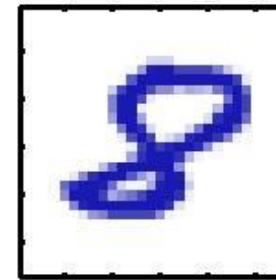
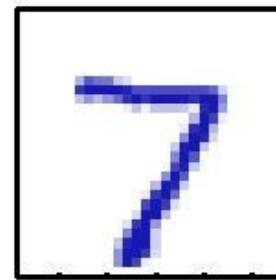
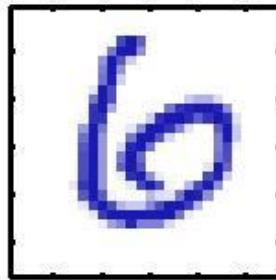
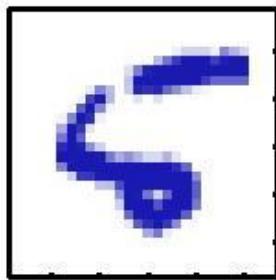
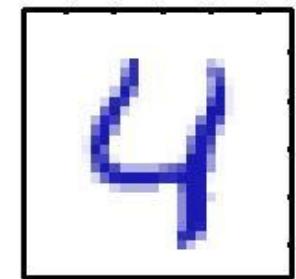
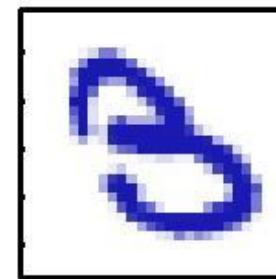
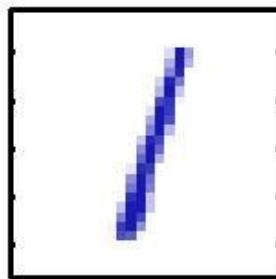
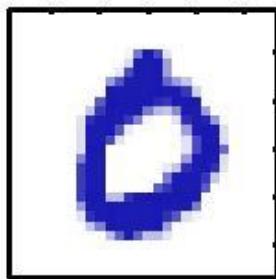
- 1、 What is pattern recognition?
 - 2、 What are curve fitting and regularization?
 - 3、 What are ML and MAP Bayesian inferences?
 - 4、 How to deal with the curse of dimensionality?
 - 5、 What is the relationship between decision theory and machine learning?
 - 6、 What are generative and discriminative models?
 - 7、 How to use entropy、KL divergence and mutual information for machine learning?
-

Outlines

- Pattern Recognition
 - Curve Fitting and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theory
 - Entropy and Information
-

Example

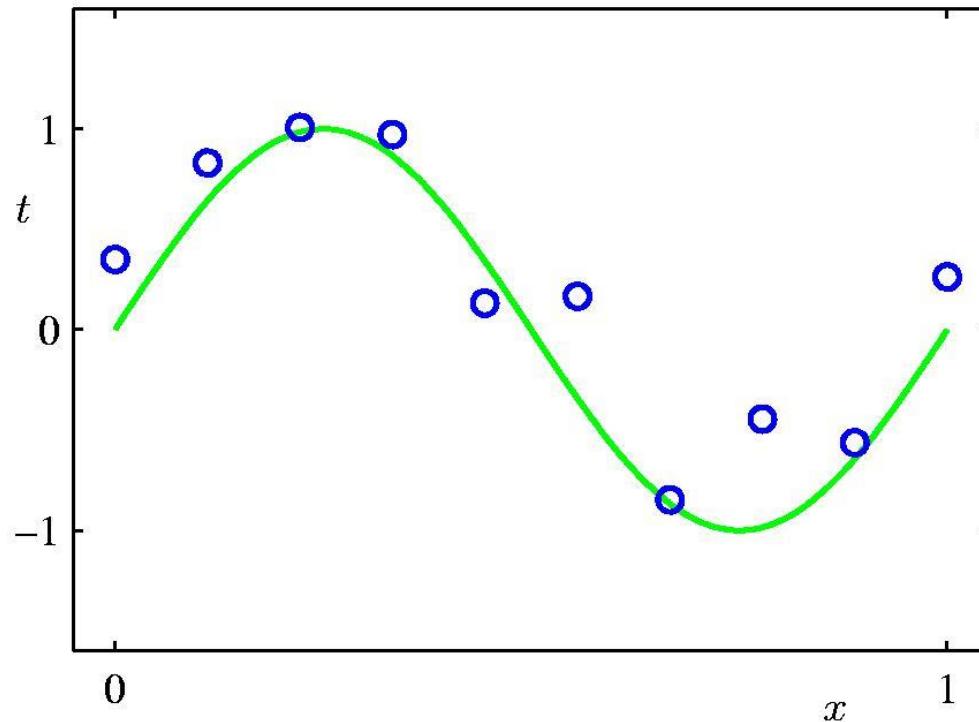
Handwritten Digit Recognition



Outlines

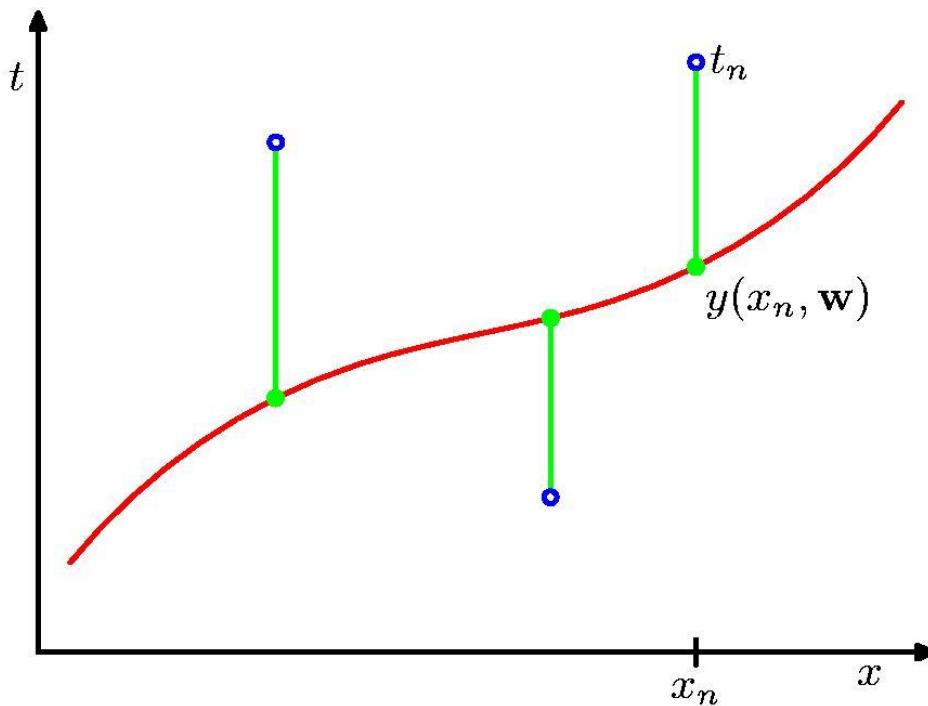
- Pattern Recognition
 - Curve Fitting and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theory
 - Entropy and Information
-

Polynomial Curve Fitting



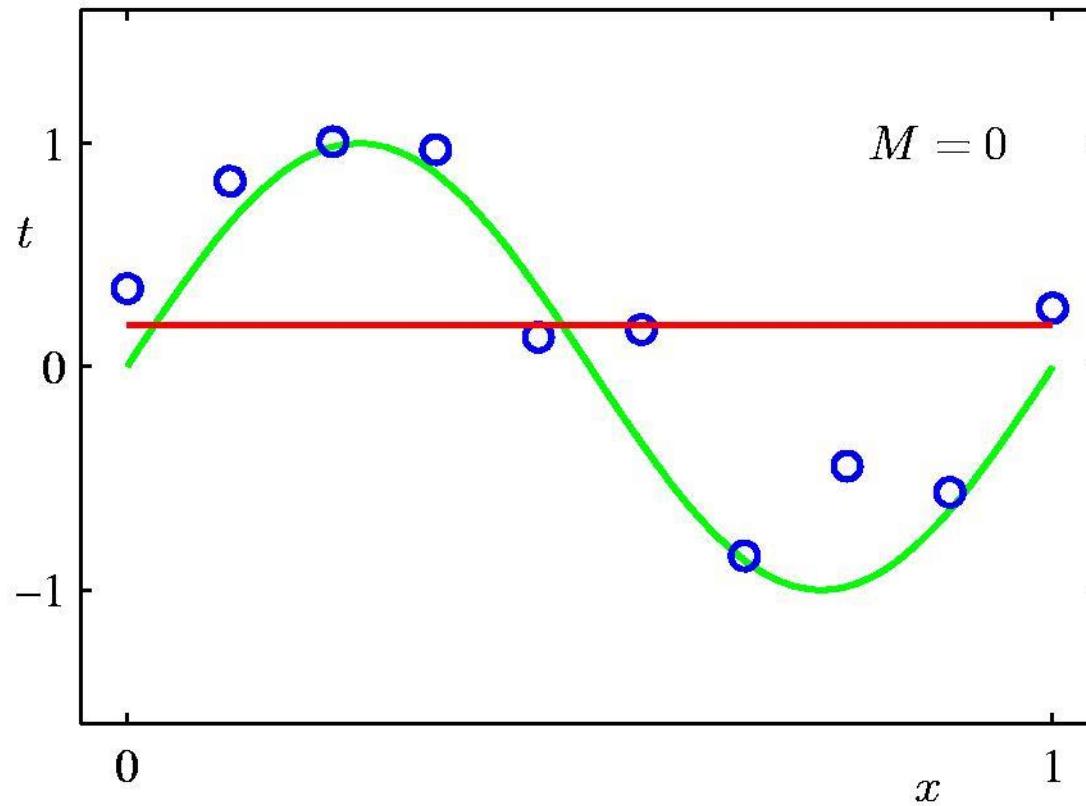
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j$$

Sum-of-Squares Error Function

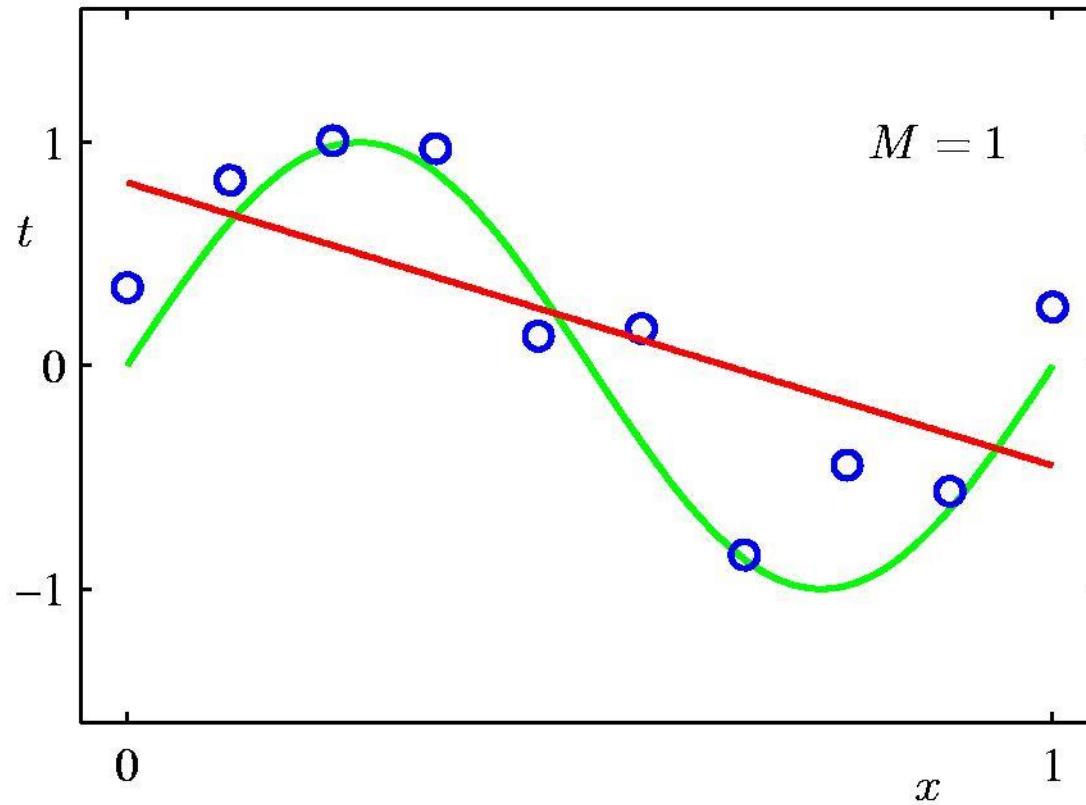


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

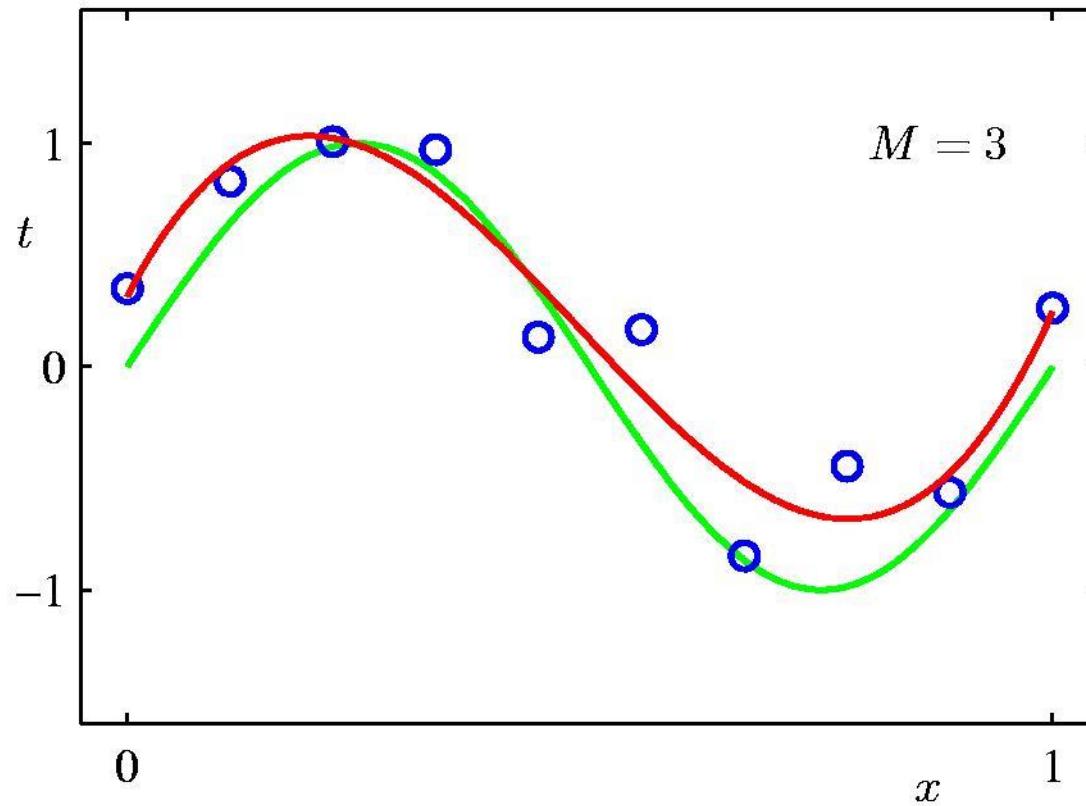
0th Order Polynomial



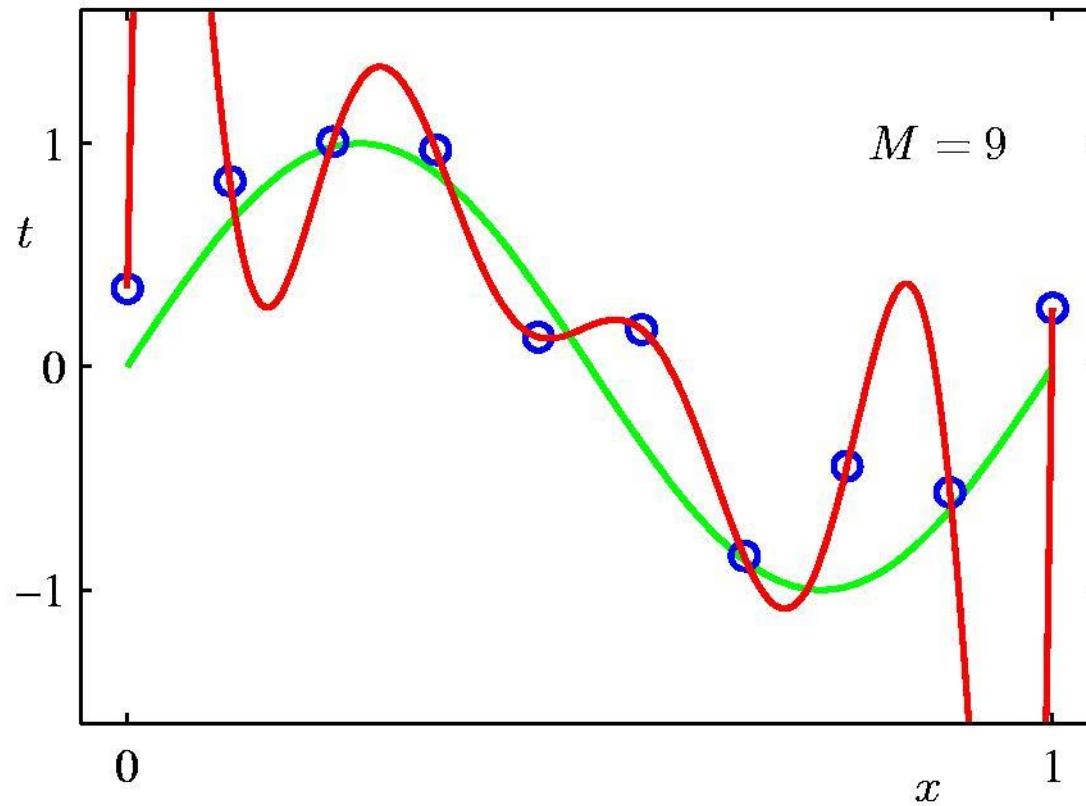
1st Order Polynomial



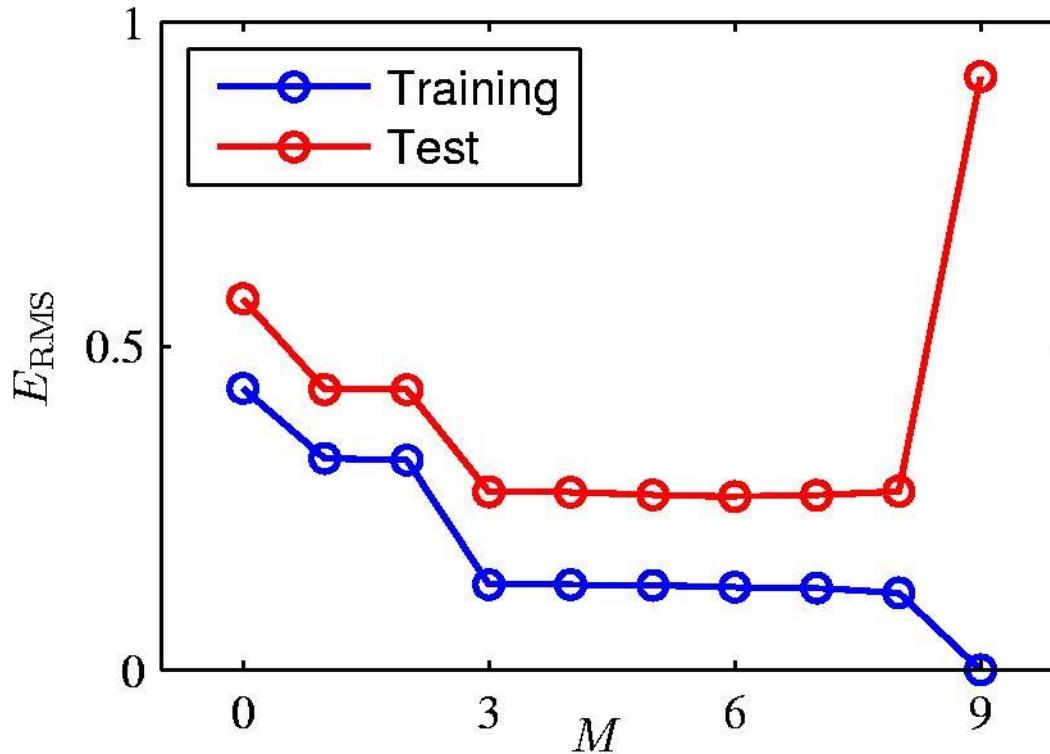
3rd Order Polynomial



9th Order Polynomial



Over-fitting



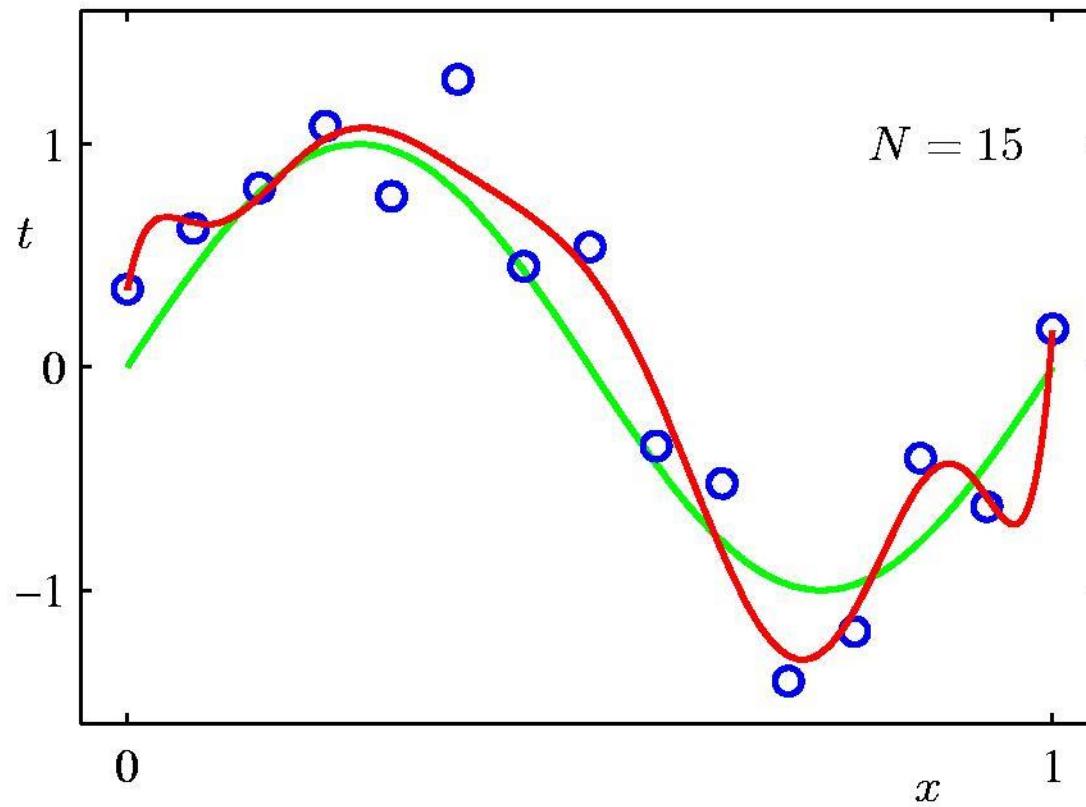
Root-Mean-Square (RMS) Error: $E_{\text{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

Polynomial Coefficients

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

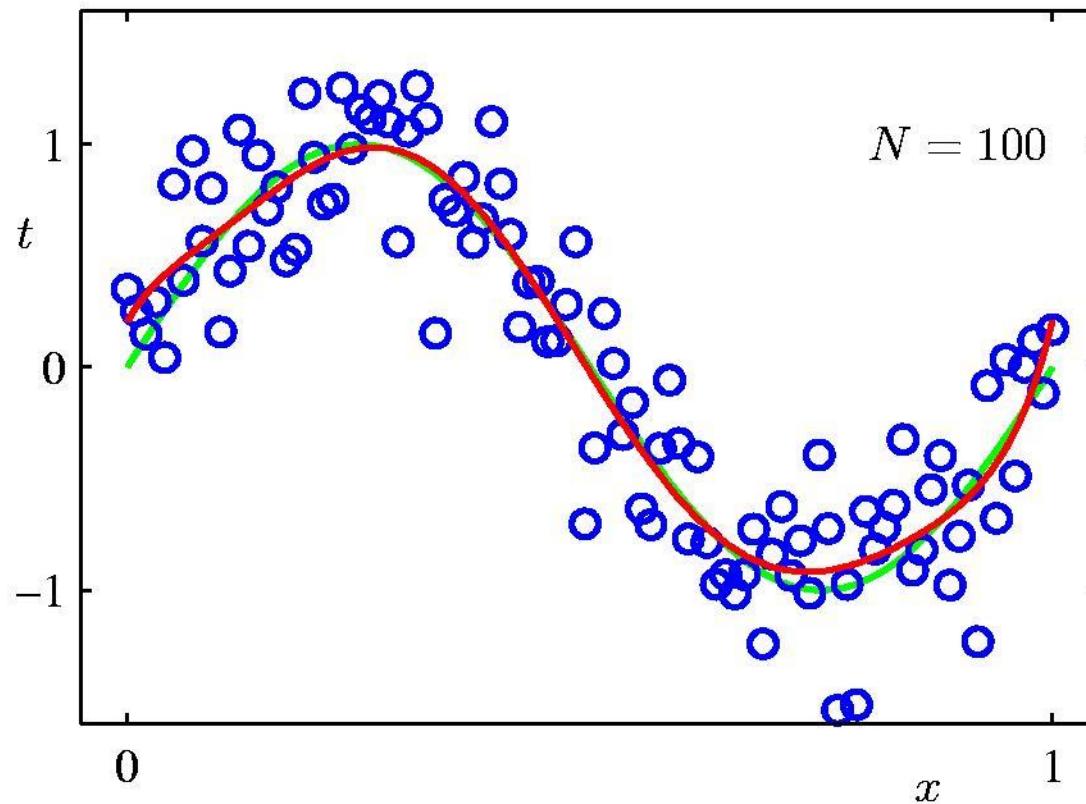
Data Set Size: $N = 15$

9th Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial

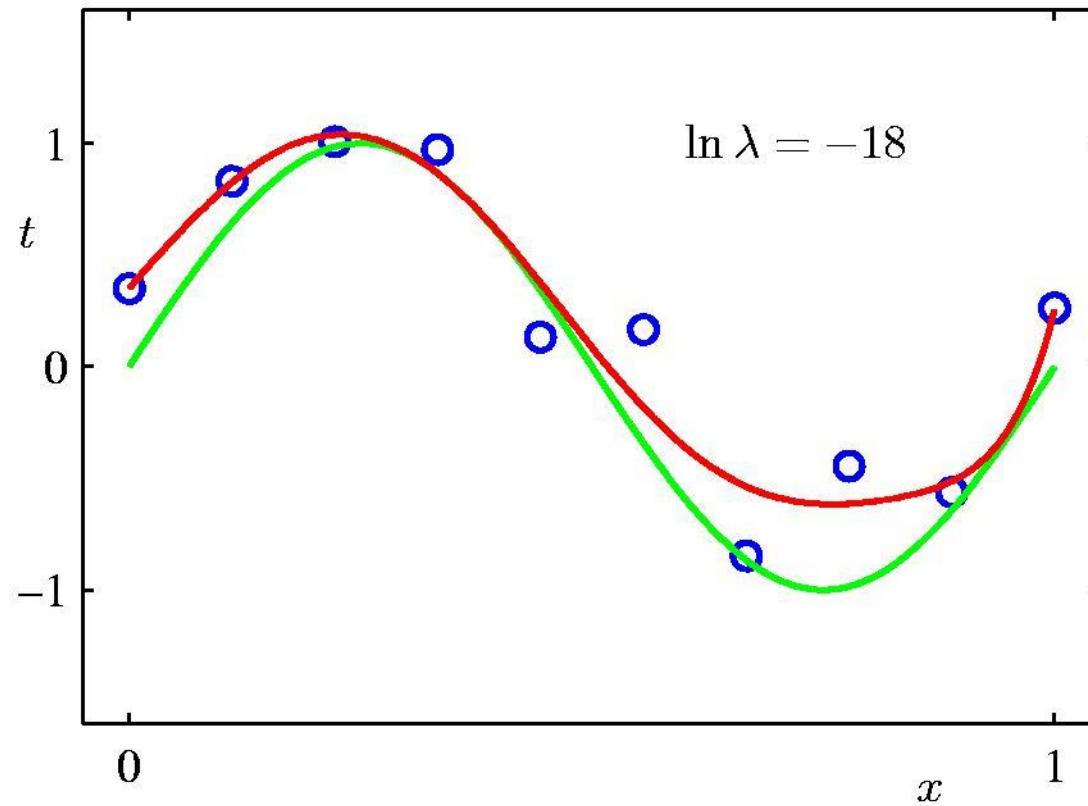


Regularization

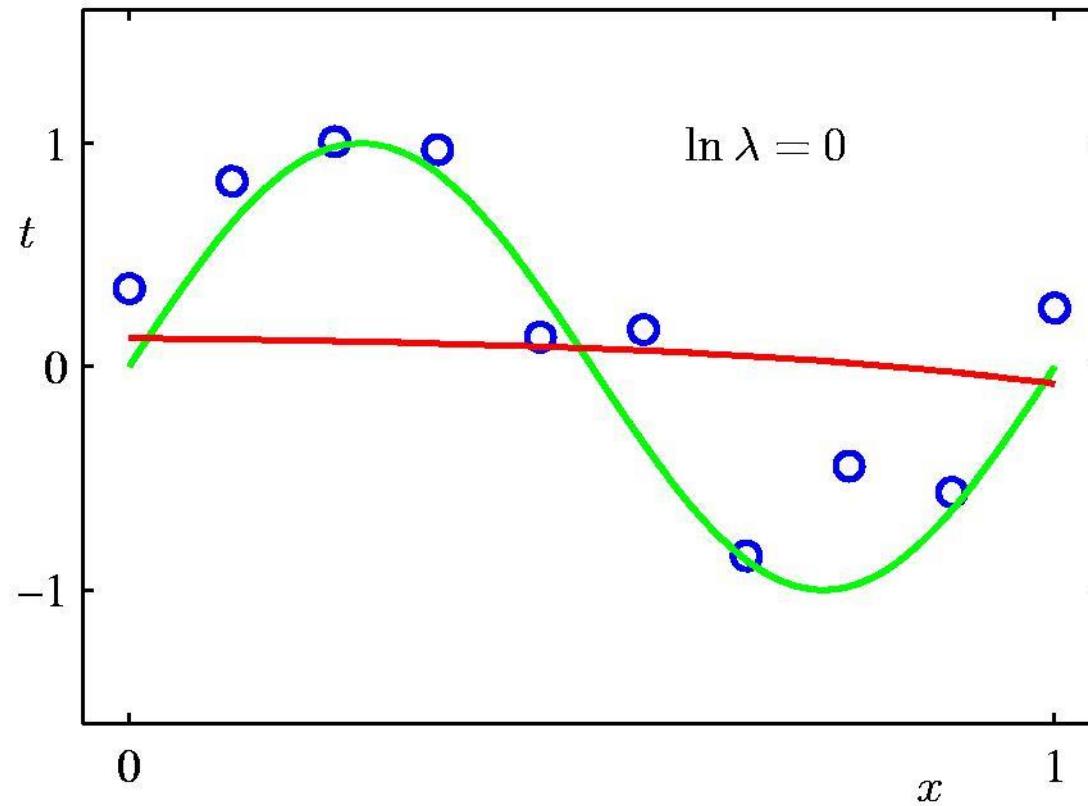
Penalize large coefficient values

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

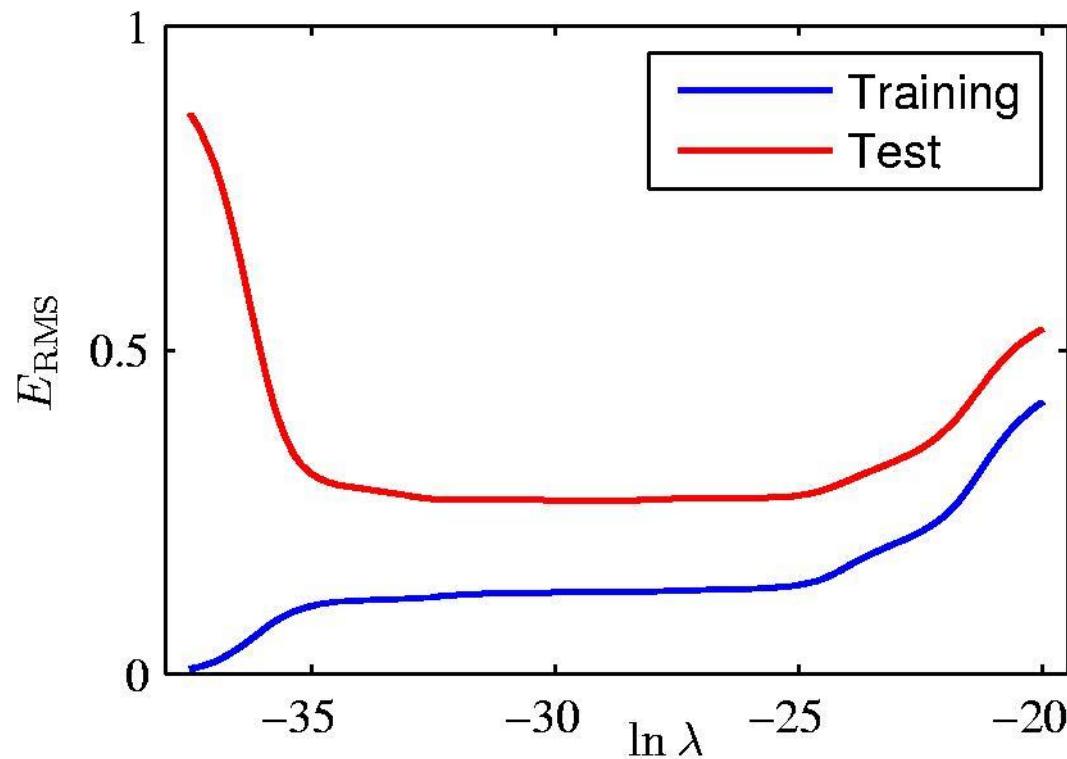
Regularization: $\ln \lambda = -18$



Regularization: $\ln \lambda = 0$



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



Polynomial Coefficients

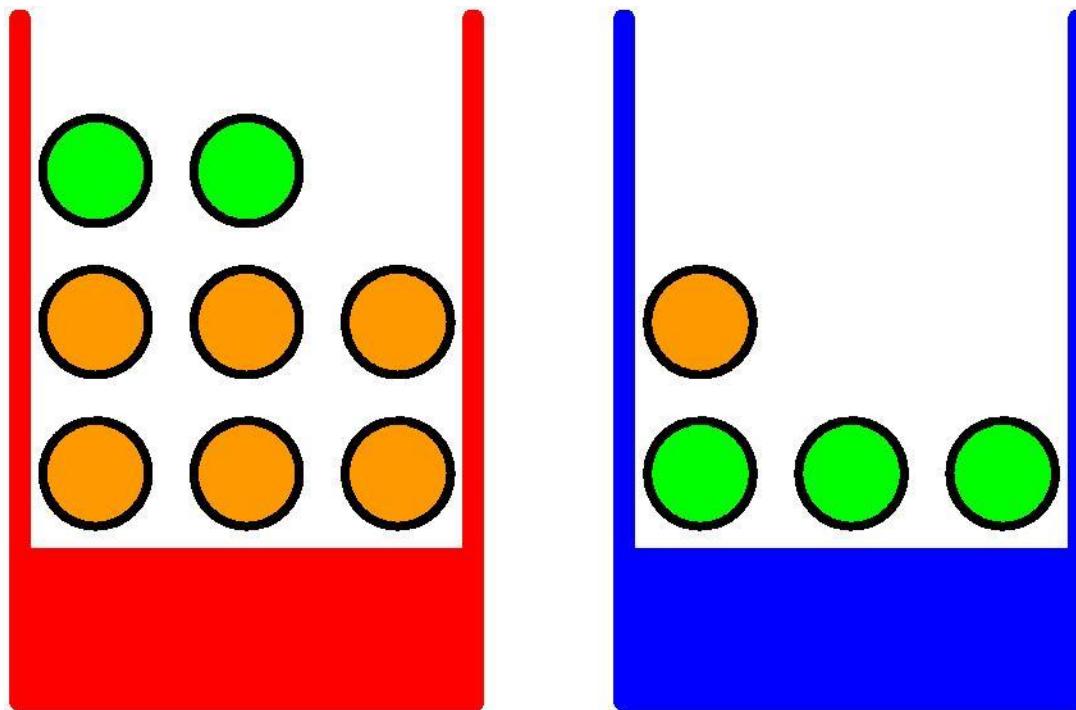
	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53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

Outlines

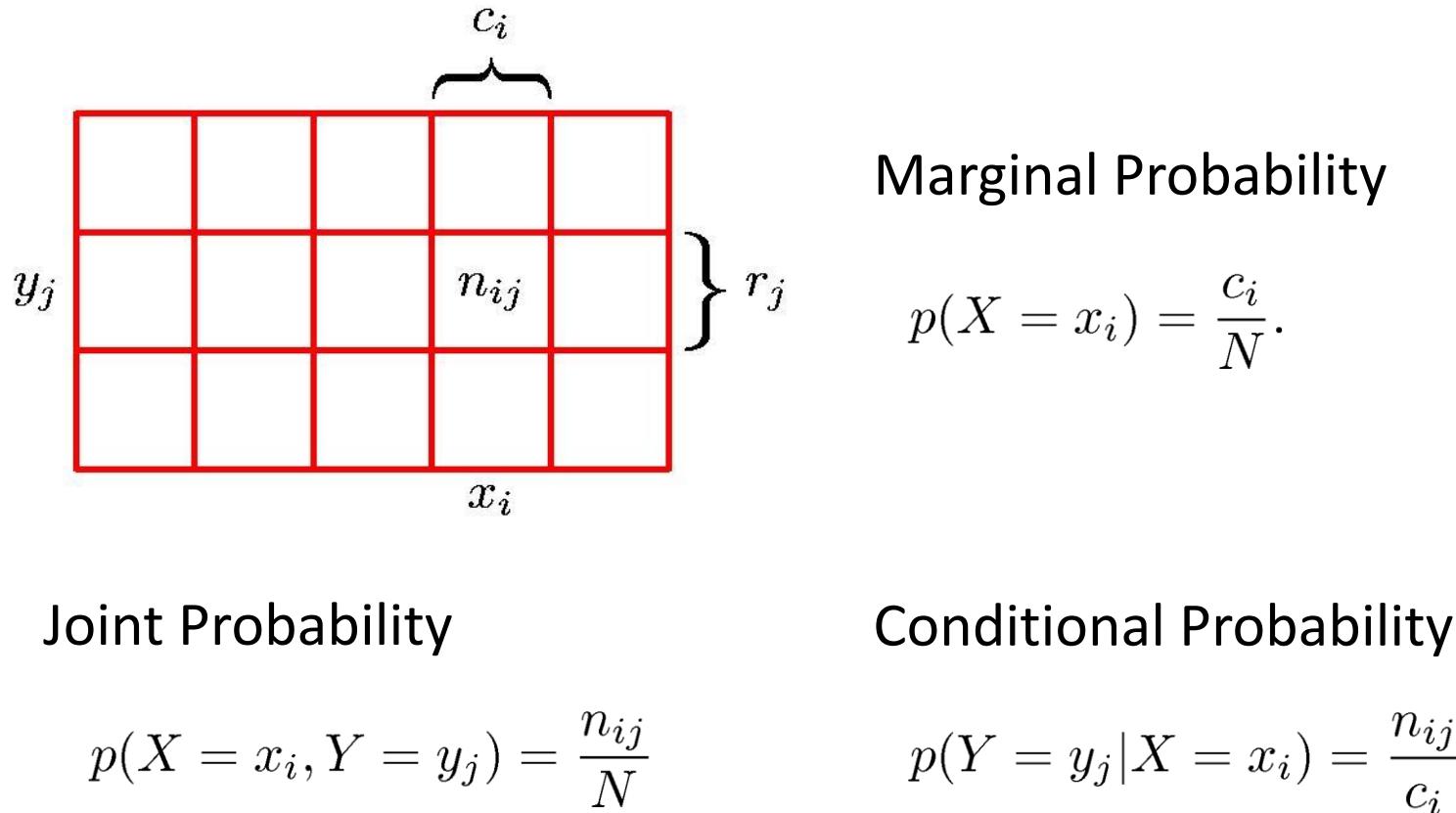
- Pattern Recognition
 - Curve Fitting and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theories
 - Entropy and Information
-

Probability Theory

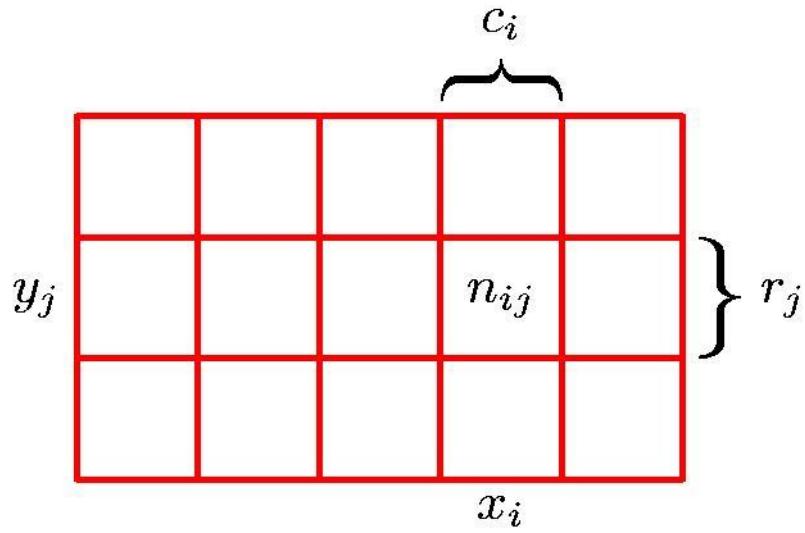
Apples and Oranges



Probability Theory



Probability Theory



Sum Rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N} \sum_{j=1}^L n_{ij}$$
$$= \sum_{j=1}^L p(X = x_i, Y = y_j)$$

Product Rule

$$\begin{aligned} p(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} \\ &= p(Y = y_j | X = x_i) p(X = x_i) \end{aligned}$$

The Rules of Probability

Sum Rule

$$p(X) = \sum_Y p(X, Y)$$

Product Rule

$$p(X, Y) = p(Y|X)p(X)$$

Bayes' Theorem

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

$$p(X) = \sum_Y p(X|Y)p(Y) \text{ : normalization}$$

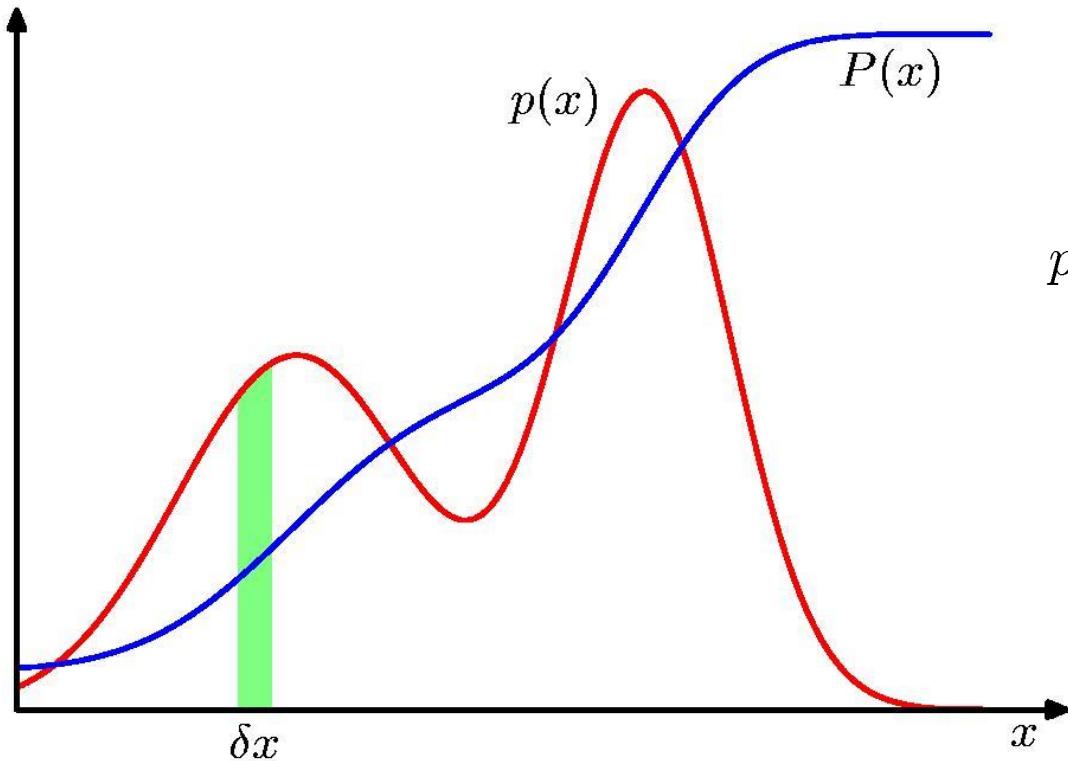
posterior \propto likelihood \times prior

$$p(Y/X)$$

$$p(X/Y)$$

$$p(Y)$$

Probability Densities



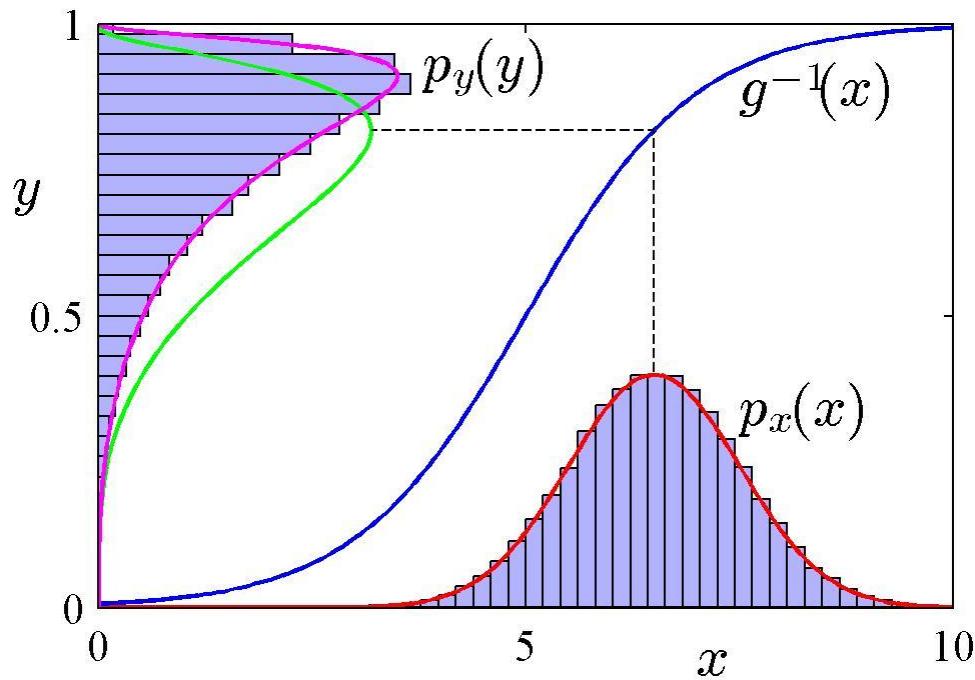
$$p(x \in (a, b)) = \int_a^b p(x) \, dx$$

$$P(z) = \int_{-\infty}^z p(x) \, dx$$

$$p(x) \geq 0$$

$$\int_{-\infty}^{\infty} p(x) \, dx = 1$$

Transformed Densities



$$\begin{aligned} p_y(y) &= p_x(x) \left| \frac{dx}{dy} \right| \\ &= p_x(g(y)) |g'(y)| \end{aligned}$$

$$x = g(y)$$

Expectations

$$\mathbb{E}[f] = \sum_x p(x)f(x)$$

$$\mathbb{E}[f] = \int p(x)f(x) \, dx$$

$$\mathbb{E}_x[f|y] = \sum_x p(x|y)f(x)$$


Conditional Expectation
(discrete)

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^N f(x_n)$$

Approximate Expectation
(discrete and continuous)

Variances and Covariances

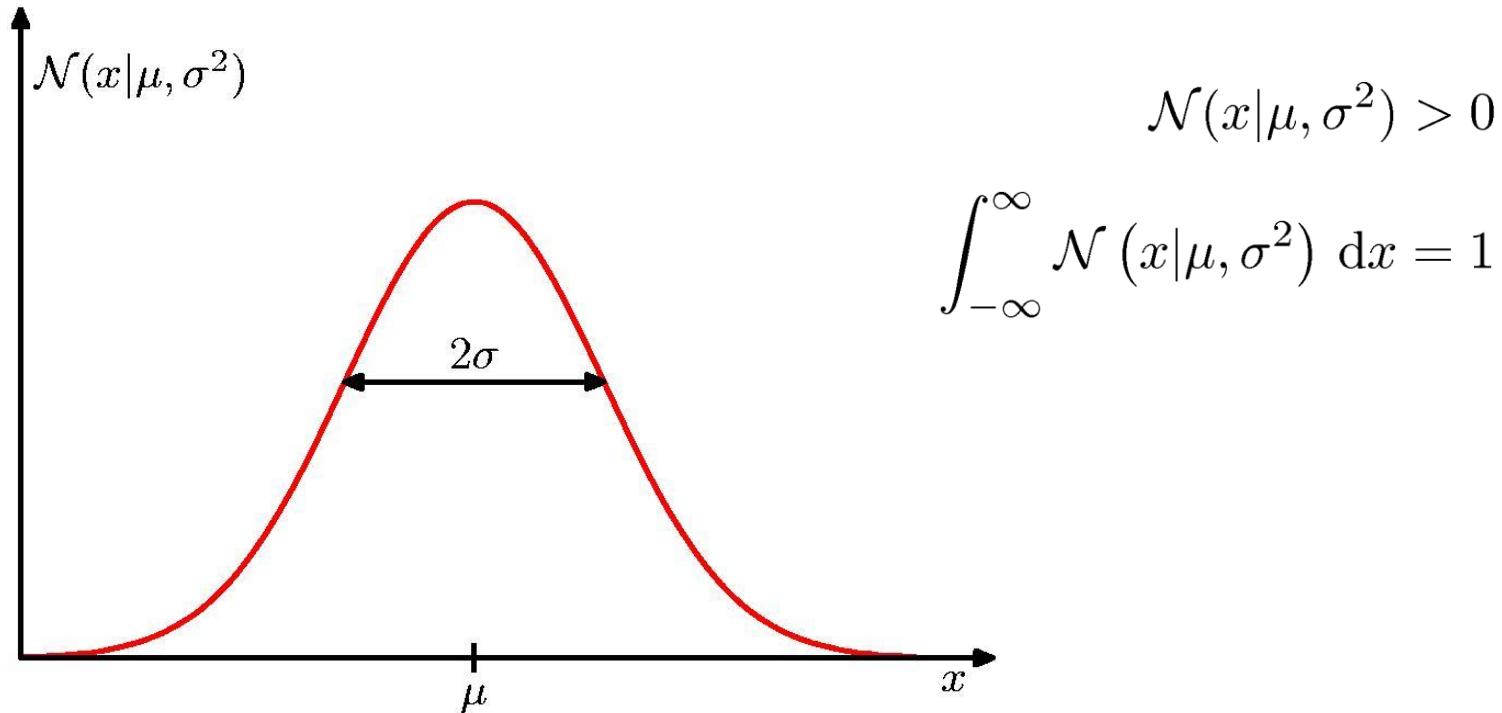
$$\text{var}[f] = \mathbb{E} \left[(f(x) - \mathbb{E}[f(x)])^2 \right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

$$\begin{aligned}\text{cov}[x, y] &= \mathbb{E}_{x,y} [\{x - \mathbb{E}[x]\} \{y - \mathbb{E}[y]\}] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]\end{aligned}$$

$$\begin{aligned}\text{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}} [\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T]\}] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^T]\end{aligned}$$

The Gaussian Distribution

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



Gaussian Mean and Variance

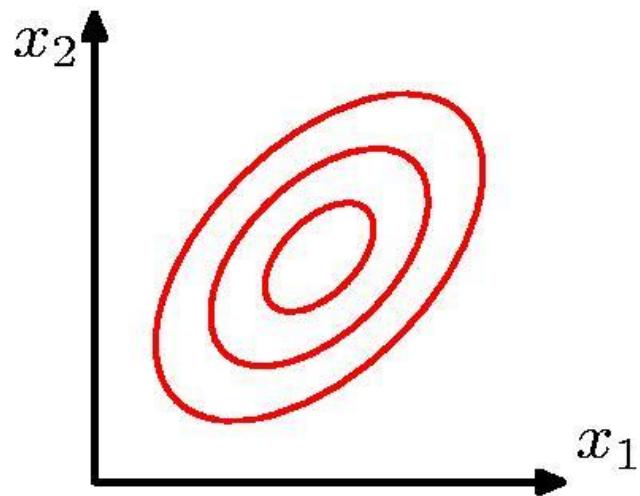
$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) x \, dx = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) x^2 \, dx = \mu^2 + \sigma^2$$

$$\text{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

The Multivariate Gaussian

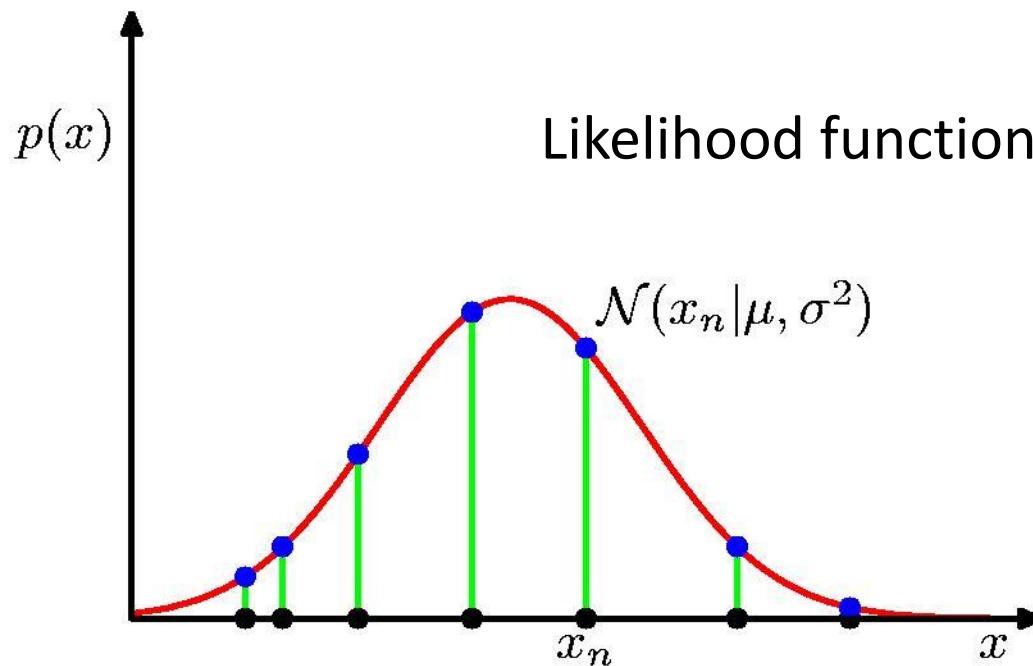
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$



Outlines

- Pattern Recognition
 - Curve Fitting and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theories
 - Entropy and Information
-

Gaussian Parameter Estimation



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

Maximum (Log) Likelihood

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

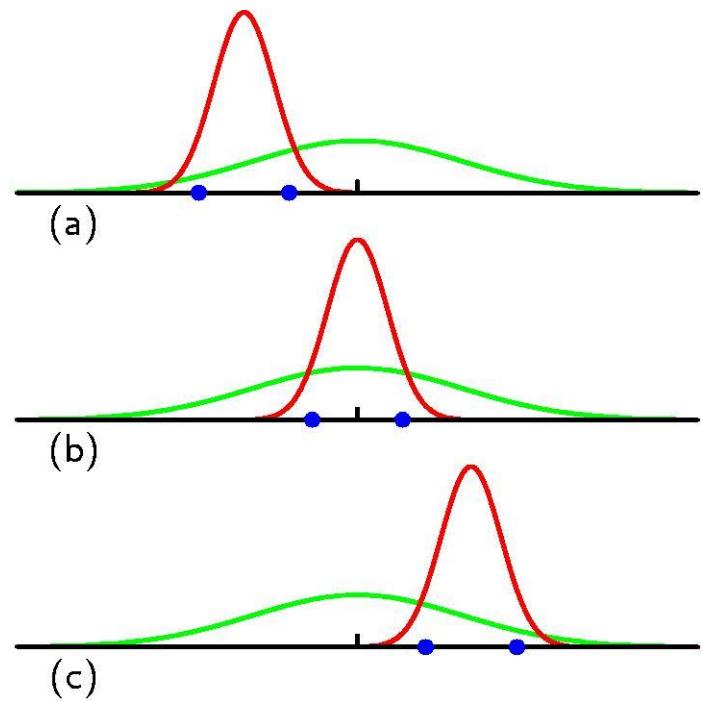
$$\mu_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N x_n \quad \sigma_{\text{ML}}^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu_{\text{ML}})^2$$

Properties of μ_{ML} and σ_{ML}^2

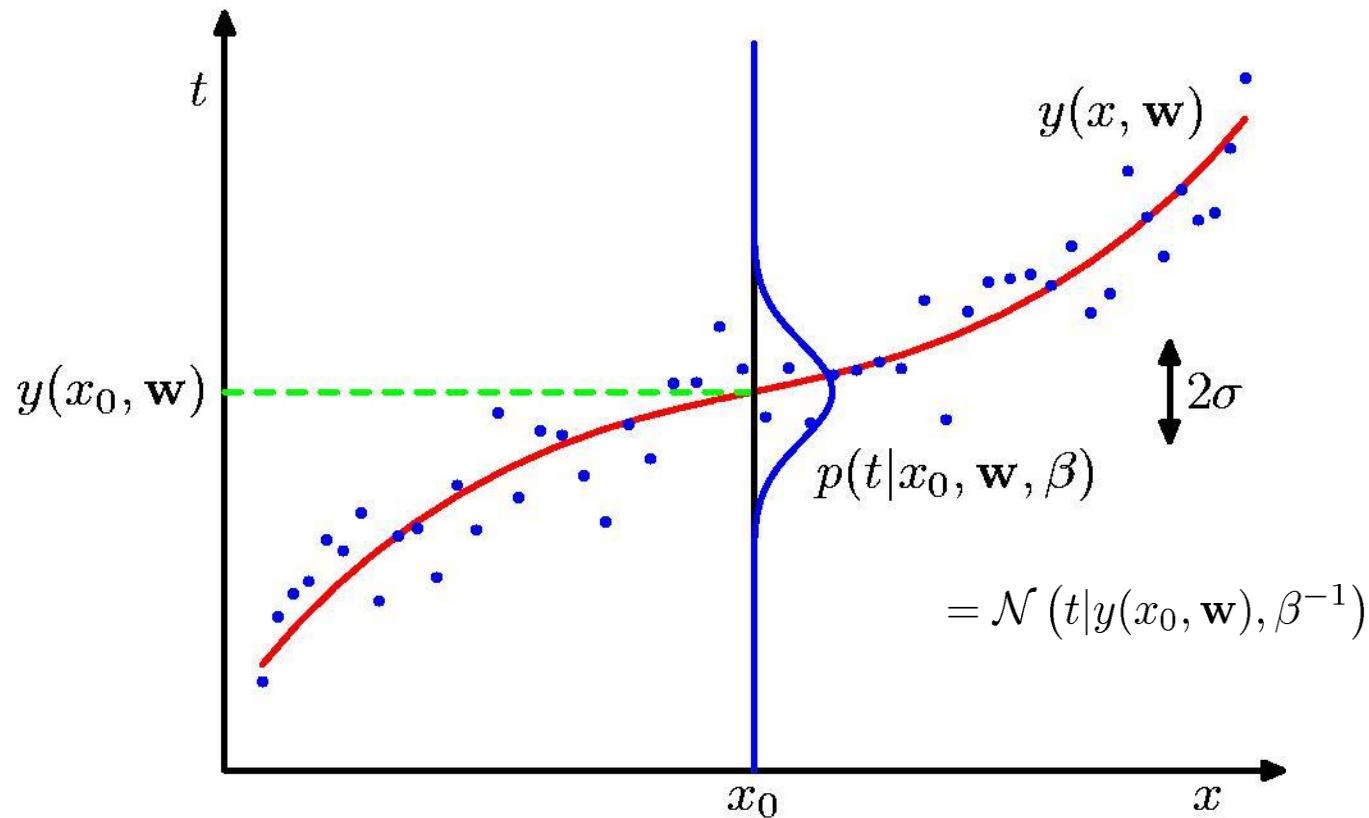
$$\mathbb{E}[\mu_{\text{ML}}] = \mu$$

$$\mathbb{E}[\sigma_{\text{ML}}^2] = \left(\frac{N-1}{N} \right) \sigma^2$$

$$\begin{aligned}\tilde{\sigma}^2 &= \frac{N}{N-1} \sigma_{\text{ML}}^2 \\ &= \frac{1}{N-1} \sum_{n=1}^N (x_n - \mu_{\text{ML}})^2\end{aligned}$$



Curve Fitting Re-visited



(t, x) : training data $\Rightarrow \mathbf{w}, \beta$ (\mathbf{w}, β, x_0) : $\Rightarrow p(t/x_0, \mathbf{w}, \beta)$

Maximum Likelihood

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

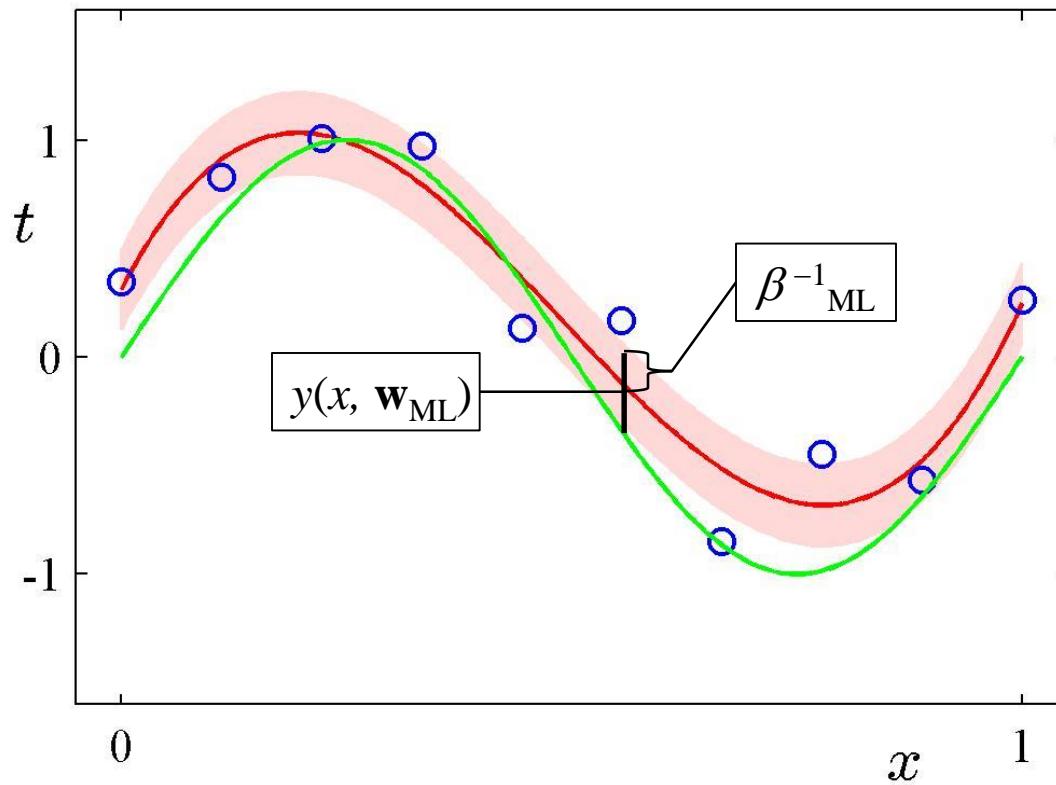
$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\underbrace{\frac{\beta}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2}_{\beta E(\mathbf{w})} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

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

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

Predictive Distribution

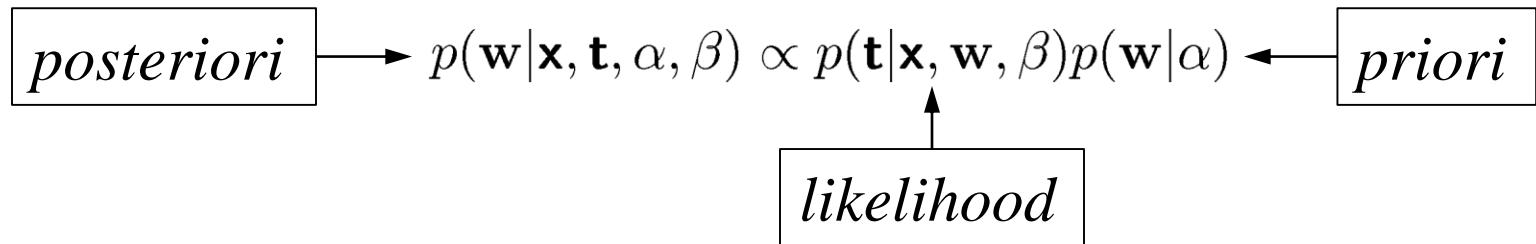
$$p(t|x, \mathbf{w}_{\text{ML}}, \beta_{\text{ML}}) = \mathcal{N}(t|y(x, \mathbf{w}_{\text{ML}}), \beta_{\text{ML}}^{-1})$$



MAP: A Step towards Bayes

MAP: Maximum *A Posteriori*

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



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

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

Bayesian Curve Fitting

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

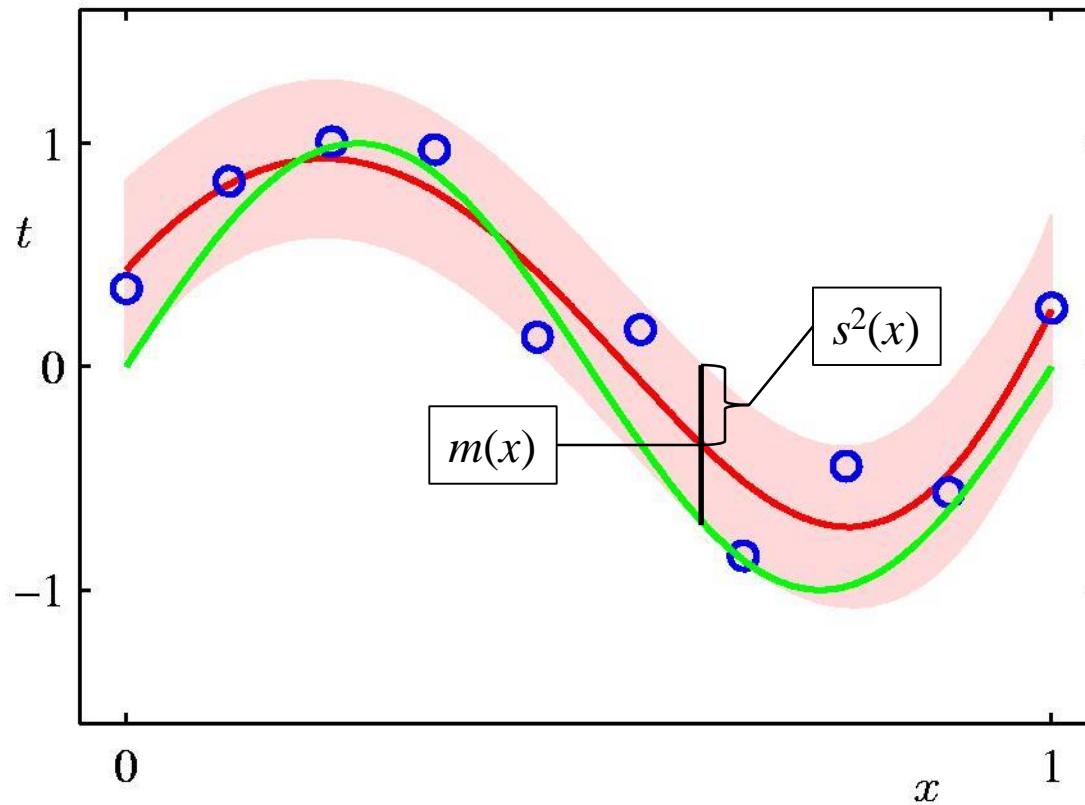
$$m(x) = \beta \boldsymbol{\phi}(x)^T \mathbf{S} \sum_{n=1}^N \boldsymbol{\phi}(x_n) t_n \quad s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \mathbf{S} \boldsymbol{\phi}(x)$$

$$\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^N \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^T \quad \boldsymbol{\phi}(x_n) = (x_n^0, \dots, x_n^M)^T$$

We will go through more details in a later lecture.

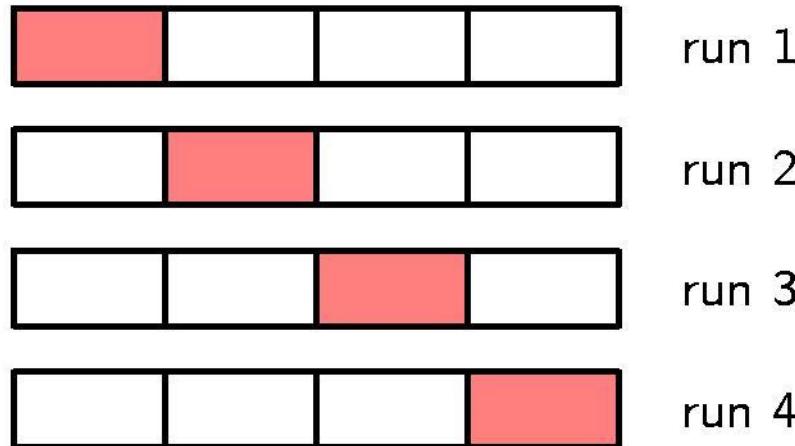
Bayesian Predictive Distribution

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



Model Selection and Evaluation

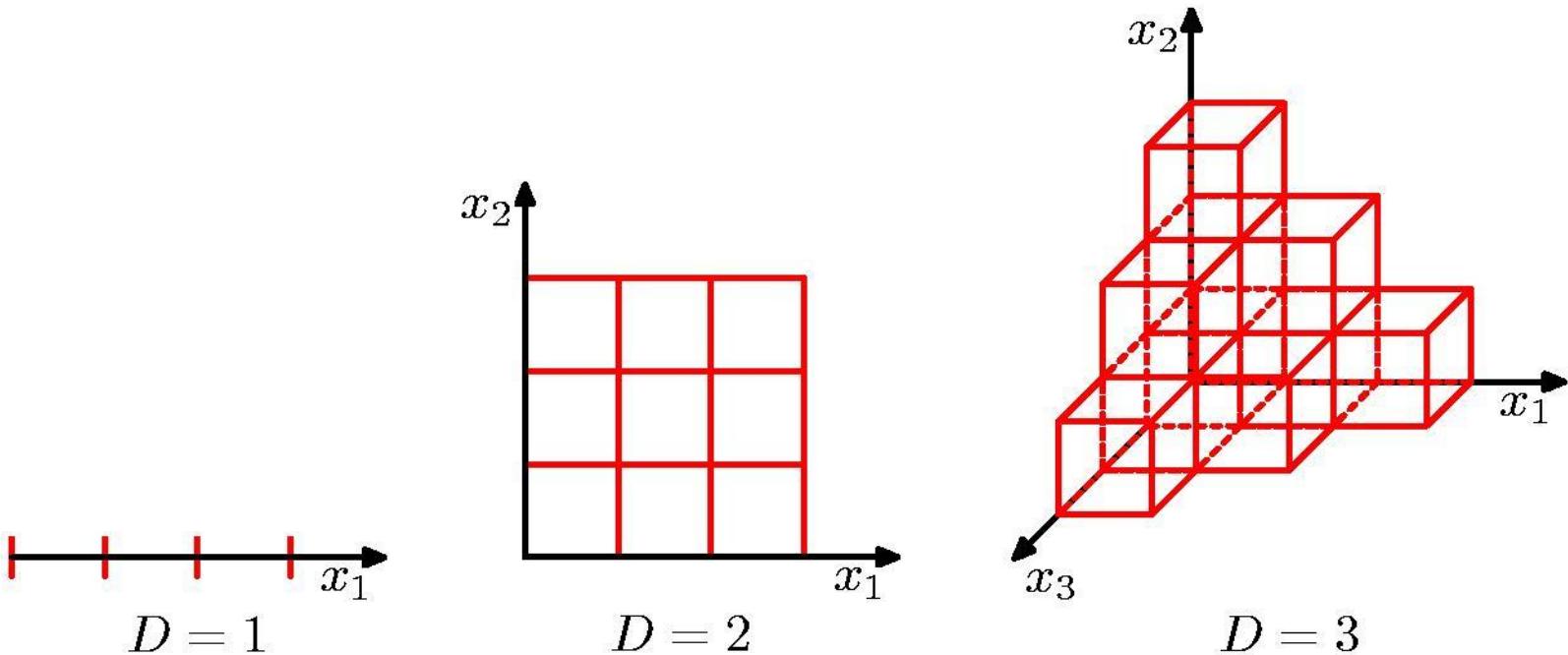
Cross-Validation



Outlines

- Pattern Recognition
 - Curve Fitting and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theory
 - Entropy and Information
-

Curse of Dimensionality

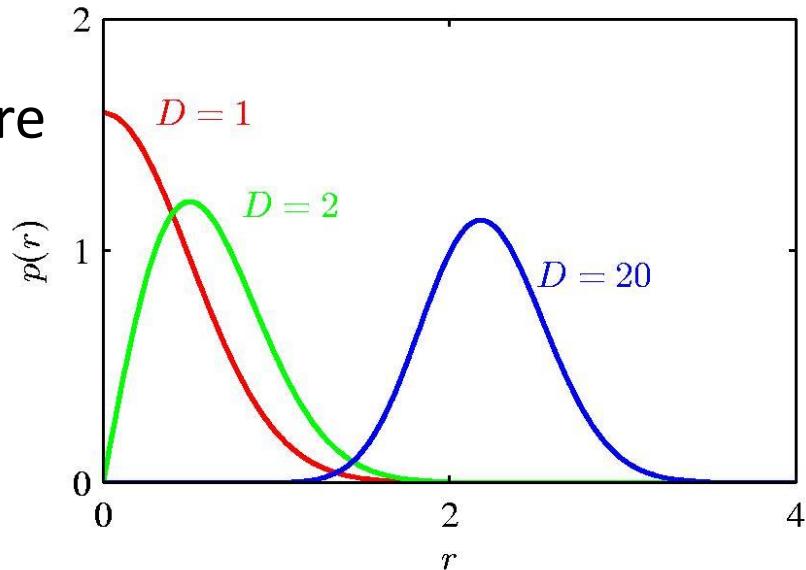
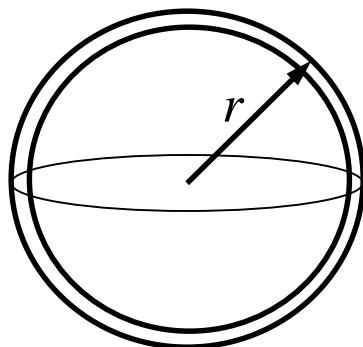


Curse of Dimensionality

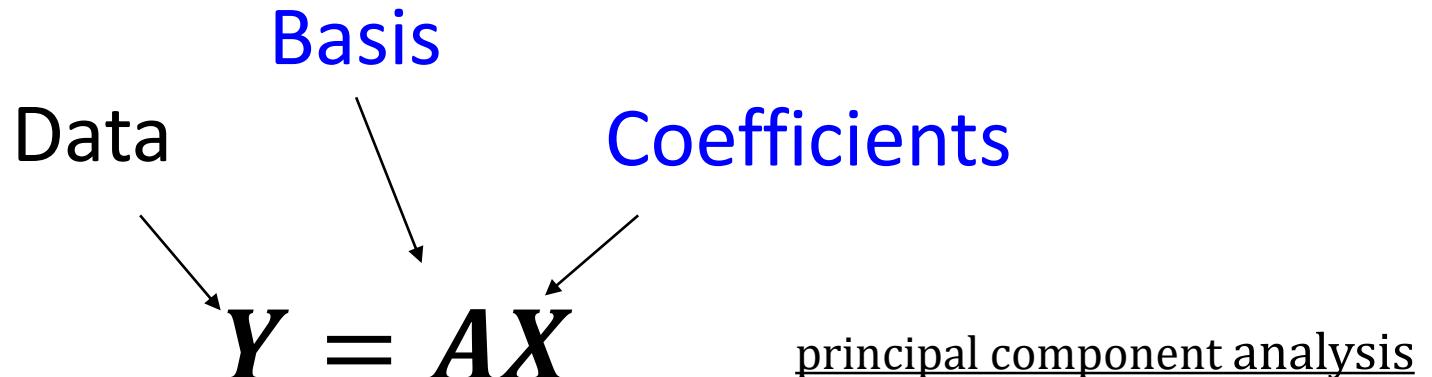
Polynomial curve fitting, $M = 3$

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

Gaussian Densities in
higher dimensions of a sphere



Reduction of Dimensionality (PCA)



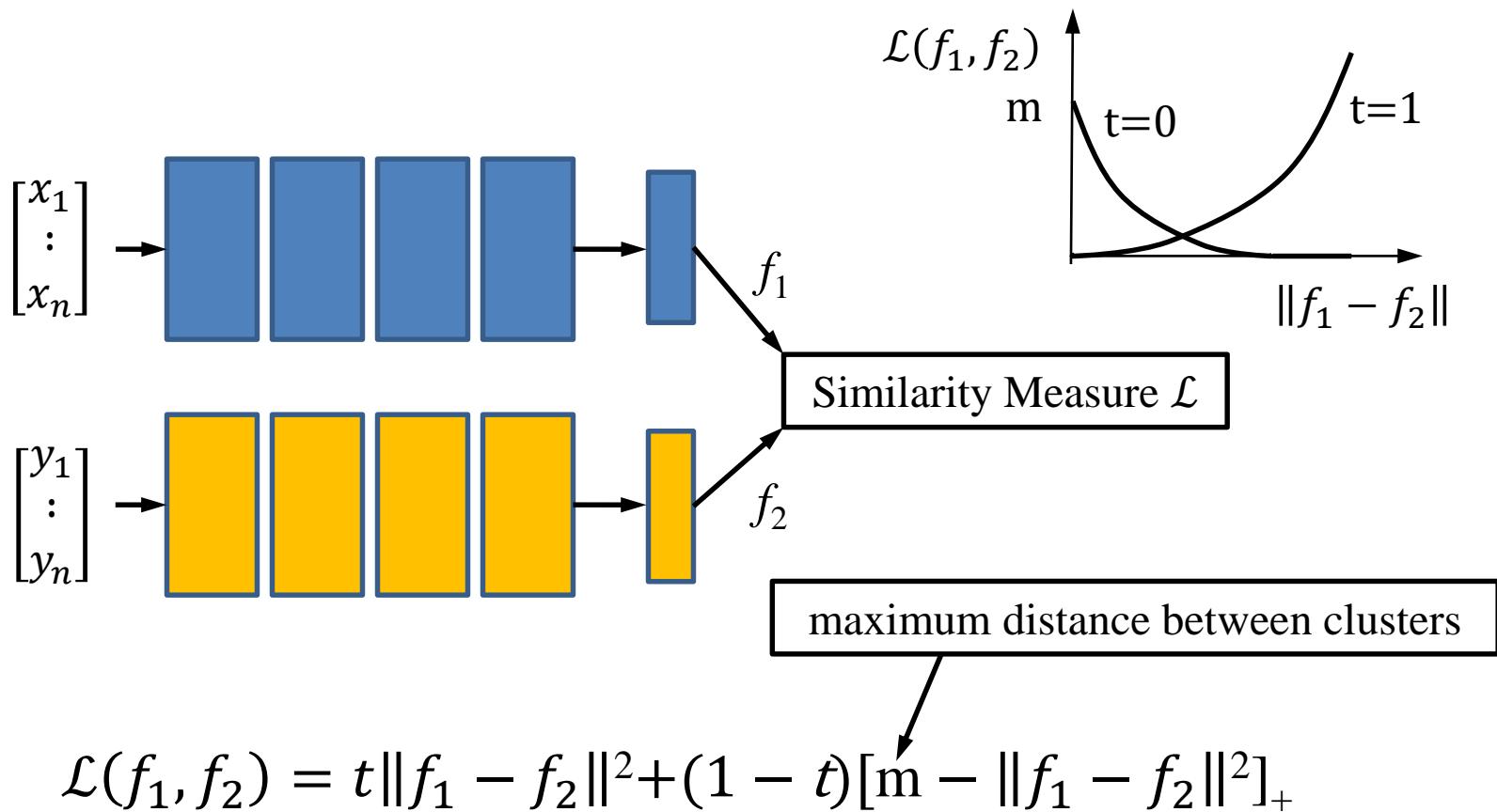
$$\max_{A_i} A_i^T COV(Y_i) A_i$$

A : rotation

$$A_i^{*T} COV(Y_i) A_i^* = \lambda_i \quad A_i^*: \text{optimal solution}$$

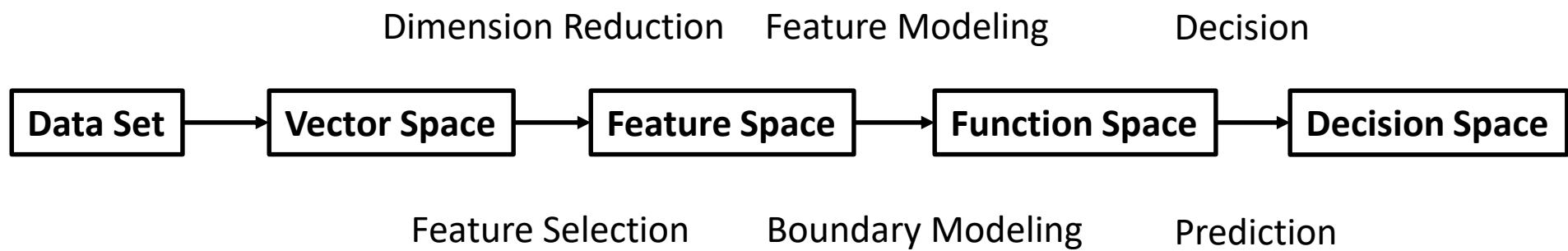
$$s.t. \quad A_i^T A_i = 1 \quad E[Y_i] = \mathbf{0}$$

Feature Extraction (Contrastive Loss)



$t=1$: two vectors belong to the same category; $[]_+$: non-negative

Machine Learning Pipeline



Outlines

- Pattern Recognition
 - Curve Fitting and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theory
 - Entropy and Information
-

Decision Theory

Inference step

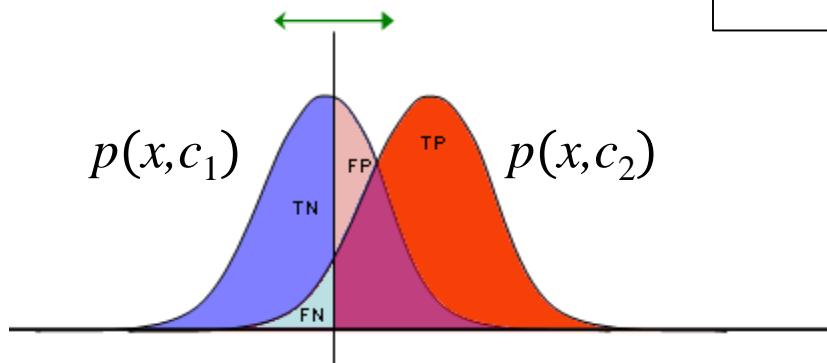
Determine either $p(t|x)$ or $p(x,t)$.

Decision step

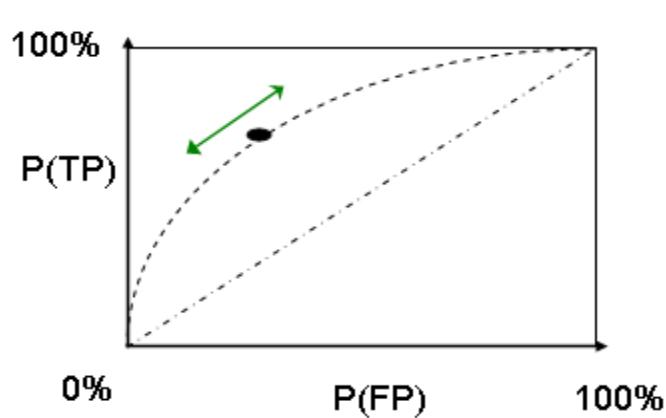
For given x , determine optimal t .

Receiver Operating Characteristic Curve

$$\begin{aligned} p(x) &= p(x, c_1) + p(x, c_2) \\ &= p(x/c_1) p(c_1) + p(x/c_2) p(c_2) \end{aligned}$$

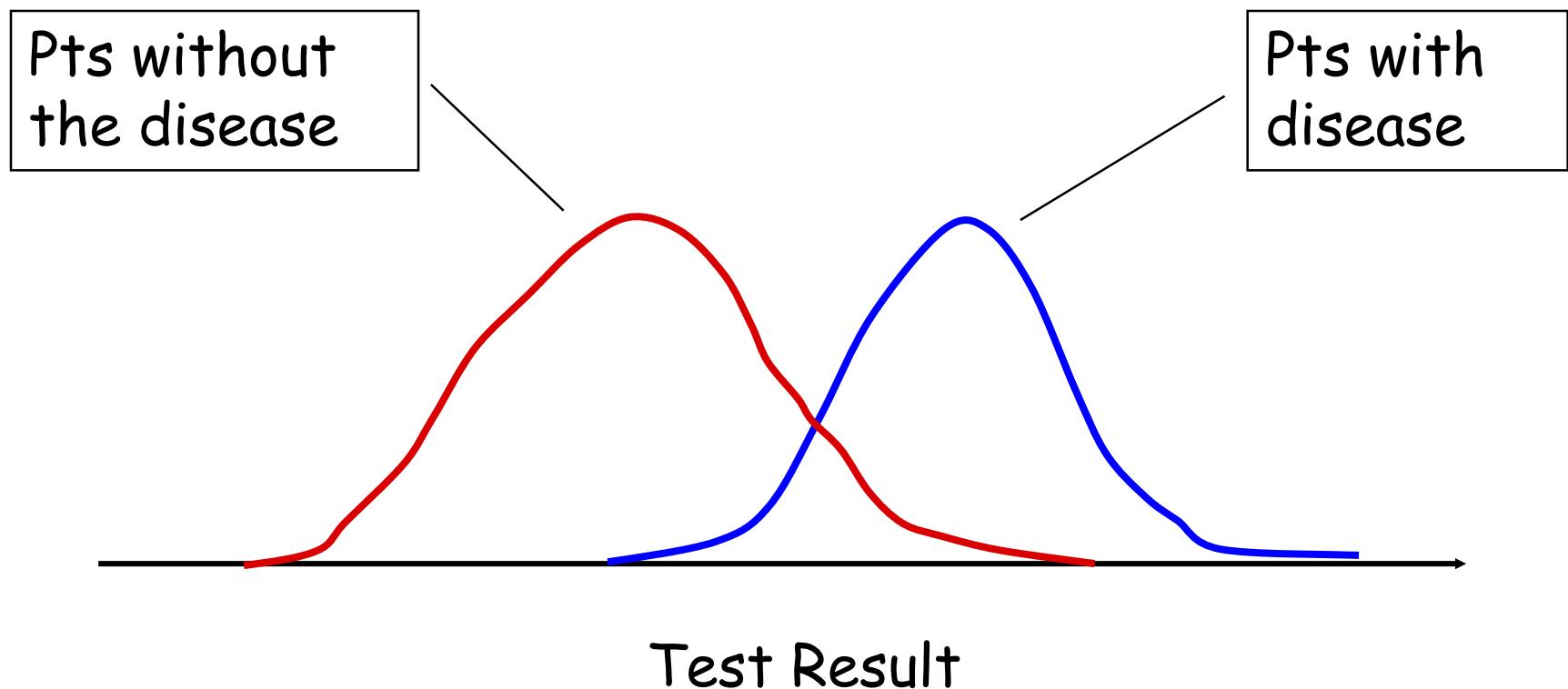


TP	FP
FN	TN
1	1

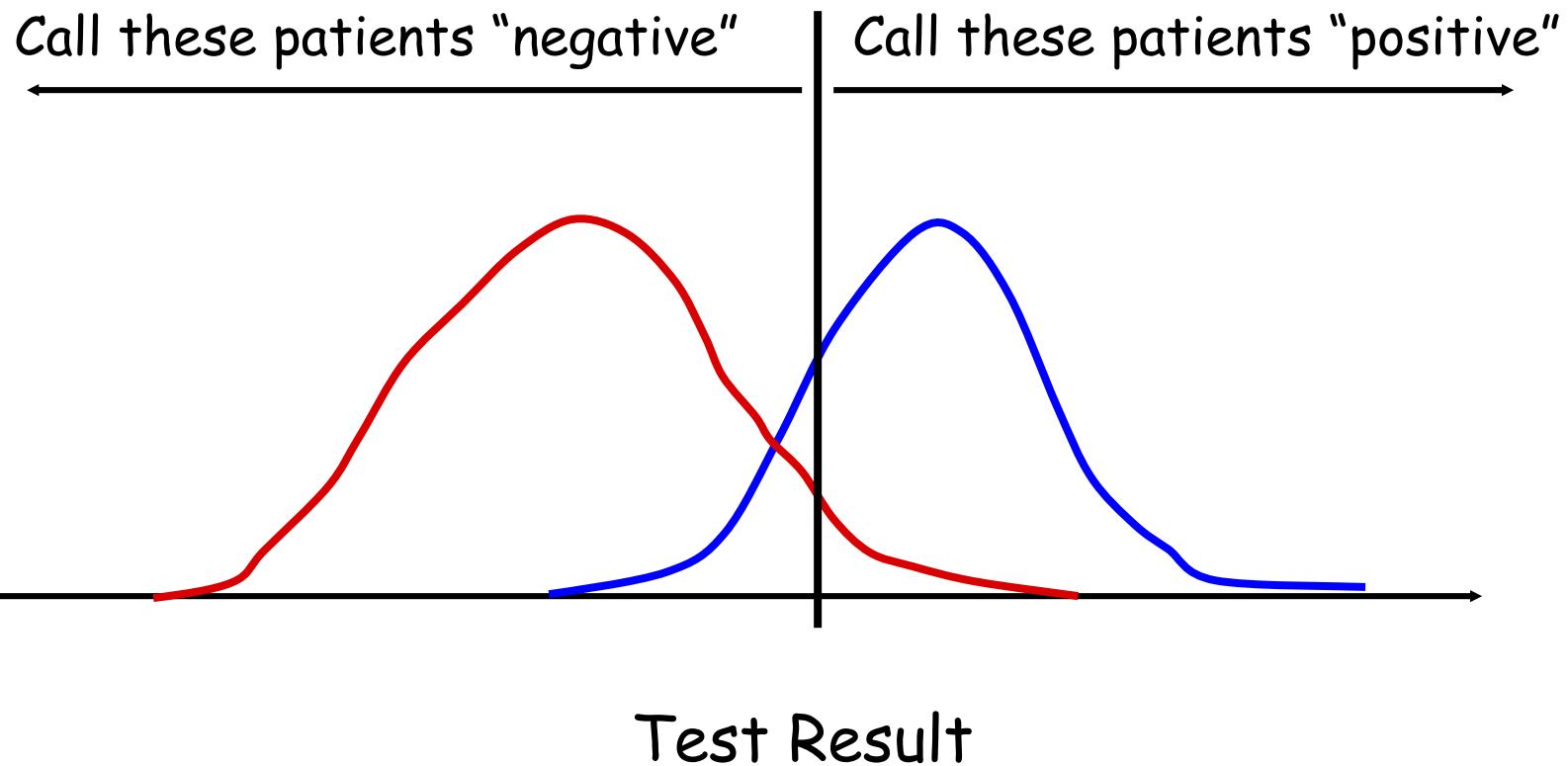


$p(c_1), p(c_2)$ 是各类别的权重系数

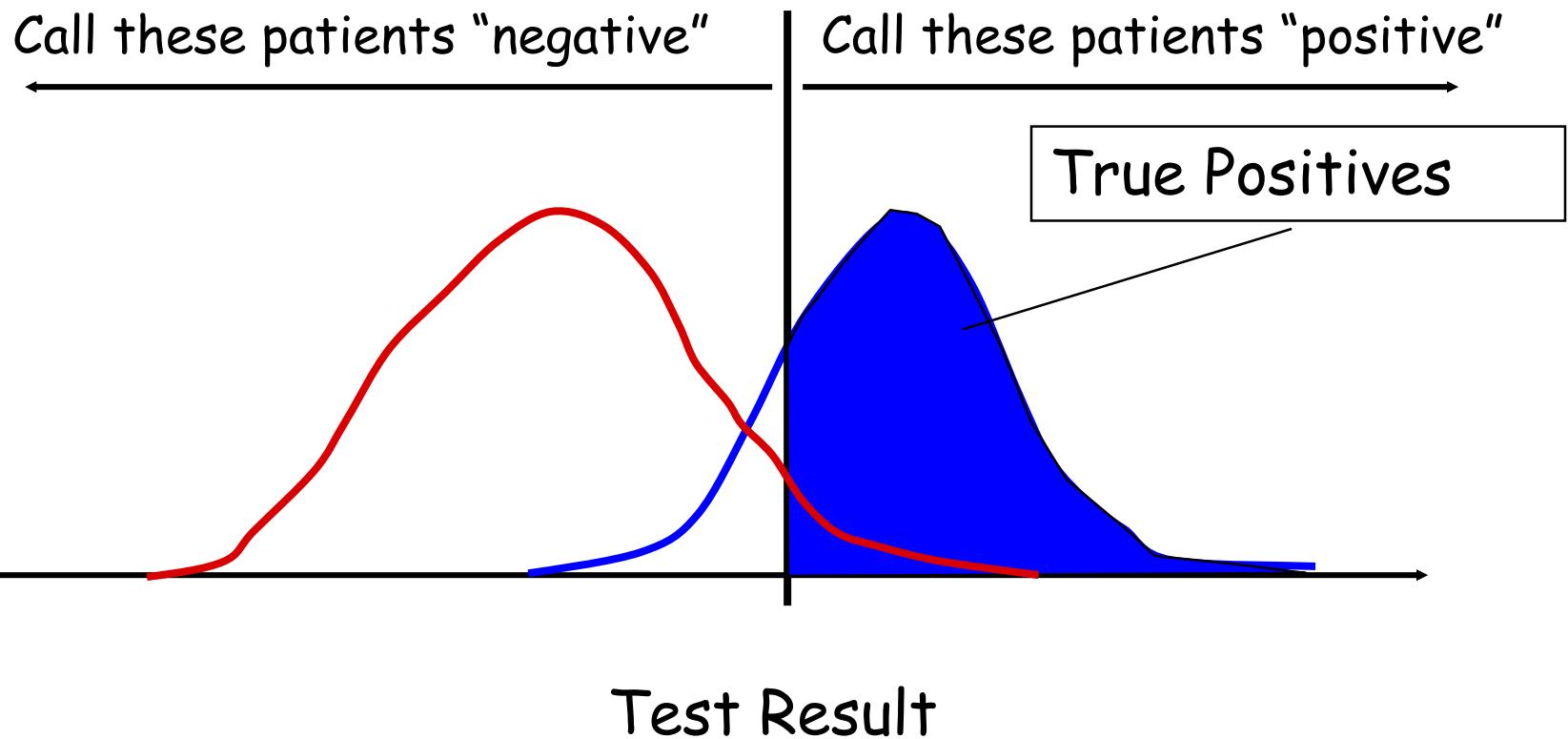
Bimodal Distribution (Data Model)



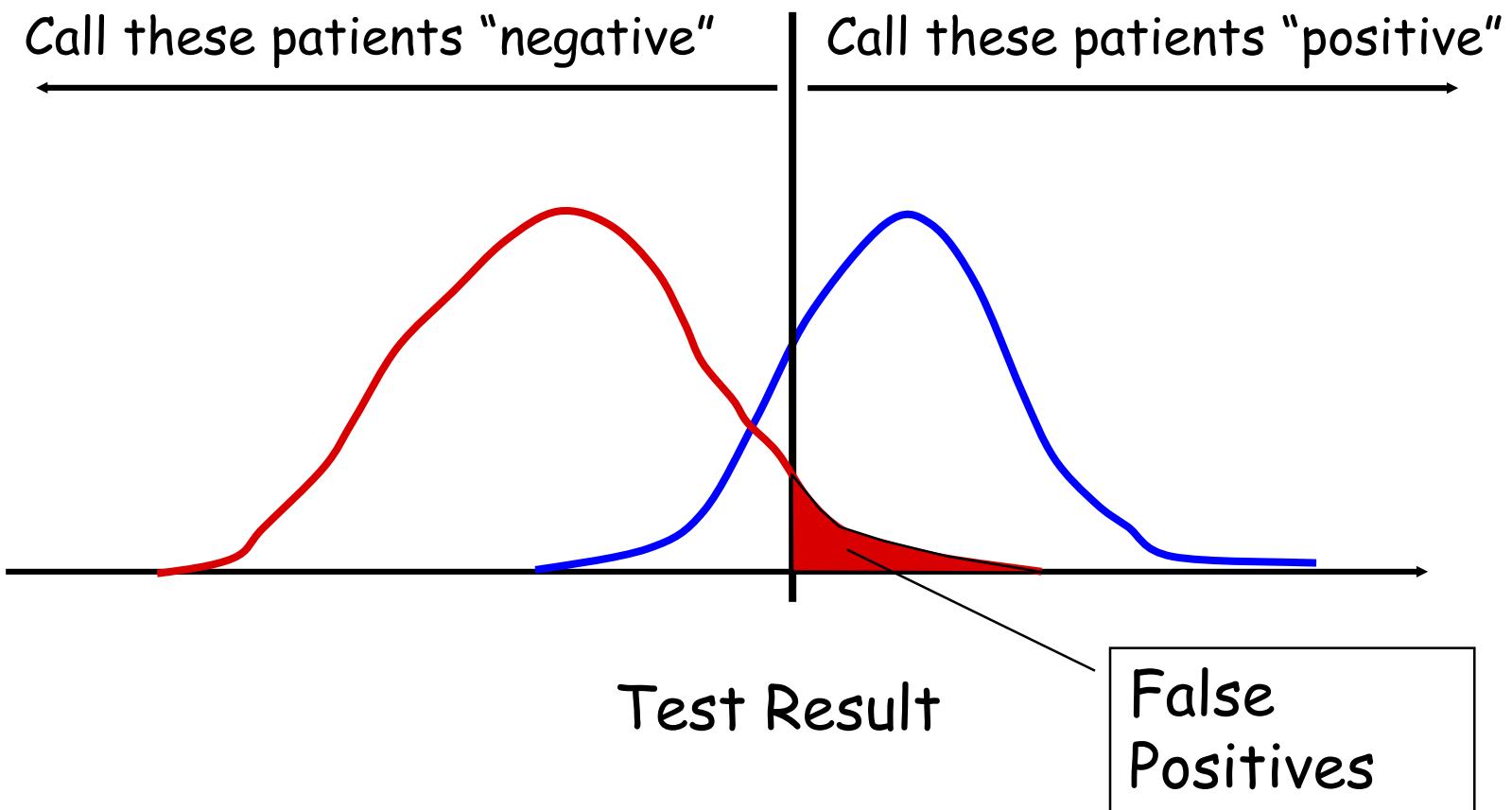
Decision Threshold (Boundary Model)



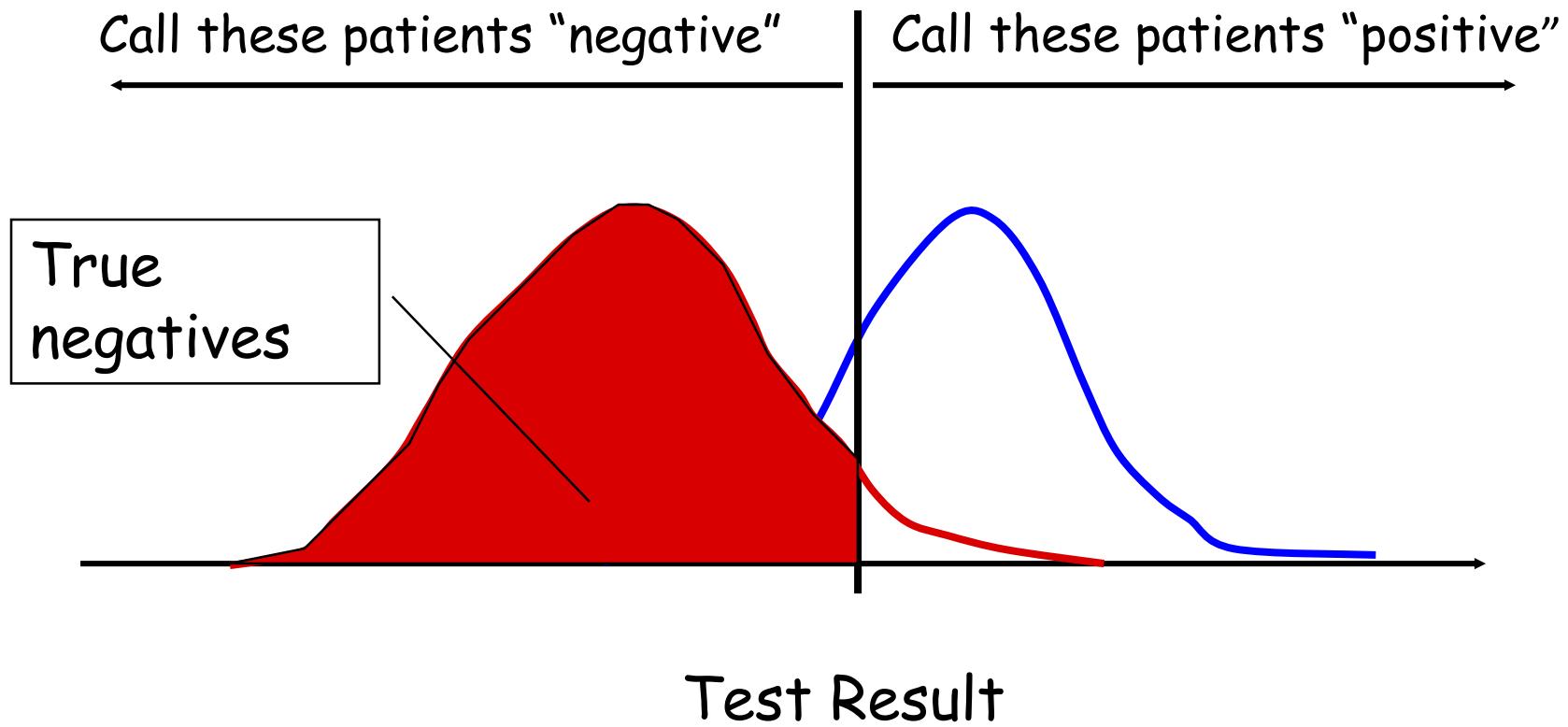
True Positive



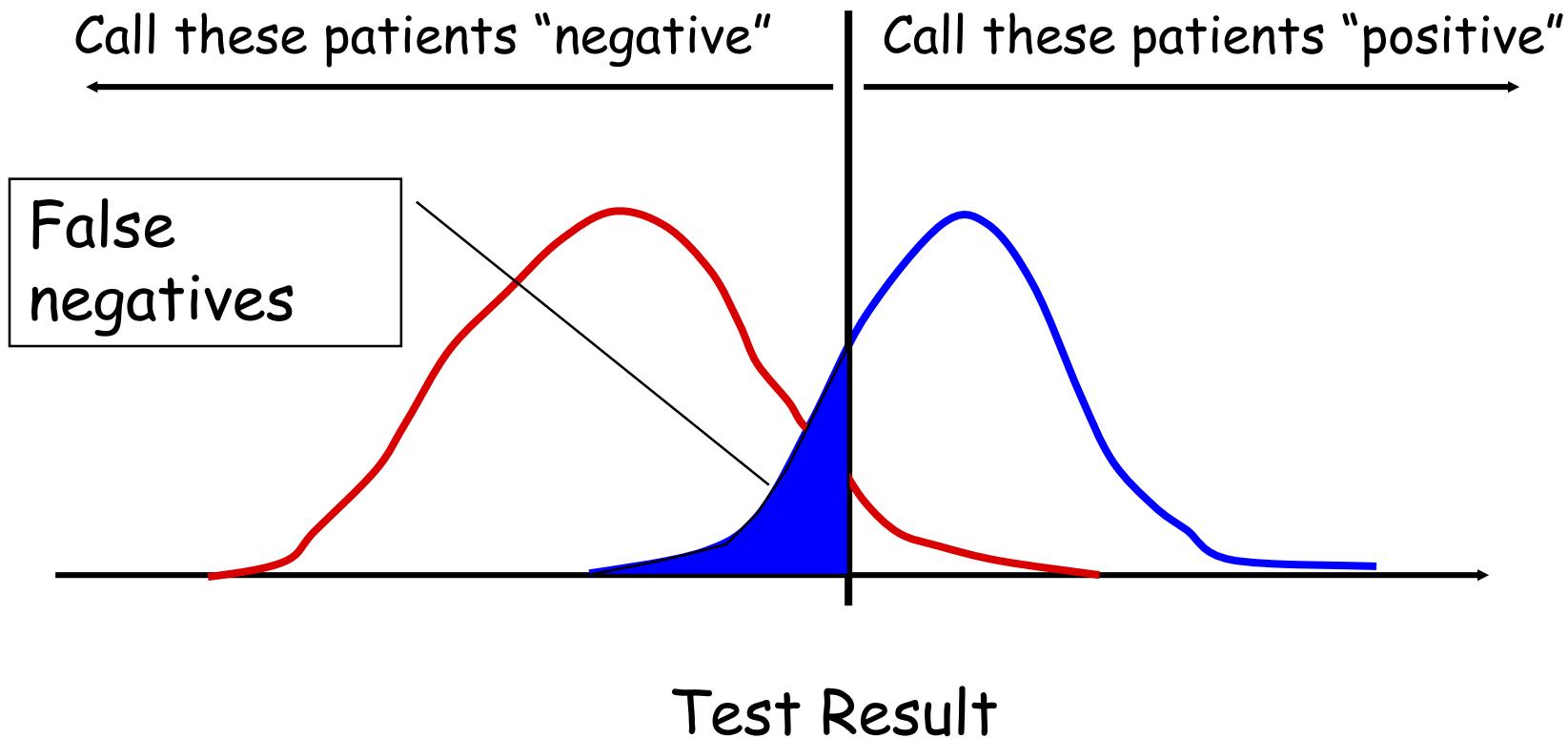
False Positive



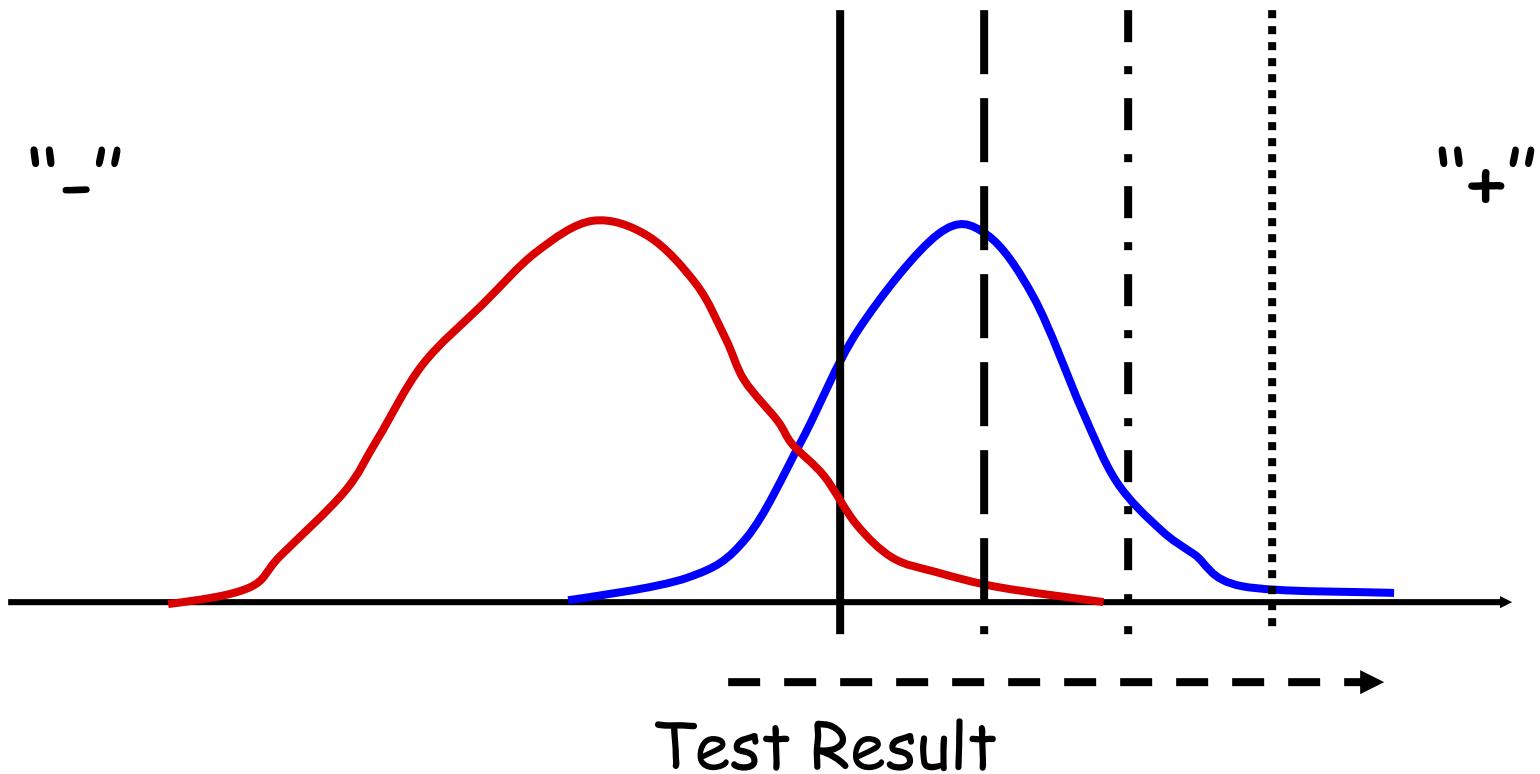
True Negative



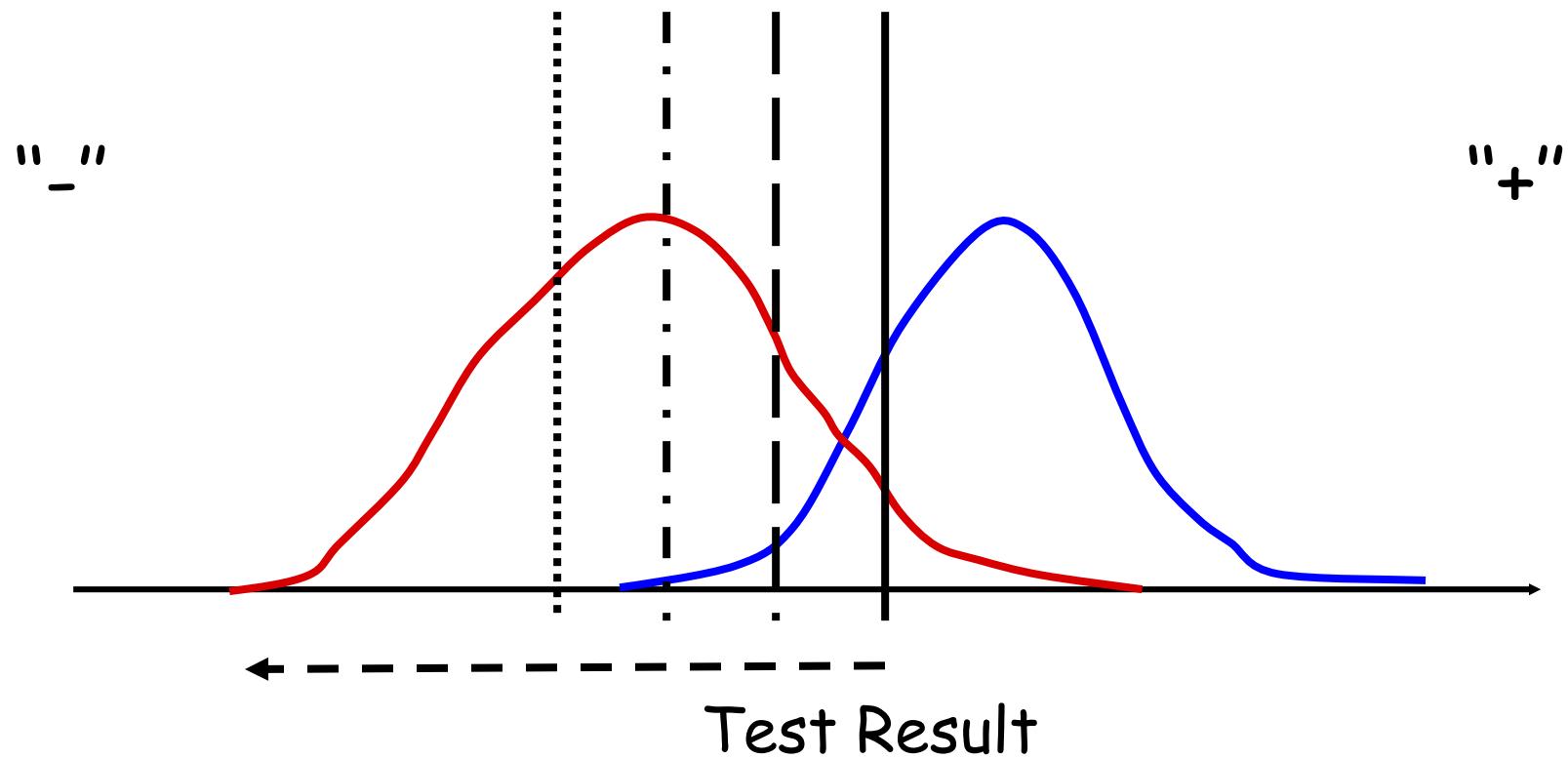
False Negative



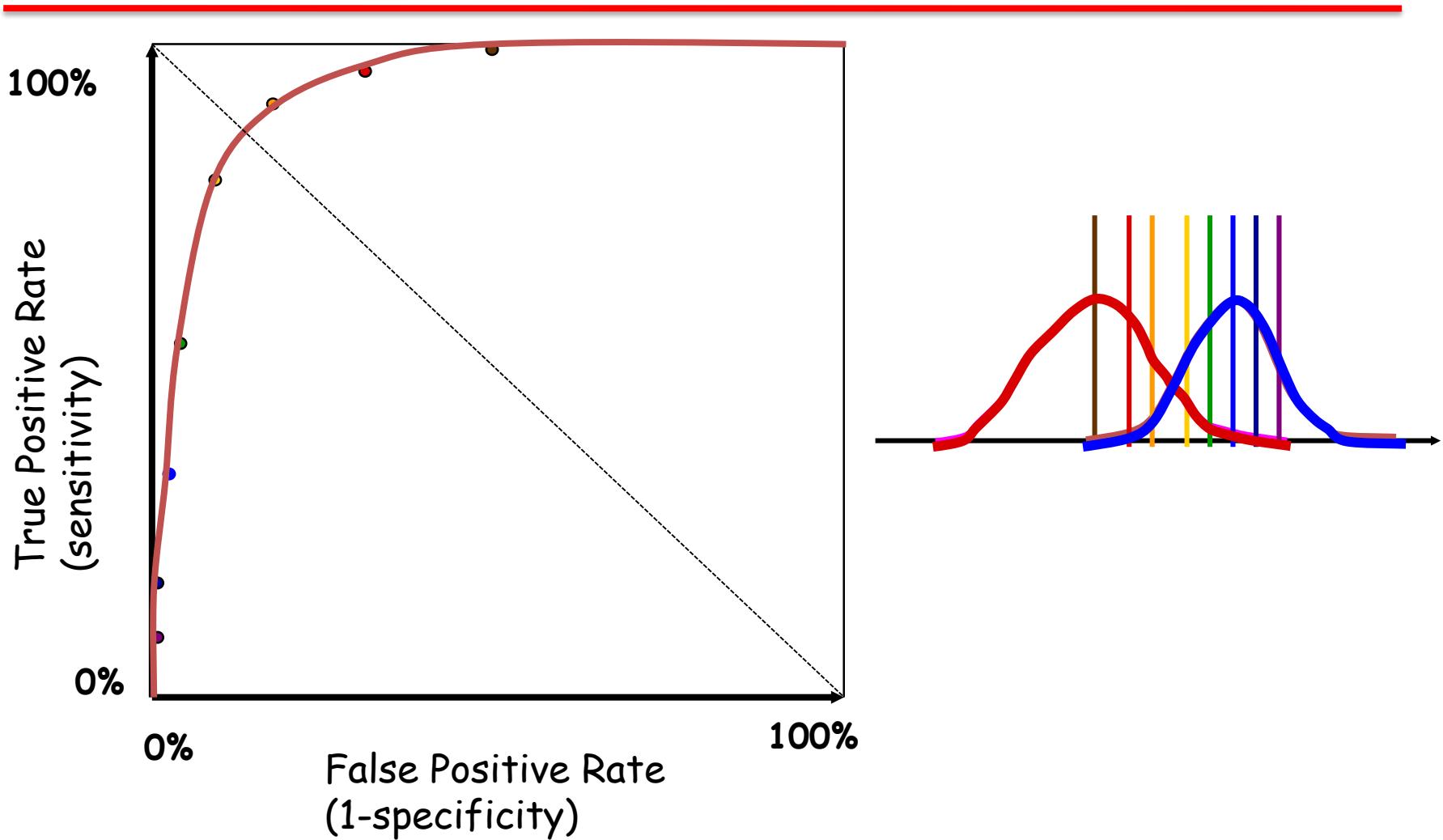
Moving the Threshold: Right



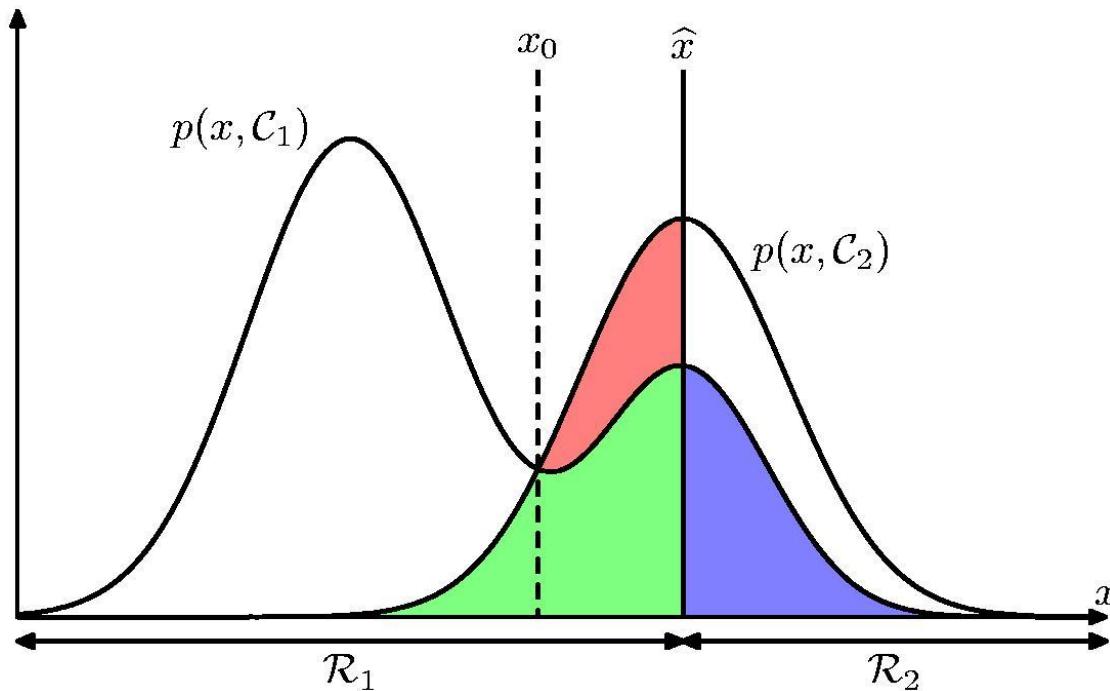
Moving the Threshold: Left



ROC Curve



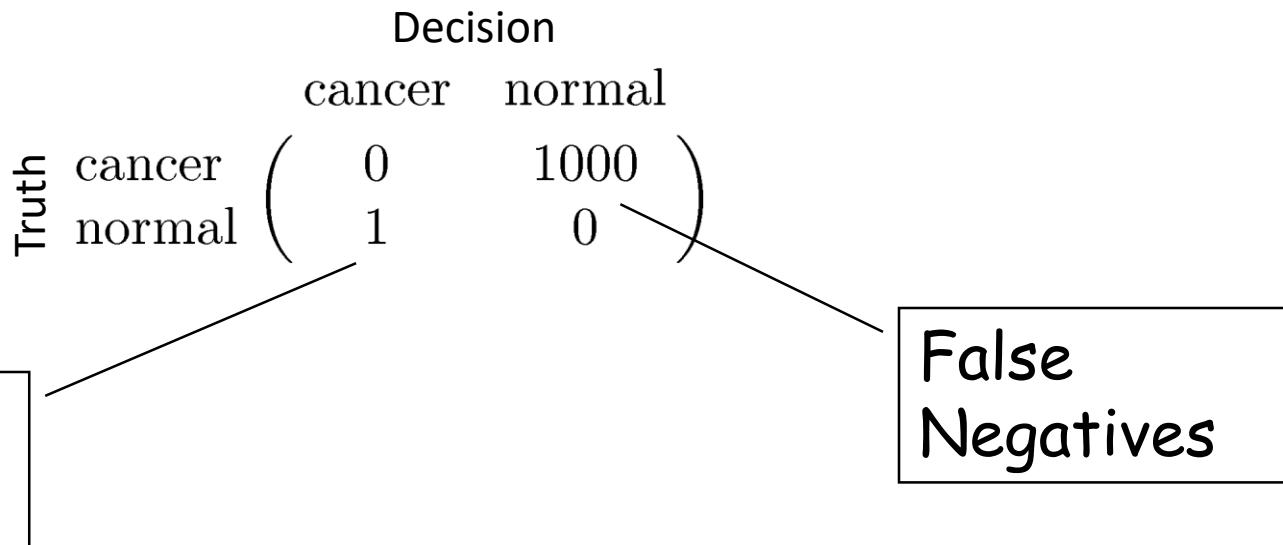
Minimum Misclassification Rate



$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2) d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1) d\mathbf{x}. \end{aligned}$$

Minimum Expected Loss

Example: classify medical images as ‘cancer’ or ‘normal’



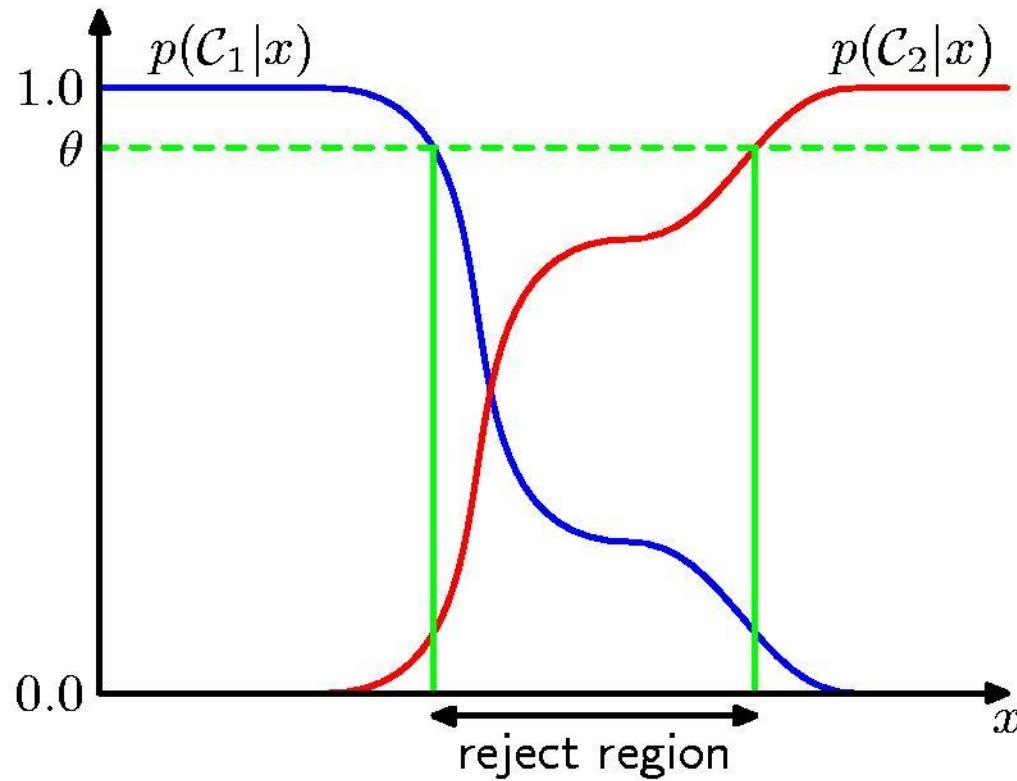
Minimum Expected Loss

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) d\mathbf{x}$$

Regions \mathcal{R}_j are chosen to minimize

$$\mathbb{E}[L] = \sum_k L_{kj} p(\mathcal{C}_k | \mathbf{x})$$

Reject Option



Why Separate Inference and Decision?

- Minimizing risk (loss matrix may change over time)
- Reject option
- Unbalanced class priors
- Combining models

Decision Theory for Regression

Inference step

Determine $p(\mathbf{x}, t)$.

Decision step

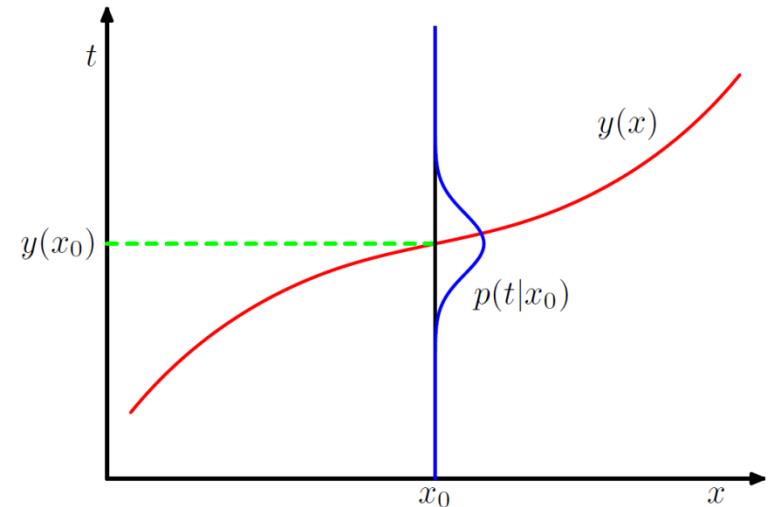
For given \mathbf{x} , make optimal prediction, $y(\mathbf{x})$, for t .

Loss function: $\mathbb{E}[L] = \iint L(t, y(\mathbf{x}))p(\mathbf{x}, t) \, d\mathbf{x} \, dt$

The Expected Squared Loss Function

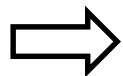
$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt$$

$\mathbb{E}(t|\mathbf{x})$ 实际上是关于 \mathbf{x} 的函数 : $u(\mathbf{x})$



$$\begin{aligned}\{y(\mathbf{x}) - t\}^2 &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t\}^2 \\ &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 + 2\{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}\{\mathbb{E}[t|\mathbf{x}] - t\} + \{\mathbb{E}[t|\mathbf{x}] - t\}^2\end{aligned}$$

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 p(\mathbf{x}) d\mathbf{x} + \int \text{var}[t|\mathbf{x}] p(\mathbf{x}) d\mathbf{x}$$



$$y(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$$

predictor

noise

$y(x)$: an estimator of the mean of t for given \mathbf{x}

Generative vs Discriminative

Generative approach:

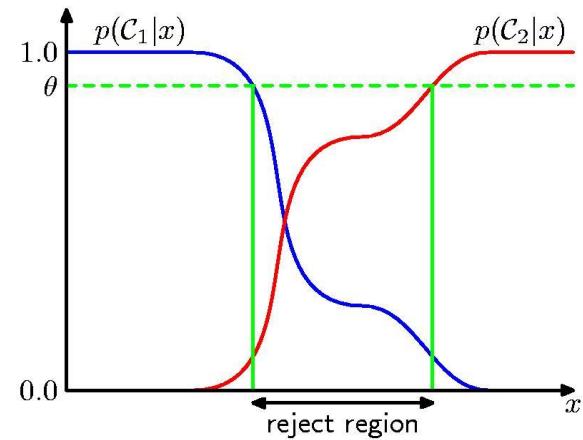
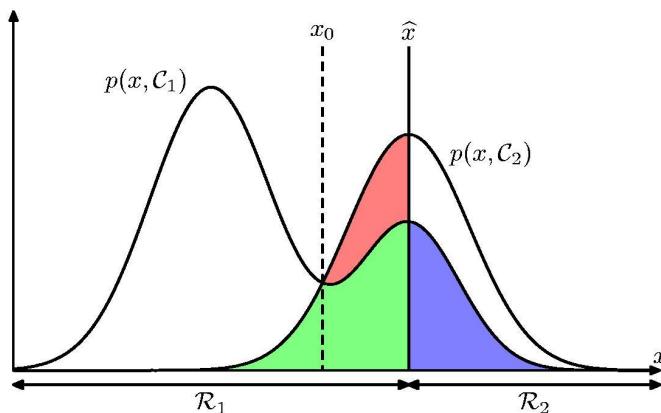
Model $p(t, \mathbf{x}) = p(\mathbf{x}|t)p(t)$

Use Bayes' theorem $p(t|\mathbf{x}) = \frac{p(\mathbf{x}|t)p(t)}{p(\mathbf{x})}$

Discriminative approach:

Model $p(t|\mathbf{x})$ directly

t : category



Outlines

- Pattern Recognition
 - Curve Fitting and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theories
 - Entropy and Information
-

Entropy

$$H[x] = - \sum_x p(x) \log_2 p(x)$$

Important quantity in

- coding theory
- statistical physics
- machine learning

Entropy

Coding theory: x discrete with 8 possible states; how many bits to transmit the state of x ?

All states equally likely

$$H[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits.}$$

Entropy

x	a	b	c	d	e	f	g	h
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$
code	0	10	110	1110	111100	111101	111110	111111

$$\begin{aligned} H[x] &= -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{16} \log_2 \frac{1}{16} - \frac{4}{64} \log_2 \frac{1}{64} \\ &= 2 \text{ bits} \end{aligned}$$

$$\begin{aligned} \text{average code length} &= \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + 4 \times \frac{1}{64} \times 6 \\ &= 2 \text{ bits} \end{aligned}$$

Entropy

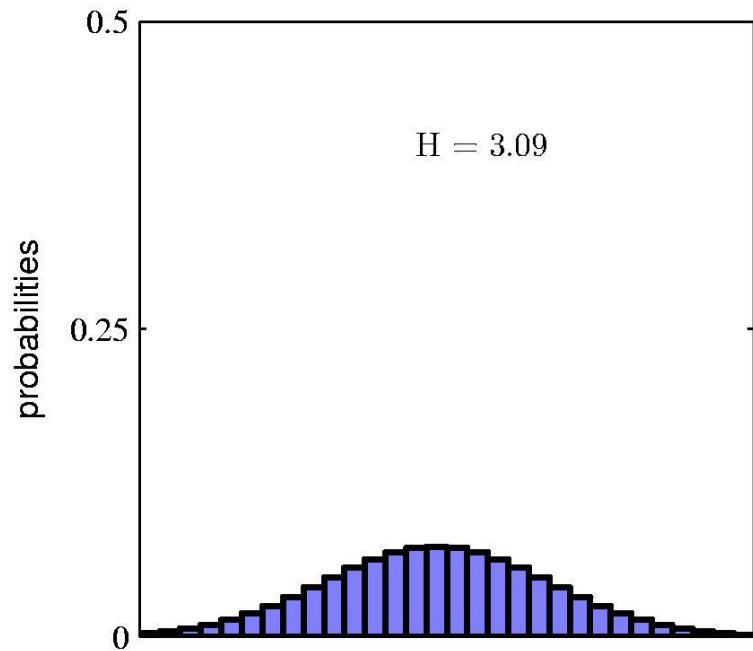
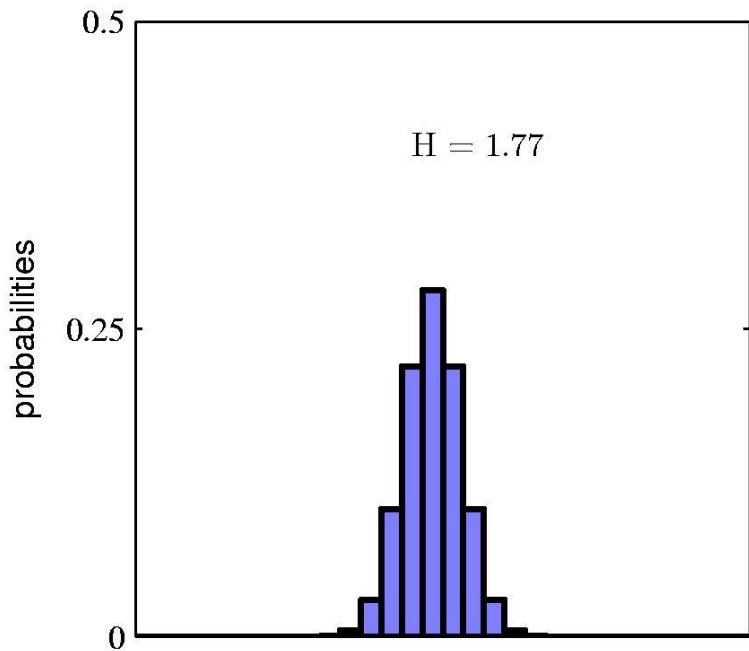
In how many ways can N identical objects be allocated M bins?

$$W = \frac{N!}{\prod_i n_i!}$$

$$H = \frac{1}{N} \ln W \simeq - \lim_{N \rightarrow \infty} \sum_i \left(\frac{n_i}{N} \right) \ln \left(\frac{n_i}{N} \right) = - \sum_i p_i \ln p_i$$

Entropy maximized when $\forall i : p_i = \frac{1}{M}$

Entropy



Differential Entropy

Put bins of width Δ along the real line

$$\lim_{\Delta \rightarrow 0} \left\{ - \sum_i p(x_i) \Delta \ln p(x_i) \right\} = - \int p(x) \ln p(x) dx$$

Differential entropy maximized (for fixed σ^2) when

$$p(x) = \mathcal{N}(x|\mu, \sigma^2)$$

in which case

$$H[x] = \frac{1}{2} \{1 + \ln(2\pi\sigma^2)\}.$$

Conditional Entropy

$$H[y|x] = - \iint p(y, x) \ln p(y|x) dy dx$$

$$H[x, y] = H[y|x] + H[x]$$

The Kullback-Leibler Divergence

$$\begin{aligned} \text{Cross Entropy } C(p||q) & \quad \text{Entropy } H(p) \\ \text{KL}(p||q) &= - \int p(\mathbf{x}) \ln q(\mathbf{x}) d\mathbf{x} - \left(- \int p(\mathbf{x}) \ln p(\mathbf{x}) d\mathbf{x} \right) \end{aligned}$$

$$= - \int p(\mathbf{x}) \ln \left\{ \frac{q(\mathbf{x})}{p(\mathbf{x})} \right\} d\mathbf{x}$$

$$\text{Cross Entropy} \quad \text{Negative Entropy}$$
$$\text{KL}(p||q) \simeq \frac{1}{N} \sum_{n=1}^N \{-\ln q(\mathbf{x}_n|\boldsymbol{\theta}) + \ln p(\mathbf{x}_n)\}$$

$$\text{KL}(p||q) \geq 0$$

$$\text{KL}(p||q) \not\equiv \text{KL}(q||p)$$

KL divergence describes a distance between model p and model q

Cross Entropy for Machine Learning

Goal of Machine Learning: $p(\text{real data}) \approx p(\text{model} / \theta)$

we assume: $p(\text{training data}) \approx p(\text{real data})$

Operation of Machine Learning: $p(\text{training data}) \approx p(\text{model} / \theta)$

$$\min_{\theta} \text{KL}(p(\text{training data}) \parallel p(\text{model} | \theta))$$



$$\min_{\theta} C(p(\text{training data}) \parallel p(\text{model} | \theta))$$

as $H(p(\text{training data}))$ is fixed

Cross Entropy for Machine Learning

$$C(p(\text{training data}) \parallel p(\text{model} | \theta))$$

Bernoulli model: $p(\text{model} / \theta) = \rho^t(1 - \rho)^{1-t}$ t_n : training data

Cross entropy : $C = -\frac{1}{N} \sum_n t_n \ln \rho + (1 - t_n) \ln(1 - \rho)$ ρ : model parameter

Gaussian model: $p(\text{model} / \theta) \propto e^{-0.5(t-\mu)^2}$ t_n : training data

Cross entropy : $C \propto \frac{1}{N} \sum_n (t_n - \mu)^2$ μ : model parameter

Mutual Information

$$\begin{aligned} I[\mathbf{x}, \mathbf{y}] &\equiv \text{KL}(p(\mathbf{x}, \mathbf{y}) \| p(\mathbf{x})p(\mathbf{y})) \\ &= - \iint p(\mathbf{x}, \mathbf{y}) \ln \left(\frac{p(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x}, \mathbf{y})} \right) d\mathbf{x} d\mathbf{y} \end{aligned}$$

$$I[\mathbf{x}, \mathbf{y}] = H[\mathbf{x}] - H[\mathbf{x}|\mathbf{y}] = H[\mathbf{y}] - H[\mathbf{y}|\mathbf{x}]$$

Mutual information describes the degree of dependence between \mathbf{x} and \mathbf{y}

Information Gain



$H[x]$: uncertain of balls

$H[x|y]$:
uncertain of balls after
weighing once

x : one ball lighter

y : weighing once

$x|y$: one ball lighter
after weighing once

$$I[x, y] = H[x] - H[x|y] = \log_2 3$$

$$H[x] = \log_2 N$$

After weighing $\frac{N}{3}$ times, all the uncertainties can be removed

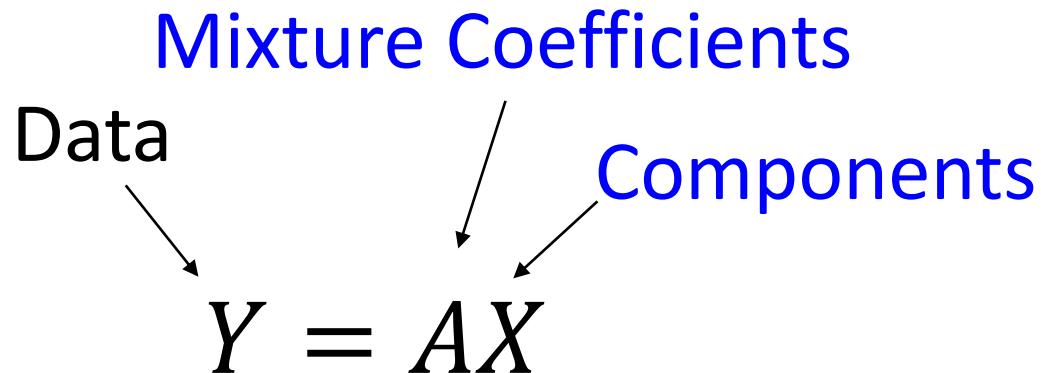
Independent Signal Separation



a alamy stock photo

EXT3MY
www.alamy.com

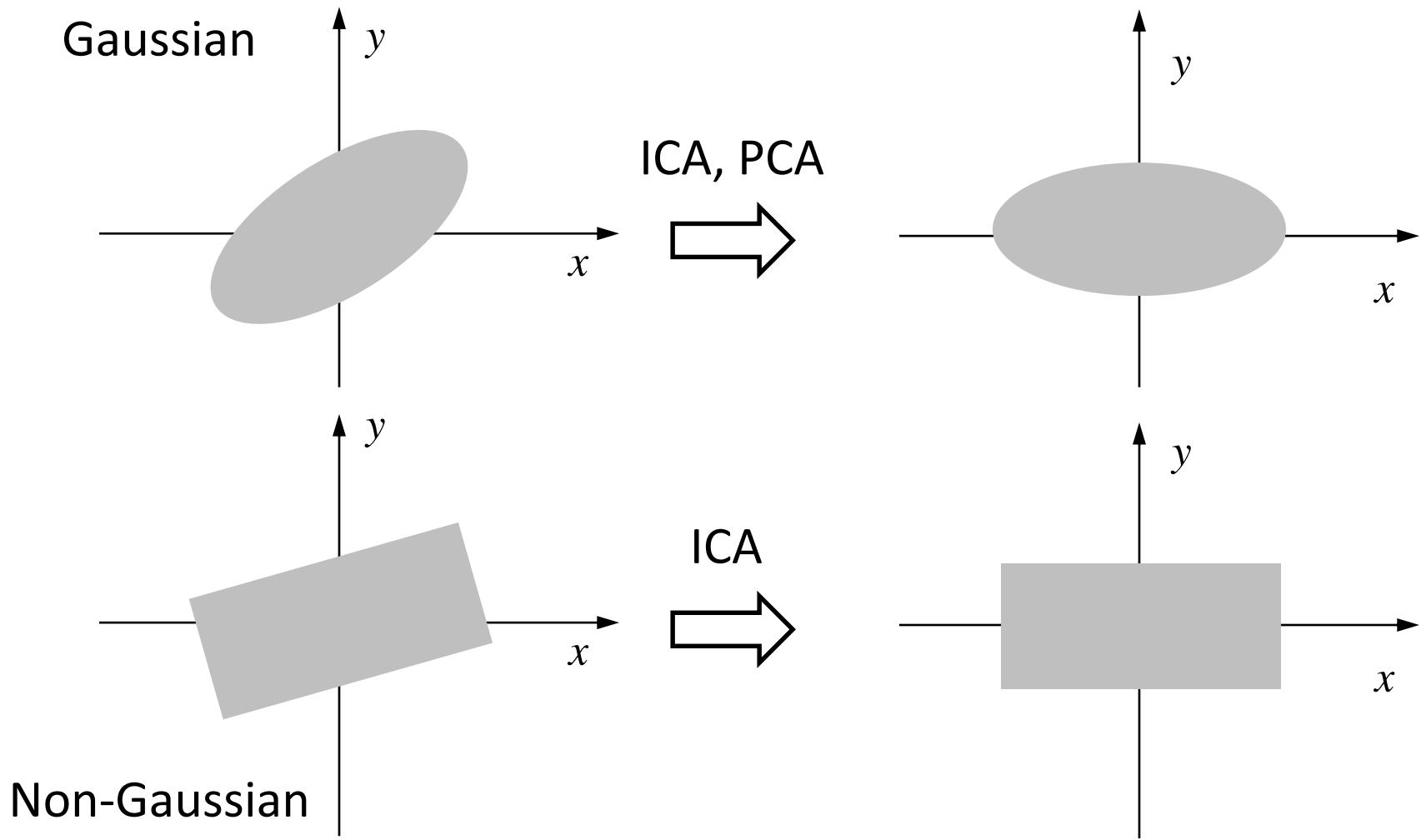
Independent Component Analysis



$$\min_A I([X_1, X_2, \dots, X_M] | A, Y)$$

After optimization, the components of X become as much independent as possible

Illustration of ICA Operation



Summary

- Pattern Recognition
 - Model Training and Regularization
 - Probabilities and Gaussian Distributions
 - Bayesian Inferences (ML and MAP)
 - Curse of Dimensionality
 - Decision Theory
 - Entropy and Information
-