Lecture 1

Statistical Machine Learning

CSE 342/542 – Winter 2022

Slides from Bishop and Duda

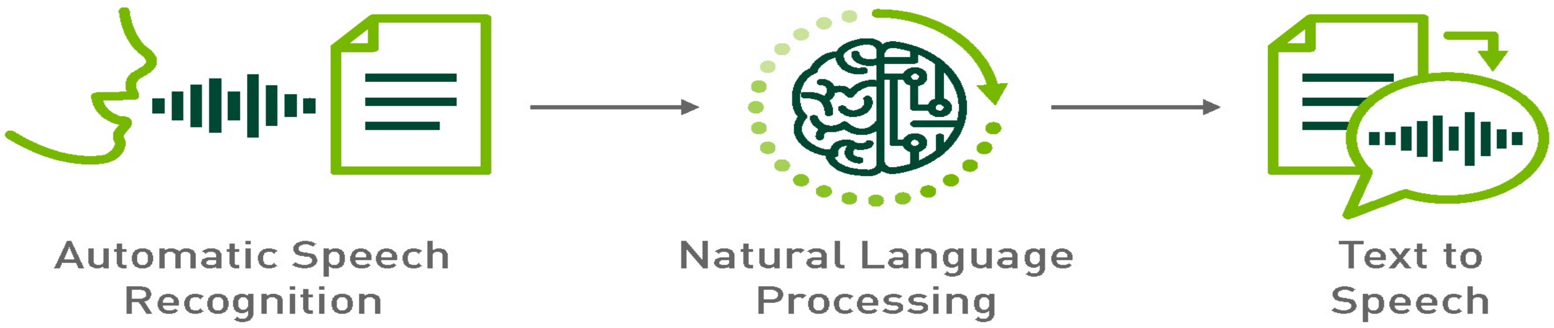
Statistical Learning vs. Machine Learning

Classical Statistics
Infer information from small datasets (Not enough data)
- time complexity, convergence

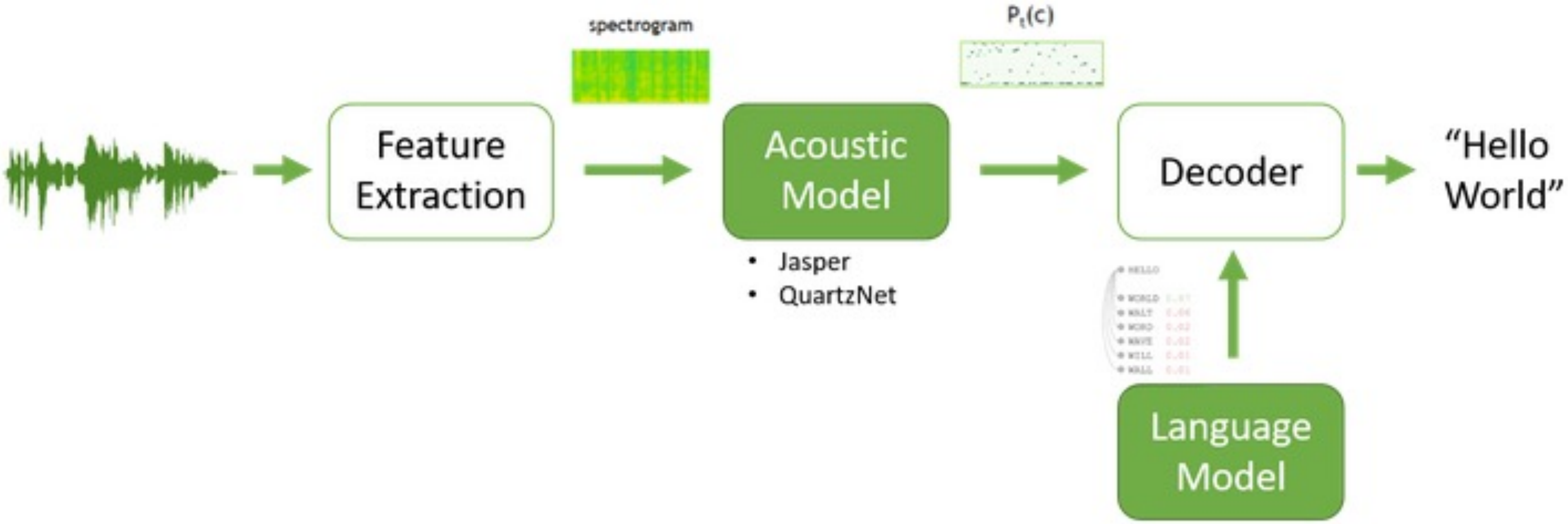
Machine Learning
Infer information from large datasets (Too many data)

Applications

- Search and recommendation (e.g. Google)
    - Represent a query using a feature and then match
- Automatic speech recognition and speaker verification
    - Convert speech to text. Match if two voices samples belong to same speaker



- Automatic speech recognition and speaker verification
  - Convert speech to text.
  - Match if two voices samples belong to same speaker



Facial recognition tool
'could help boost pigs' wellbeing'

Contd.

- Object re-id
  - Given a query sample, retrieve all instances of the same object
  - Potential use in surveillance and industrial application where multiple robots operate on same object and need to re-identify

Classification

Classification:
Predicting a discrete random variable Y from another random variable X.

Classification

$\mathcal{Y} = \{0, 1\} \rightarrow$ Binary classification

Consider data $(X_1, Y_1), \ldots, (X_n, Y_n)$ where

$$X_i = (X_{i1}, \ldots, X_{id}) \in \mathcal{X} \subset \mathbb{R}^d$$

is a d-dimensional vector and Y_i takes values in some finite set $\mathcal{Y}$. A **classification rule** is a function $h : \mathcal{X} \to \mathcal{Y}$. When we observe a new X, we predict Y to be $h(X)$.

Age, height, weight, grades of 2022 SML as train set
Test on students for 2023 whether or not they will take SML

Note:
Distribution is same as still the students are from IIITD and not from say some arbitrary primary school or from similar university but Law/Medical/Management school.

Test and train samples do not overlap.

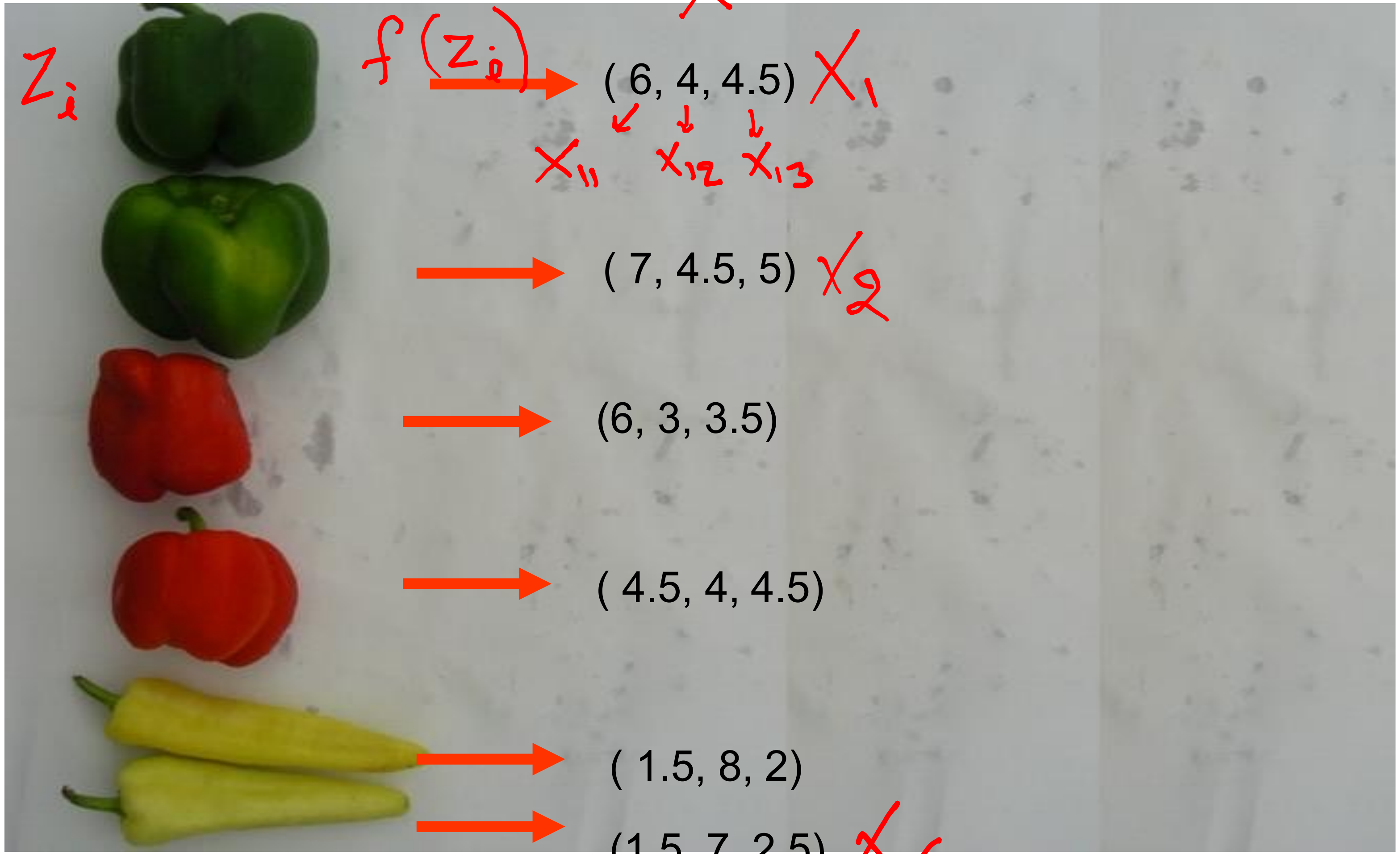
PhD defense – pre-processing of entire set.

Data

$f(z_i) \in \mathbb{R}^3$

Features (X)

X



Features and labels

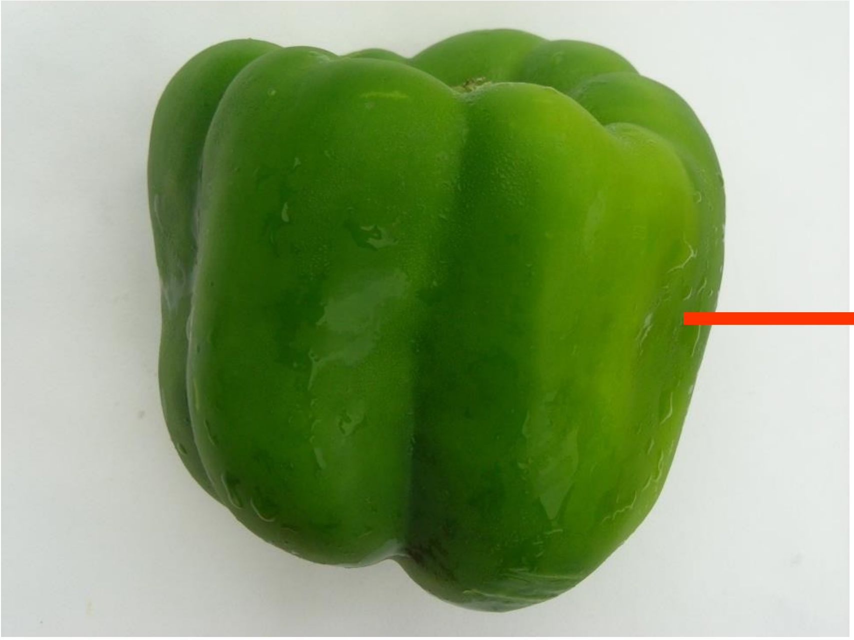
$h(X) \longrightarrow y$ $\qquad\qquad$ y

→ (6, 4, 4.5)	→ **Green Pepper**	0
→ (7, 4.5, 5)	→ **Green Pepper**	
→ (6, 3, 3.5)	→ **Red Pepper**	1
→ (4.5, 4, 4.5)	→ **Red Pepper**	
→ (1.5, 8, 2)	→ **Hot Pepper**	2
→ (1.5, 7, 2.5)	→ **Hot Pepper**	

Classification (New point)

$\xrightarrow{f}$ (7, 4, 4.5) $\xrightarrow{h(7,\ 4,\ 4.5)}$?

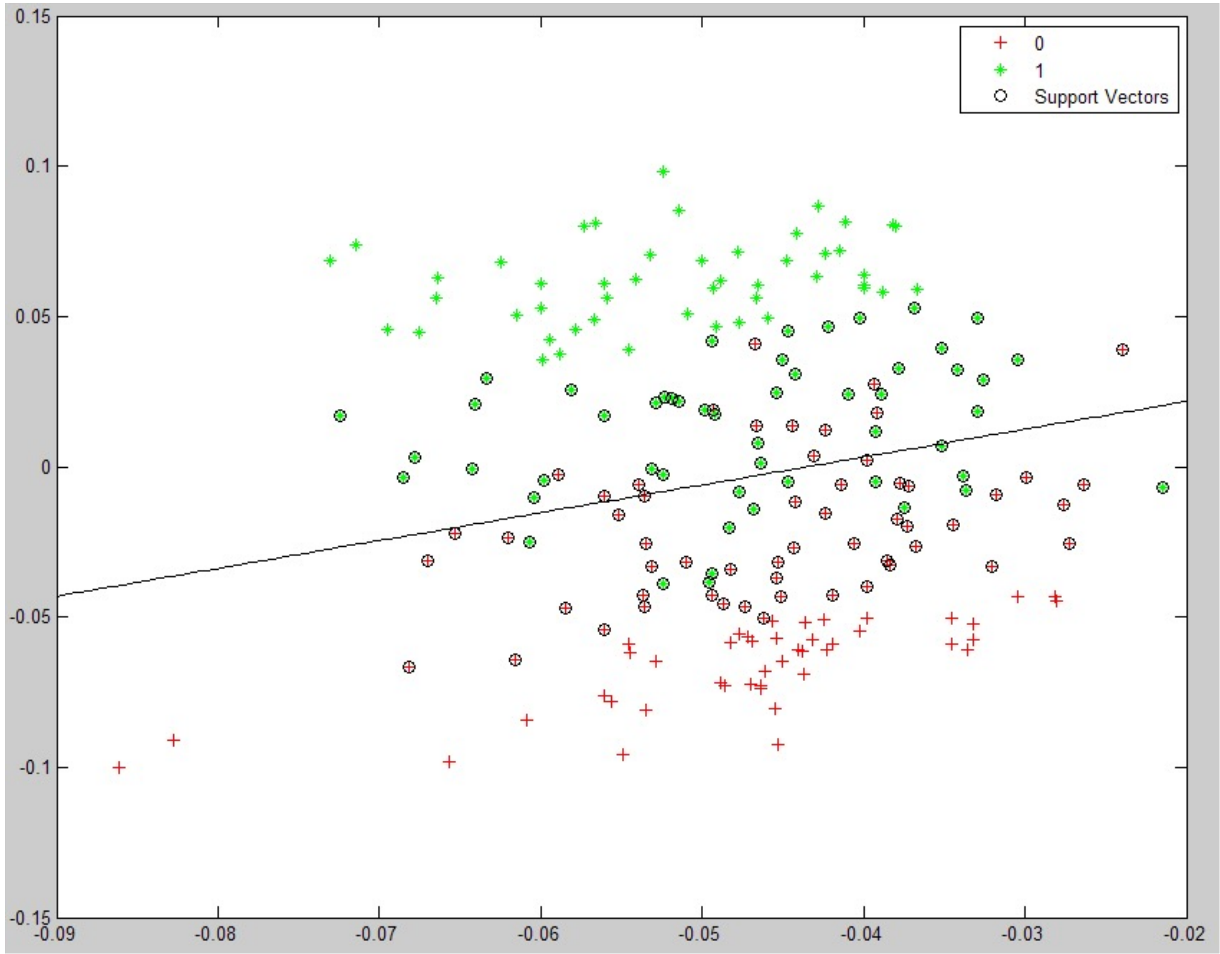
Classification (New point)



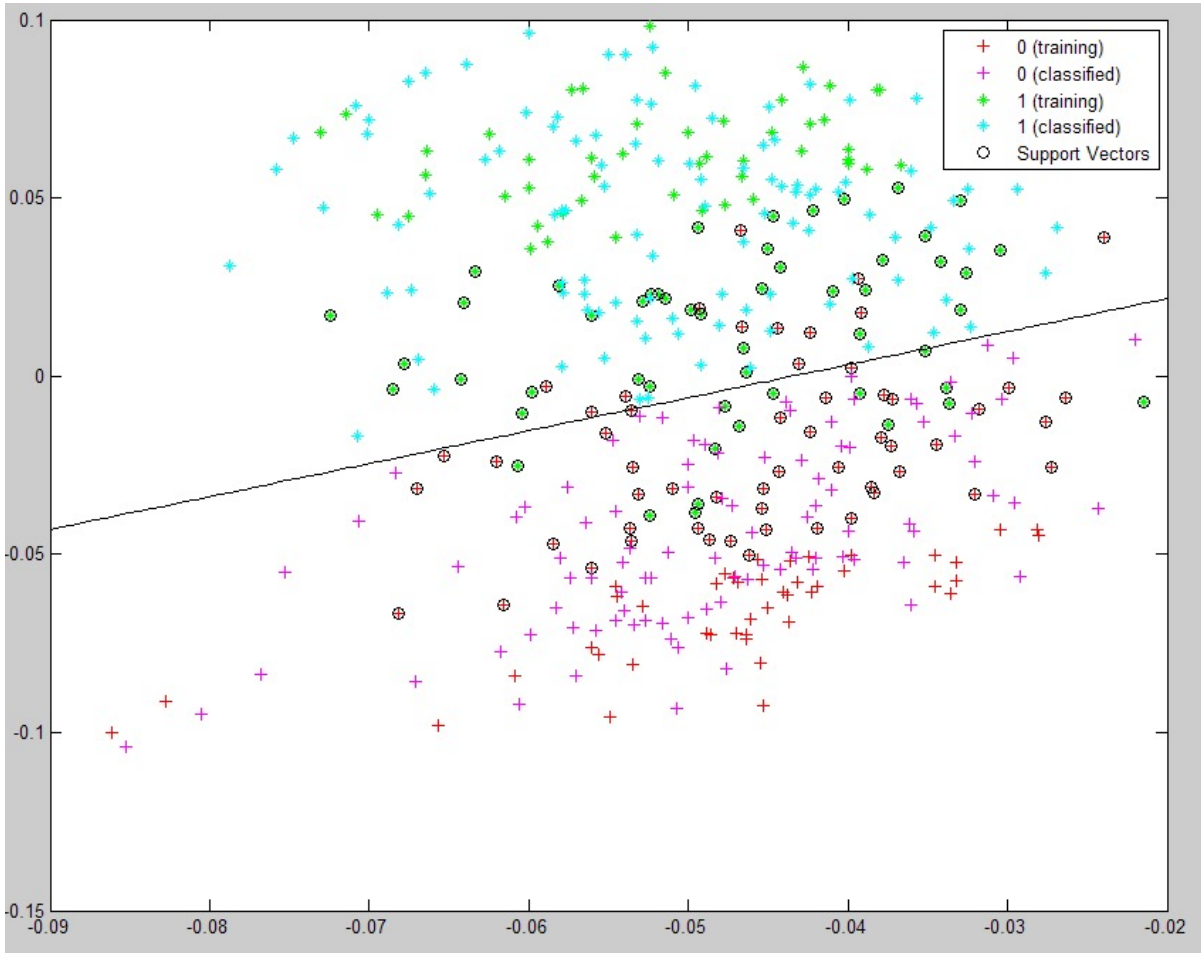
(6, 4, 4.5)

$$h(6, 4, 4.5) \longrightarrow ?$$

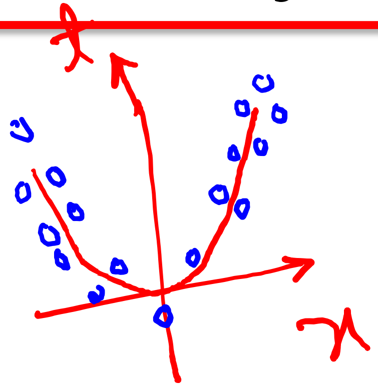
Classification

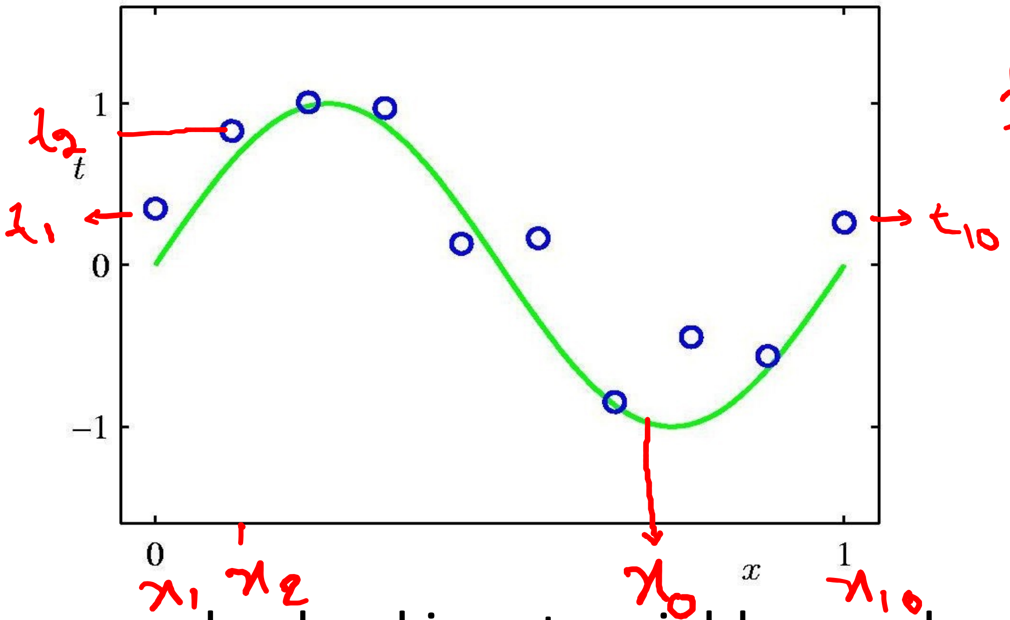


Classification



Polynomial Curve Fitting / Regression.





$t \in \mathbb{R}$

Suppose we observe a real-valued input variable x and we wish to use this observation to predict the value of a real-valued target variable t.

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

$w^T x$

Closed form solution

$W^T X$

$$\begin{cases} t_1 = a x_1^2 + b x_1 + c \\ t_2 = a x_2^2 + b x_2 + c \\ \vdots \\ t_N = a x_N^2 + b x_N + c \end{cases}$$

$x_0 \rightarrow$ what is t_0?

$t_0 \leftarrow a x_0^2 + b x_0 + c$

$W = [w_0 \; w_1 \cdots\cdots w_M]^T$

$W \in R^{M+1}$

$X = [1 \; x \; x^2 \ldots x^M]^T$

for any i^{th} point

$y_i = W^T X_i$

$y_i = X_i^T W$

Other way

We want the prediction to be close to true value. What can we do?

Other way

We want the prediction to be close to true value. What can we do?

- Create a model as a function of variables and inputs
- Define an error function – distance, cost, loss, between prediction and true value
- Minimize error to learn variables
- Use the learnt variables to predict for unseen points

error function: $\phi(W)$

Given

Now suppose that we are given a training set comprising N observations of x, written **x** ≡ (x_1 , . . . , x_N)’ , together with corresponding observations of the values of t,denoted **t**≡(t_1,...,t_N)’ .

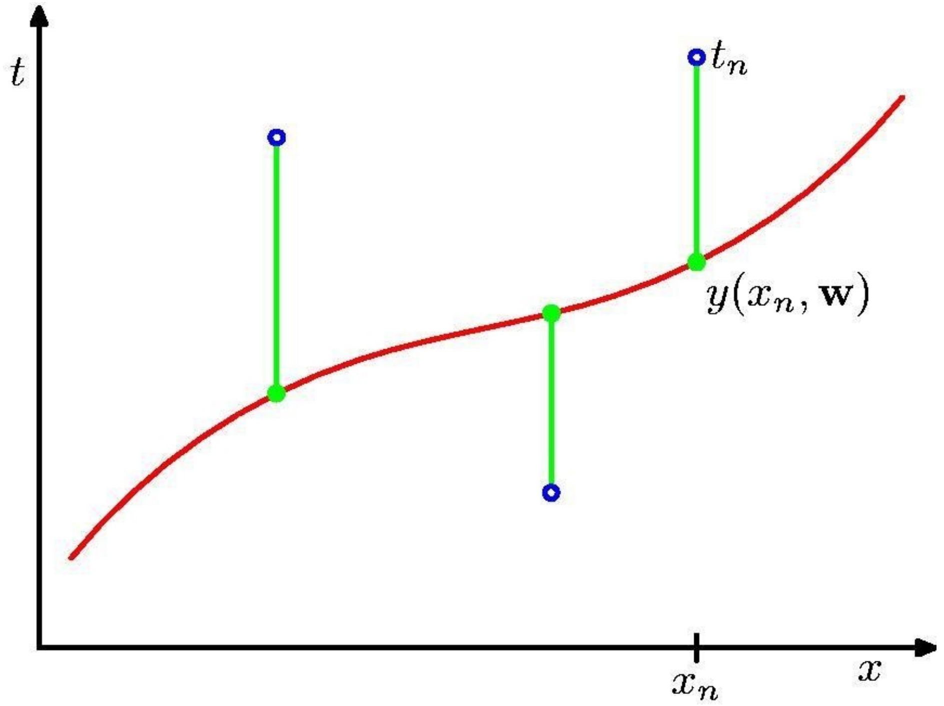
N = 10 data points.

The input data set **x** was generated by choosing values of x_n, for n = 1,...,N, spaced uniformly in range [0,1], and the target data set **t** was obtained by first computing the corresponding values of the function sin(2πx) and then adding a small level of random noise having a Gaussian distribution

Goal

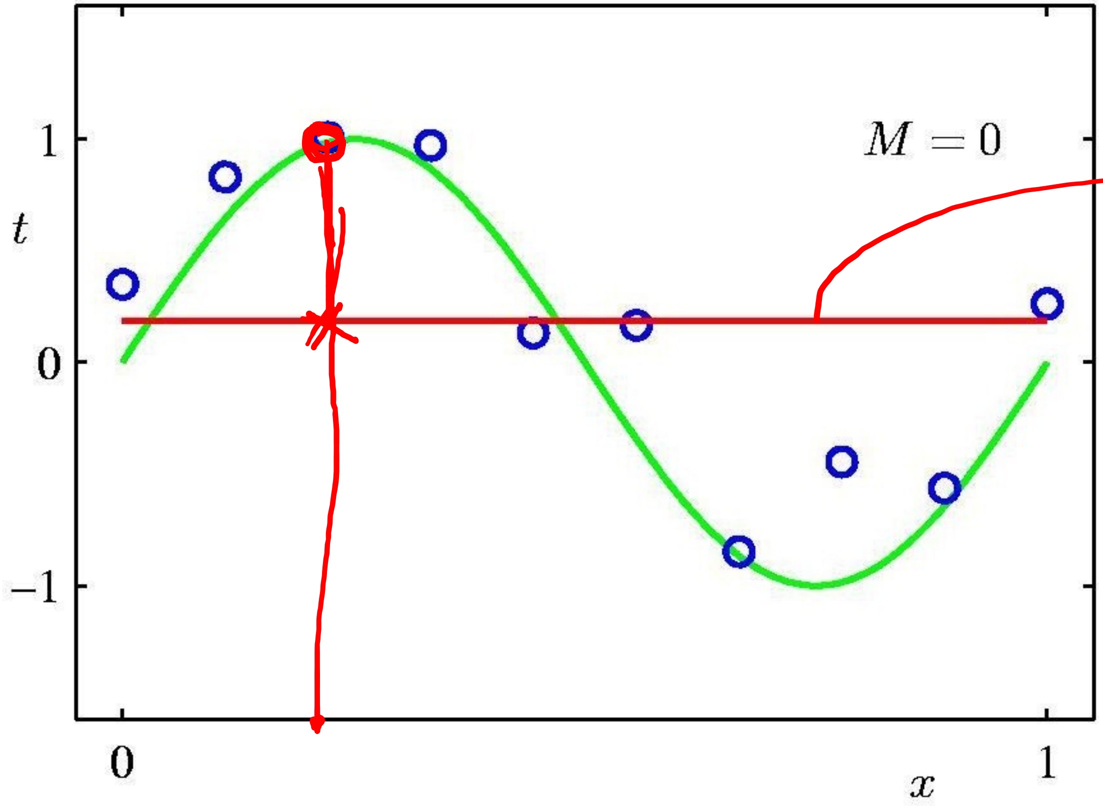
Our goal is to exploit this training set in order to make predictions of the value t of the target variable for some new value x of the input variable.

Sum-of-Squares Error Function



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

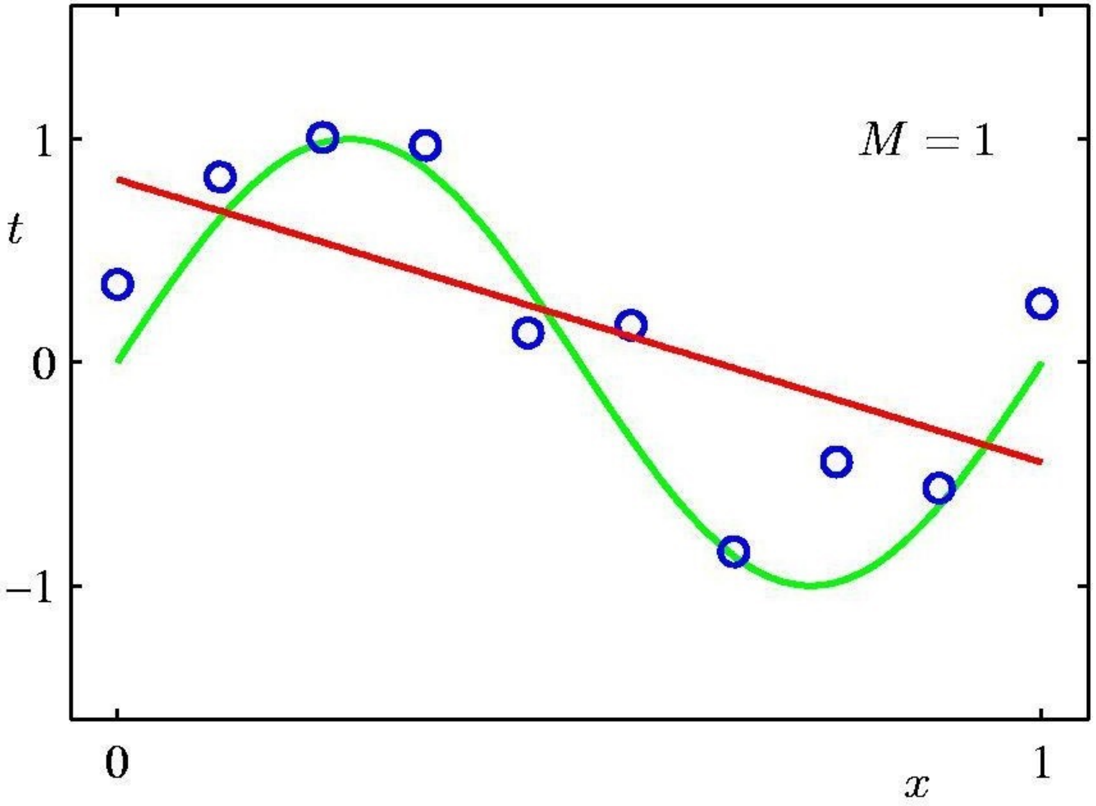
0^{th} Order Polynomial



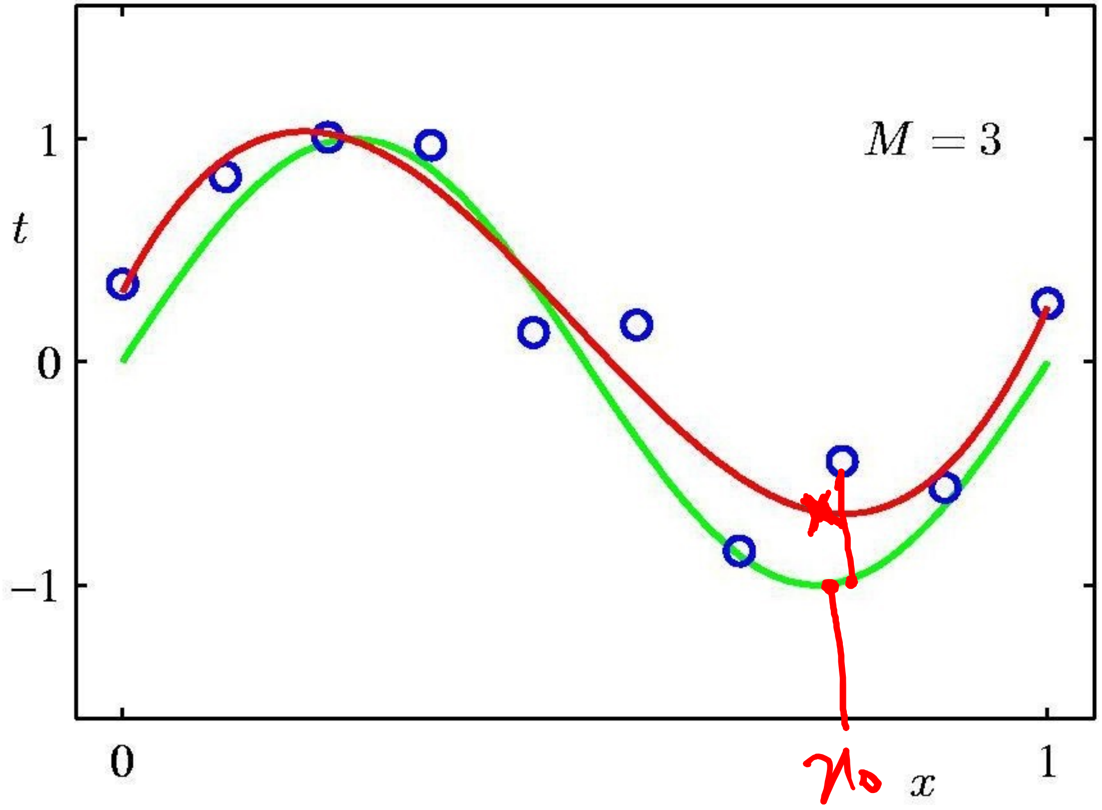
Prediction

$w_0 = \text{mean}(t)$

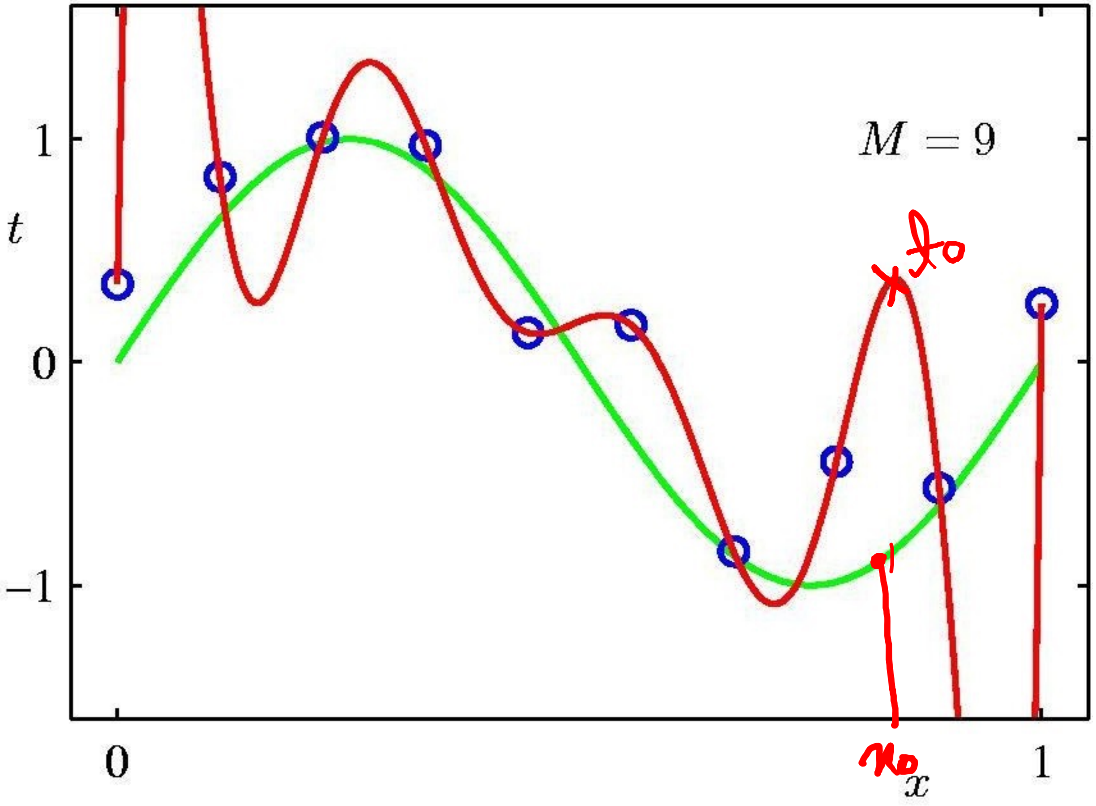
1st Order Polynomial



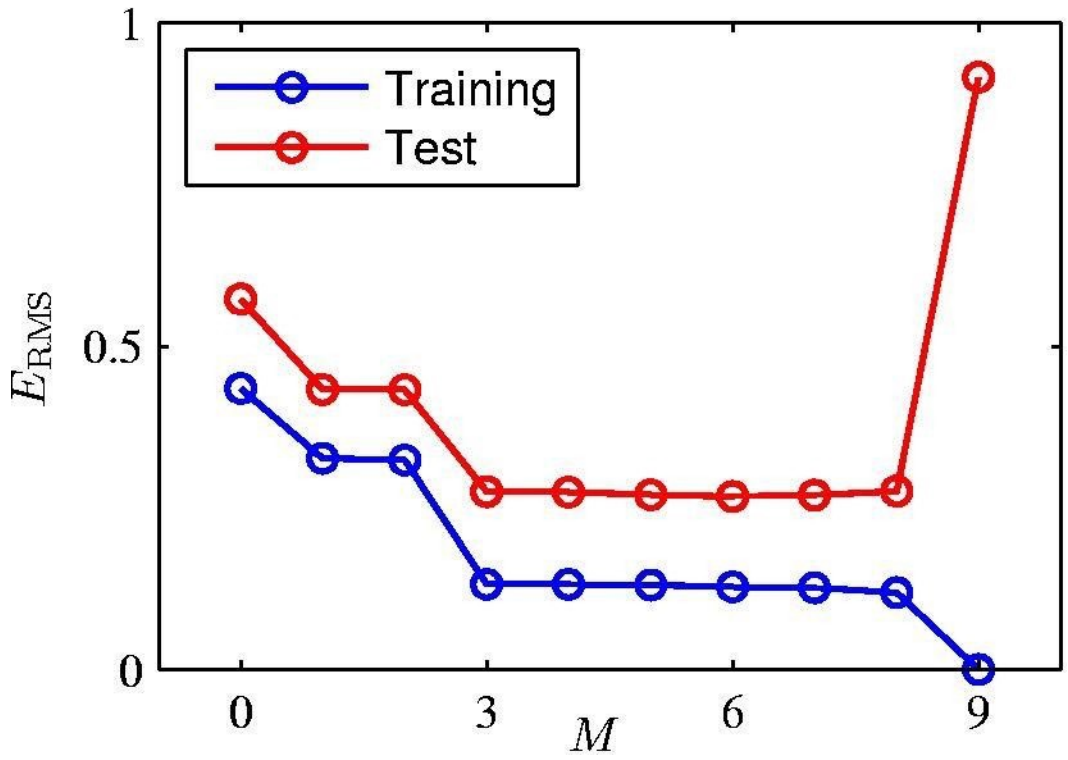
3rd Order Polynomial



9th Order Polynomial



Over-fitting

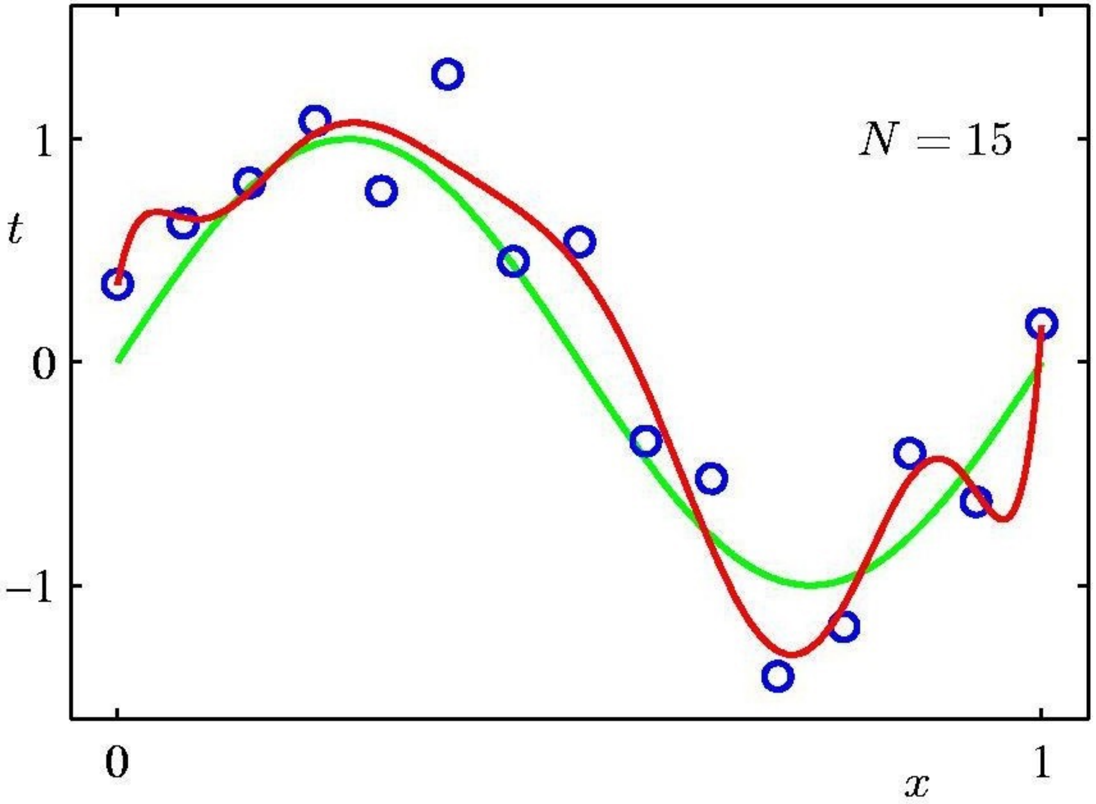


Polynomial Coefficients

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	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

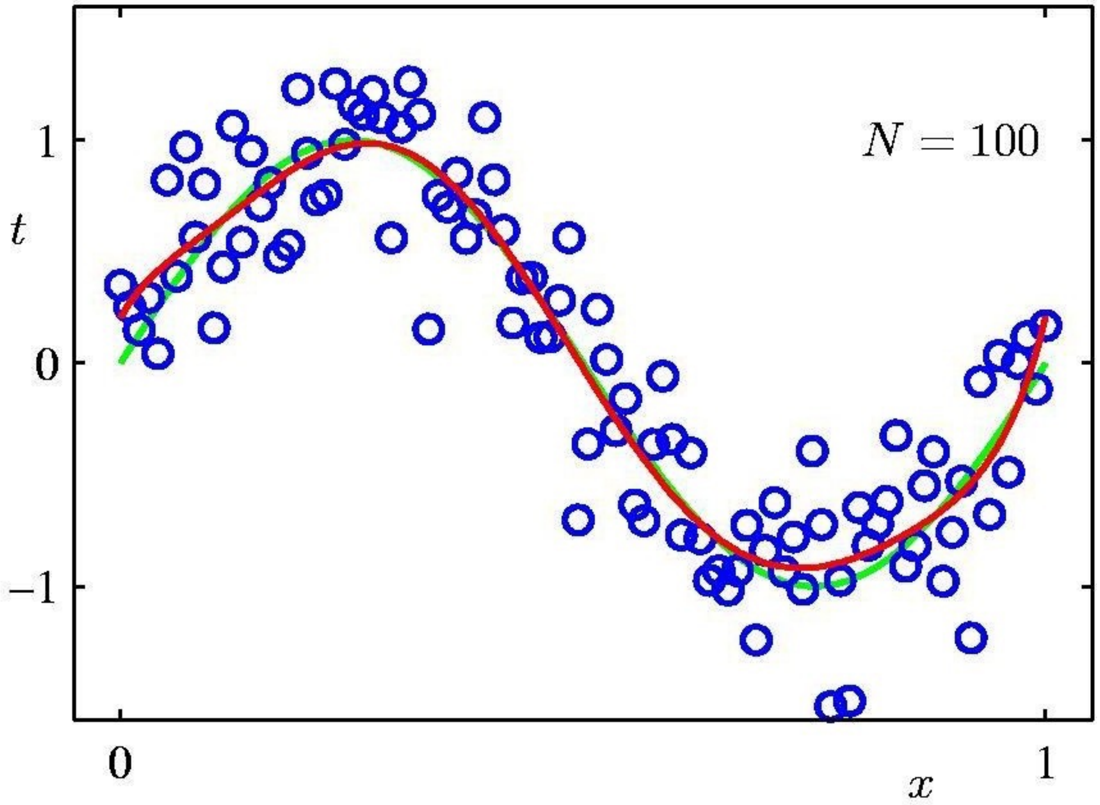
Data Set Size: $N = 15$

9th Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial



Regularization

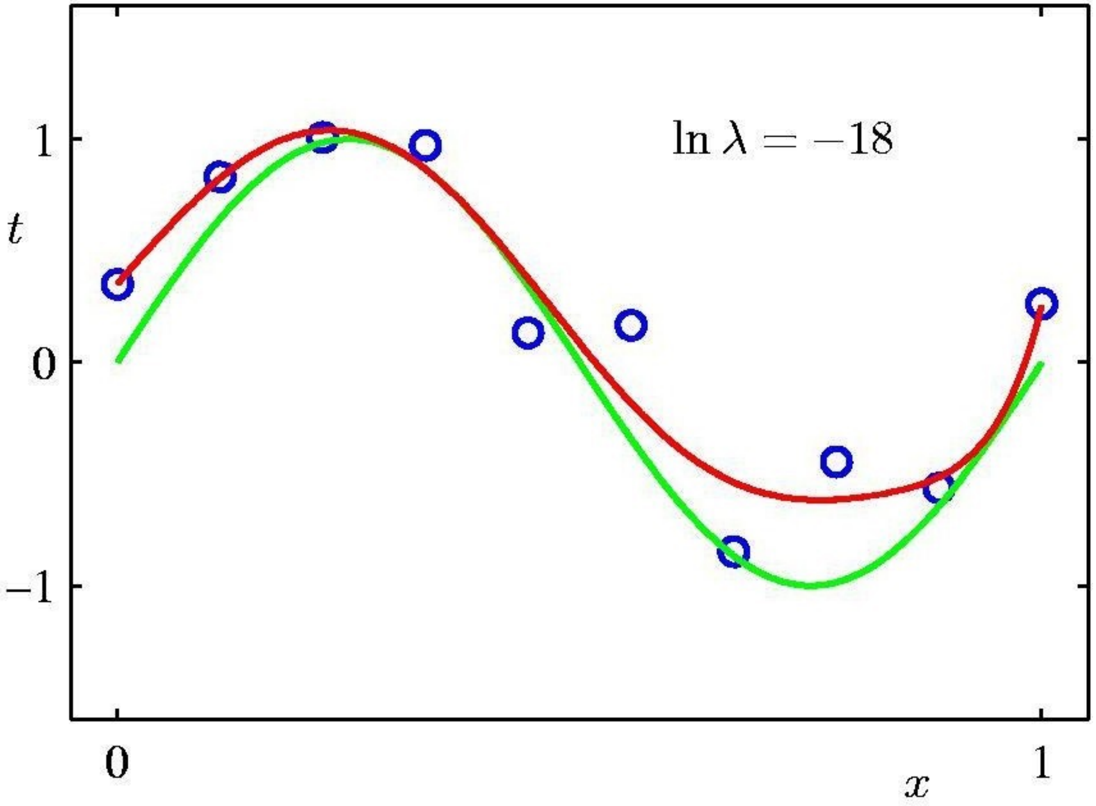
As large coefficients lead to huge change with a very small change in x, what should we do?

Regularization

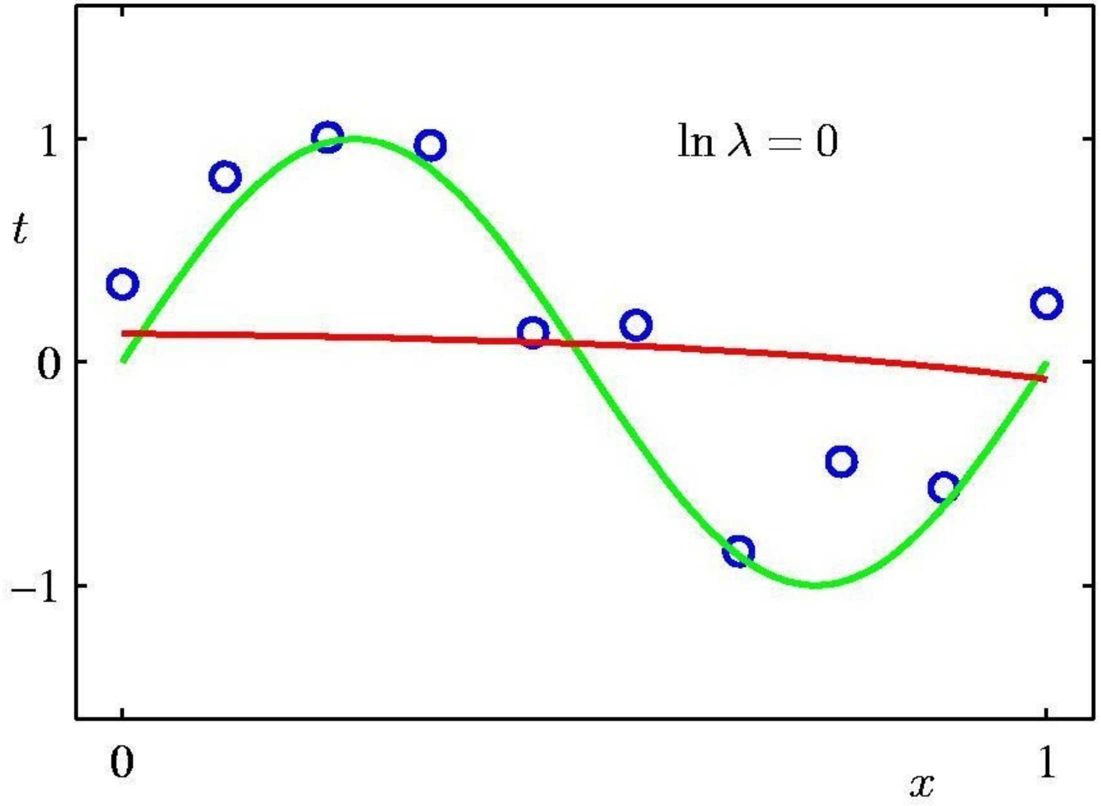
Penalize large coefficient values

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

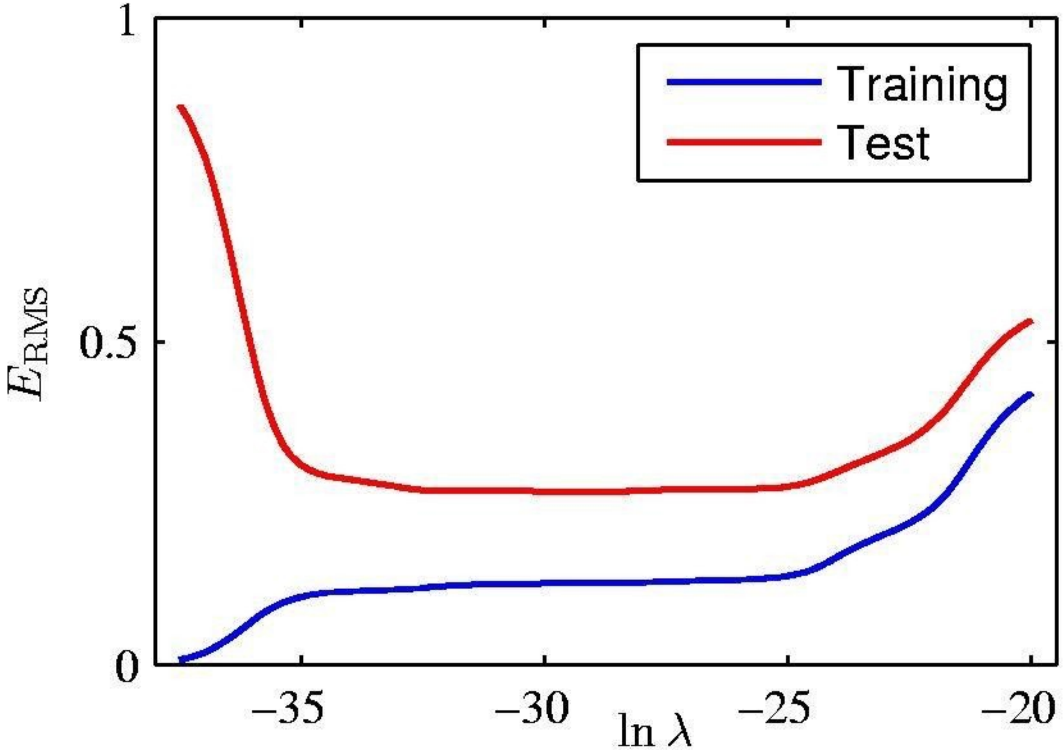
Regularization: $\ln \lambda = -18$



Regularization: $\ln \lambda = 0$



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

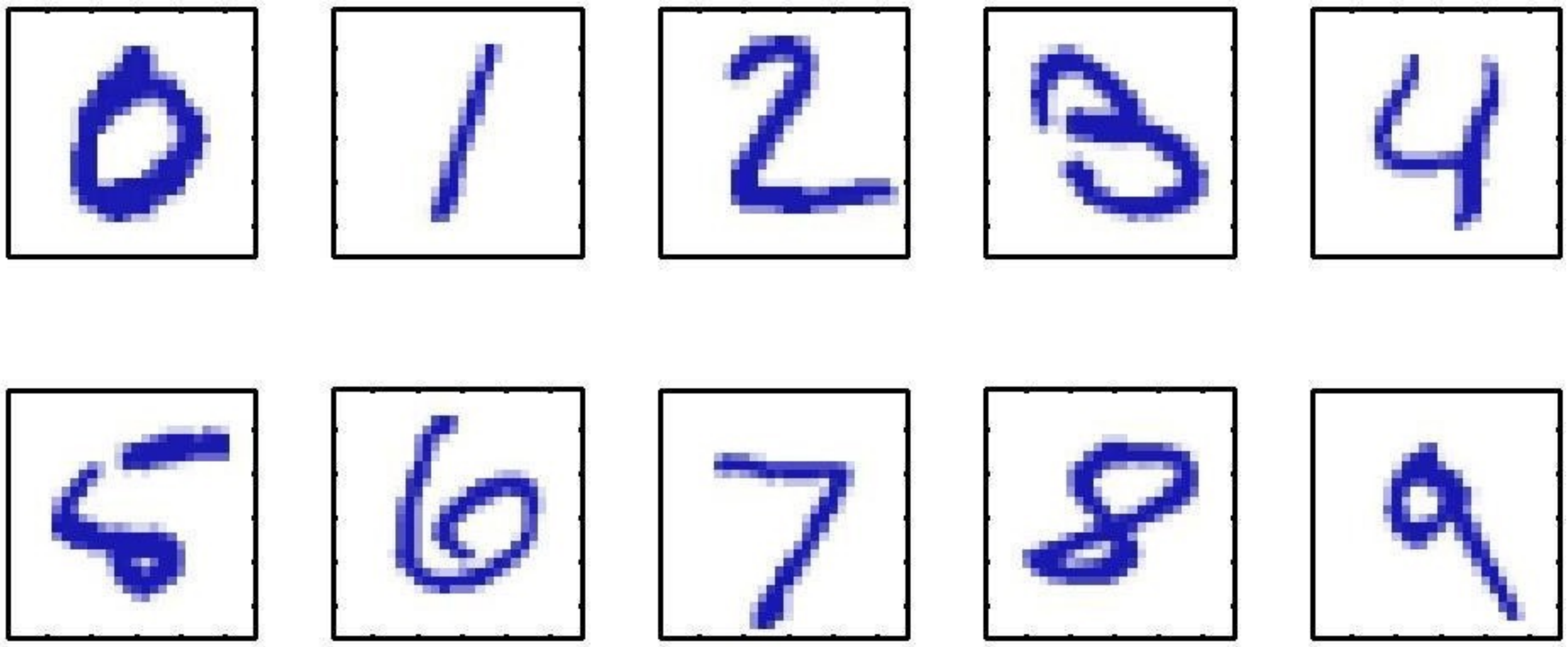


Polynomial Coefficients

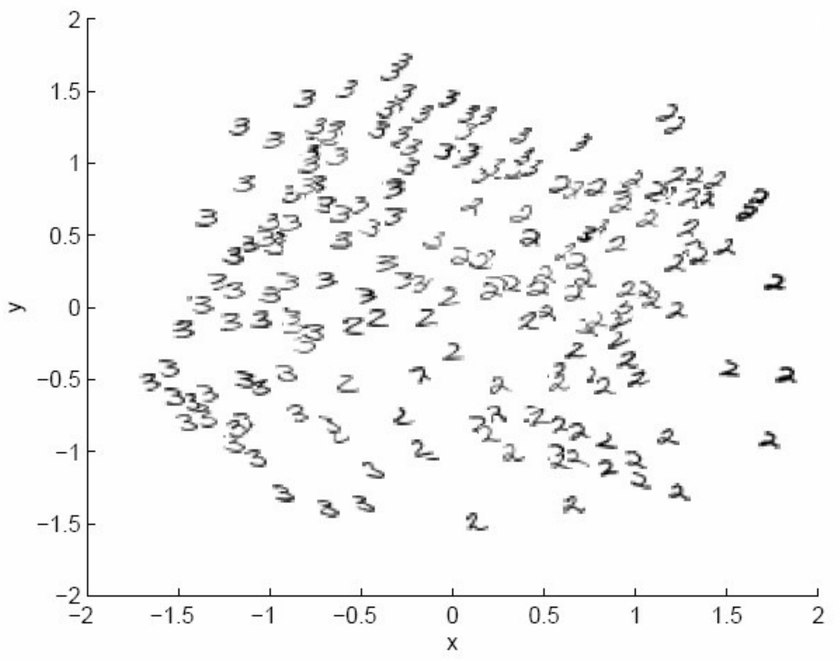
	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$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

Example

Handwritten Digit Recognition



Unsupervised Learning (Only data, no labels)

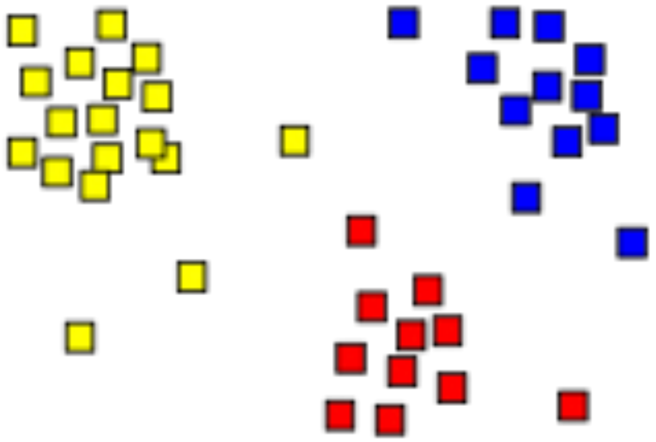


A canonical dimensionality reduction problem from visual perception. The input consists of a sequence of 64-dimensional vectors, representing the brightness values of 8 pixel by 8 pixel images of digits 2 and 3. Applied to $n = 400$ raw images. A two-dimensional projection is shown, with the original input images.

Clustering

Organizing data into clusters such that there is

- high intra-cluster similarity
- low inter-cluster similarity
- Informally, finding natural groupings among objects.



Tentative topics

- Feature extraction (FDA, PCA)
- Bayes Decision theory and Error bounds (Chernoff, Bhattacharya, Hoeffding)
- Parameter estimation (MLE/MAP)
- Linear classifier (Discriminant analysis, LDA, QDA)
- Gaussian processes (Regression)
- Neural networks (FFNN, weight decay, regularization)
- Deep Learning
- Bagging (reducing var)
- Boosting (reducing bias) – AdaBoost, Gradient Boosting
- Clustering (spectral clustering, min/ratio cut)

Reference books

- The recommended books that cover the similar material are:
- Hastie, Tibshirani, Friedman

Elements of Statistical Learning.

- Bishop

Pattern Recognition and Machine Learning.

- Murphy

Machine Learning: a Probabilistic Perspective

- *Duda*

Pattern Classification

Prereq

Prob and Stats
Python/Matlab
Vector calculus (desirable)

COs

- Students will be able to understand the various key paradigms for machine learning and pattern classification
- Students will be able to apply suitable feature extraction and classification technique to solve a given pattern classification problem
- Students will be able to design a complete machine learning/pattern classification algorithm and evaluate the performance

Evaluation

- Assignment (50%) (5)
- Quiz 20% (3)
- Midsem 15%
- Endsem 15%

- All mandatory

Further Reading

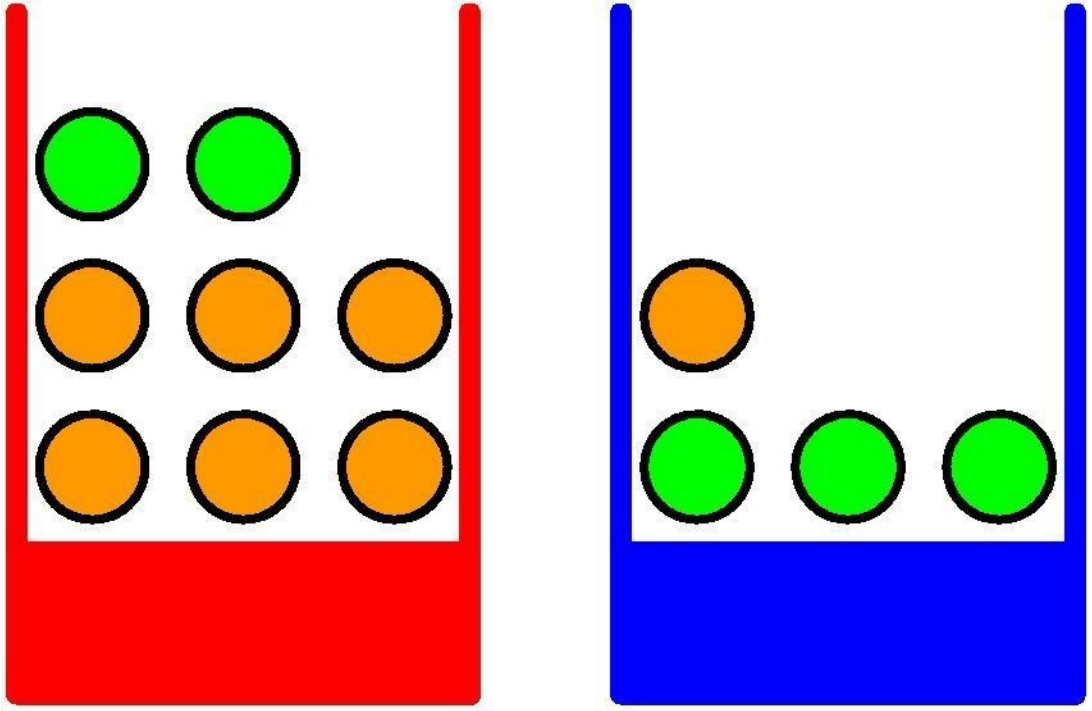
- Theoretical: AISTATS, ICML, JMLR, NeurIPS
- Systems + theory: CVPR, ICCV, ECCV, AAAI, IEEE Transactions

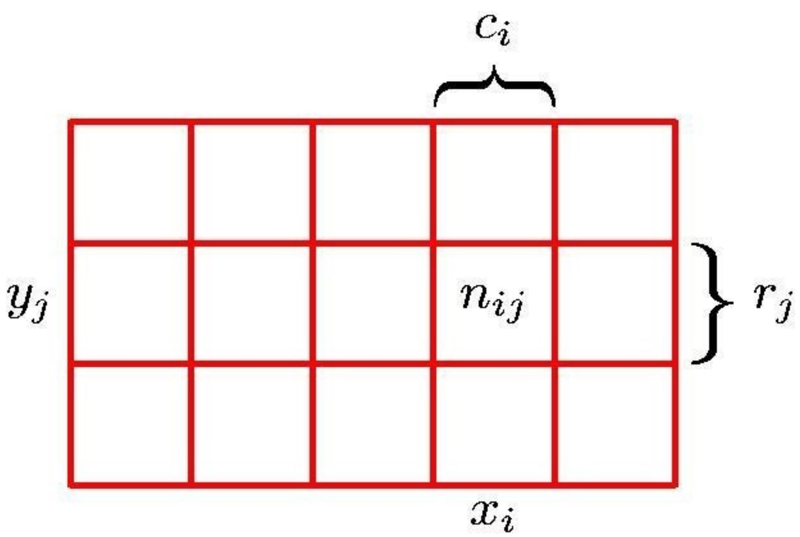
- 91-100 A/A+
- 81-90 A-
- 71-80 B

- <30 F

Probability Theory

Apples and Oranges



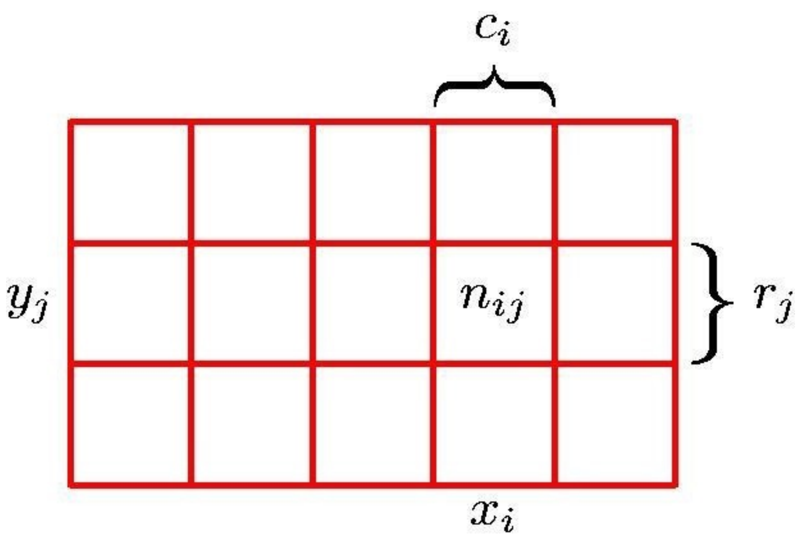


Let X and Y be random variables.

Let their be N trial during which we sample both of variables X and Y.

Let the number of times X=xi and Y=yj occur is nij.

Probability Theory



Marginal Probability

$$p(X = x_i) = \frac{c_i}{N}.$$

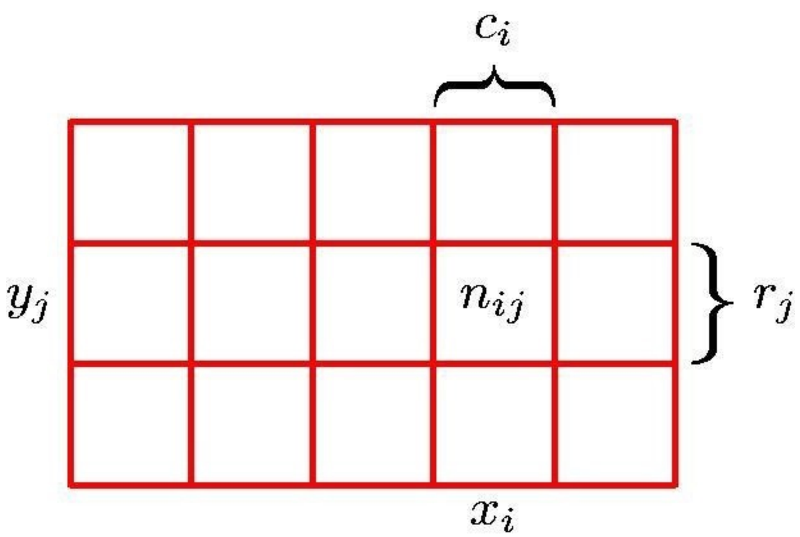
Joint Probability

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

Conditional Probability

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

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

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

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

Ex. Consider red and blue boxes

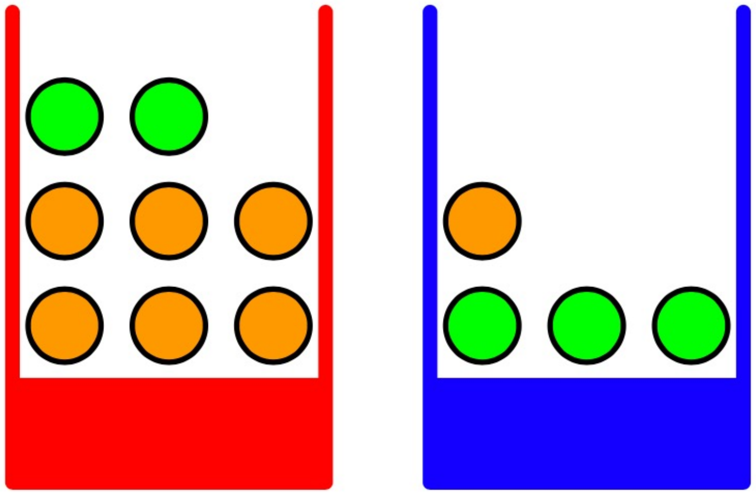
$p(B = r) = 4/10$ and $p(B = b) = 6/10$

$p(F = a|B = r) = 1/4$

$p(F = o|B = r) = 3/4$

$p(F = a|B = b) = 3/4$

$p(F = o|B = b) = 1/4.$



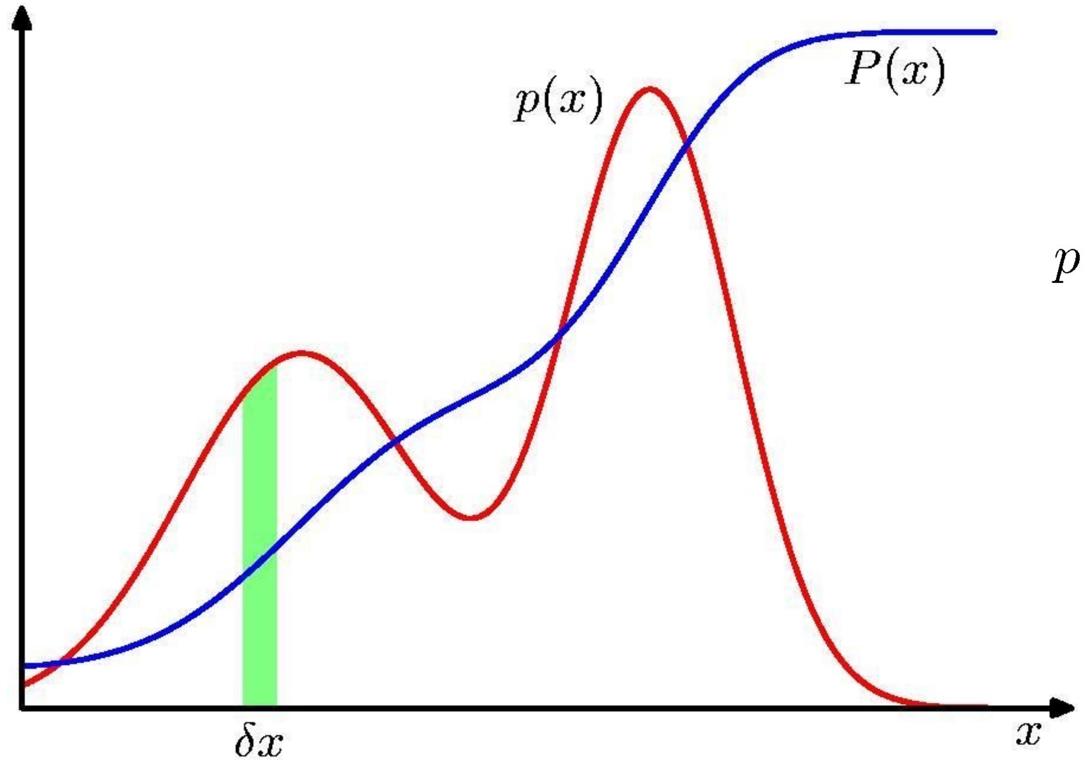
Contd.

We are told that a piece of fruit has been selected and it is an orange, and we would like to know which box it came from.

Bayes: $p(B=r|F=o) = p(F = o|B = r)\ p(B=r)\ /\ p(F=o)$

$p(F = o) = p(F = o|B=r)p(B=r)+p(F = o|B=b)p(B=b)$

Probability Densities



$$p(x \in (a,b)) = \int_a^b p(x)\,\mathrm{d}x$$

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

$$p(x) \geqslant 0 \qquad \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1$$

Expectations

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

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

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

Conditional Expectation
(discrete)

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

Approximate Expectation
(discrete and continuous)

Ex. A uniform pdf in (-a,a).