# Machine Learning

4771

Instructors:
Adrian Weller and Ilia Vovsha

# Topic 1

- Introduction: Instructors and TAs
- Machine Learning: What, Why and Applications
- Syllabus, policies, texts, web page
- Historical Perspective
- Machine Learning Tasks and Tools
- Digit Recognition Example
- Different Approaches

# Machine Learning: What/Why

Statistical Data-Driven Computational Models

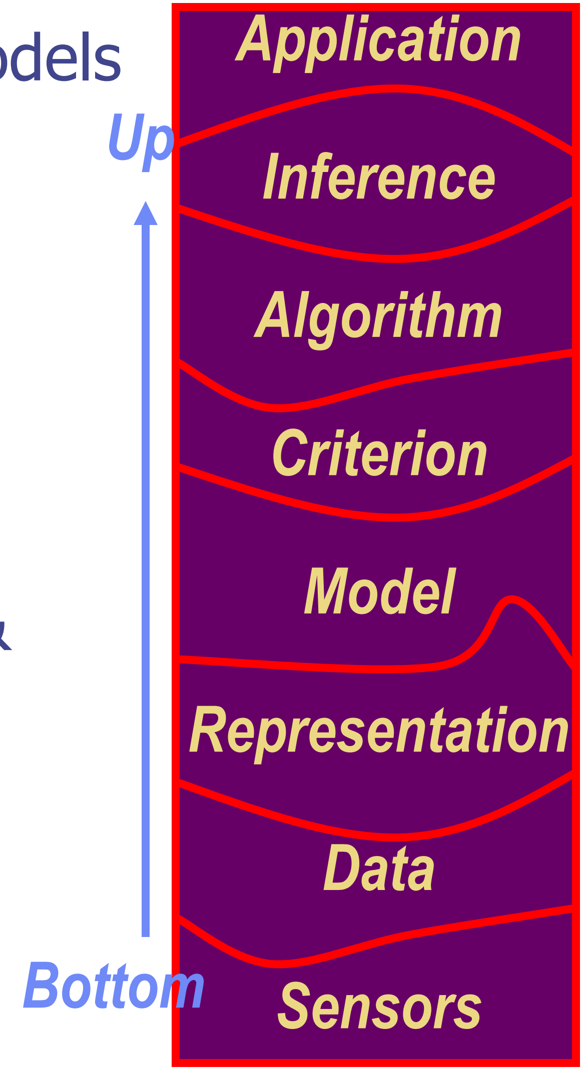
Real domains (vision, speech, behavior):
  no $E=MC^2$
  noisy, complex, nonlinear
  have many variables
  non-deterministic
  incomplete, approximate models

Need: statistical models driven by data & sensors, a.k.a Machine Learning

Bottom-Up: use data to form a model

Why? Complex data everywhere, audio, video, internet

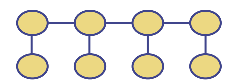
Intelligence ~ Learning ~ Prediction



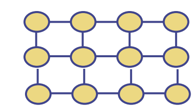
# Machine Learning Applications

- ML: Interdisciplinary (CS, Math, Stats, Physics, OR, Psych)
- Data-driven approach to AI
- Many domains are too hard to do manually

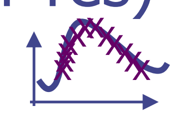
Speech Recognition (HMMs, ICA)



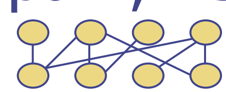
Computer Vision (face rec, digits, MRFs, super-res)



Time Series Prediction (weather, finance)



Genomics (micro-arrays, SVMs, splice-sites)
NLP and Parsing (HMMs, CRFs, Google)
Text and InfoRetrieval (docs, google, spam, TSVMs)
Medical (QMR-DT, informatics, ICA)



Behavior/Games (reinforcement, gammon, gaming)

# Course Details & Requirements

- Probability/Stats, Linear Algebra, Calculus, AI
- Mathematical & Data Driven approach to AI
- Lots of Equations!
- Texts: Introduction to Graphical Models
  by M. Jordan & C. Bishop (Online)
  Pattern Classification (3rd Edition)
  by Duda, Hart and Stork
  Pattern Recognition & Machine Learning
  by C. Bishop (Spring 2006 Edition)
- Homework: Every 2-3 weeks
- Grading: homework, midterm & final examination
- Software Requirements: Matlab software, account

# Course Web Page

**http://www.cs.columbia.edu/~coms4771**

**Slides will be available on handouts web page**

**Each week, check NEWS link for readings, homework deadlines, announcements, etc.**

**We encourage:**

**Post questions, topics etc. to the Courseworks Bulletin Board**

**Find study partner(s) but write up work individually**

# Syllabus

- **Intro to Machine Learning**
- **Bayesians & Frequentists**
- **Least Squares Estimation**
- **Logistic Regression**
- **Perceptrons**
- **Neural Networks**
- **Statistical Learning Theory**
- **Support Vector Machines**
- **Kernels**
- **Probability Models**
- **Maximum Likelihood**
- **Multinomial Models**
- **Bernoulli Models**
- **Gaussian Models**
- **Principal Components Analysis**
- **Bayesian Inference**
- **Exponential Family Models**
- **Mixture Models**
- **K-means**
- **Expectation Maximization**
- **Graphical Models**
- **Bayesian Networks**
- **Junction Tree Algorithm**
- **Hidden Markov Models**

# Historical Perspective (Bio/AI)

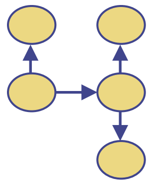
- 1917: Karel Capek (Robot)
- 1943: McCullogh & Pitts (Bio, Neuron)
- 1947: Norbert Weiner (Cybernetics, Multi-Disciplinary)
- 1949: Claude Shannon (Information Theory)
- 1950: Minsky, Newell, Simon, McCarthy (Symbolic AI, Logic)
- 1957: Rosenblatt (Perceptron)
- 1959: Arthur Samuel
  Coined Machine Learning
  Learning Checkers

- 1969: Minsky & Papert (Perceptron Linearity, no XOR)
- 1974: Werbos (BackProp, Nonlinearity)
- 1986: Rumelhart & McLelland (MLP, Verb-Conjugation)
- 1980's: NeuralNets, Genetic Algos, Fuzzy Logic, Black Boxes

# Historical Perspective (Stats)

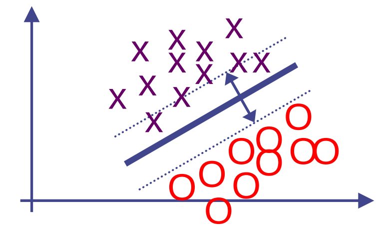
- 1763: Bayes (Prior, Likelihood, Posterior)
- 1920's: Fisher (Maximum Likelihood)
- 1937: Pitman (Exponential Family)
- 1969: Jaynes (Maximum Entropy)
- 1970: Baum (Hidden Markov Models)
- 1978: Dempster (Expectation Maximization)
- 1980's: Vapnik (VC-Dimension)
- 1990's: Lauritzen, Pearl (Graphical Models)

- 2000's: Bayesian & Statistical & Structure & Priors
  Graphical Models: Expectation Maximization,
  Kalman Filtering, Hidden Markov Models,
  Sigmoid Belief Nets, Markov Random Fields
  SVMs, Learning Theory, Boosting, Kernels

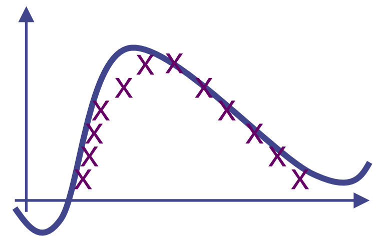


# Machine Learning Tasks

## Classification y=sign(f(x))

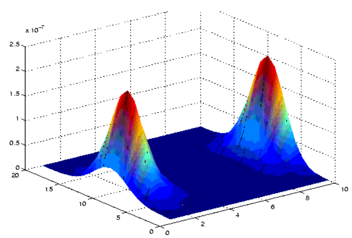


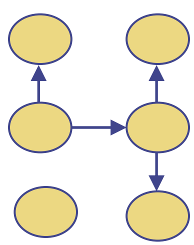
## Regression y=f(x)



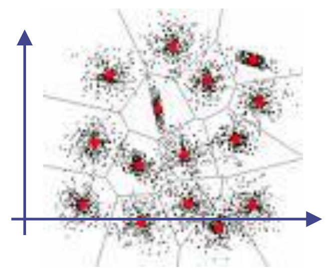
Supervised

## Modeling p(x)

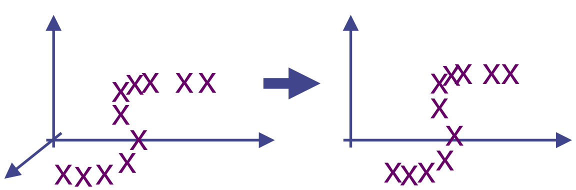




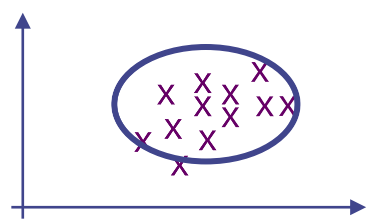
## Clustering



## Feature Selection

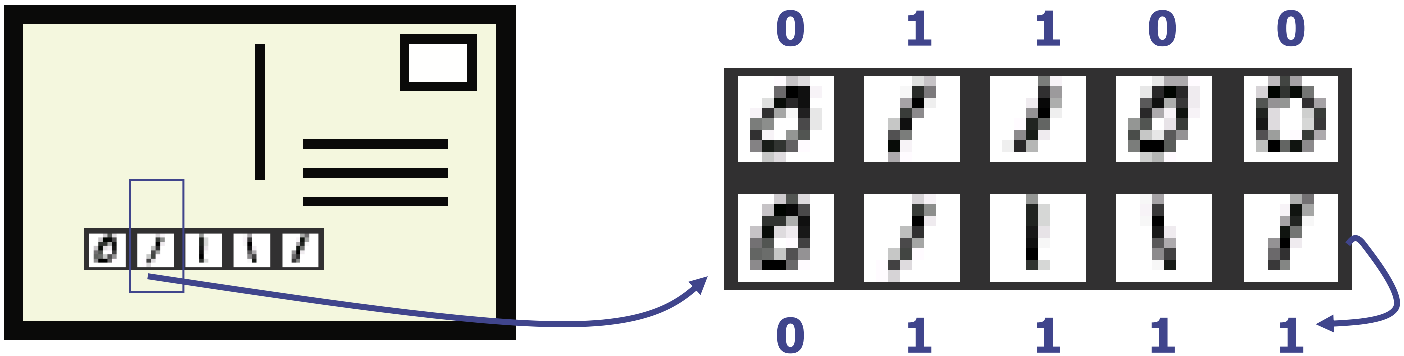


## Detection p(x)<t

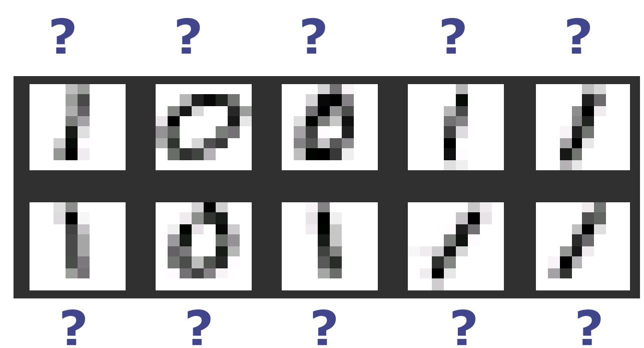


Unsupervised

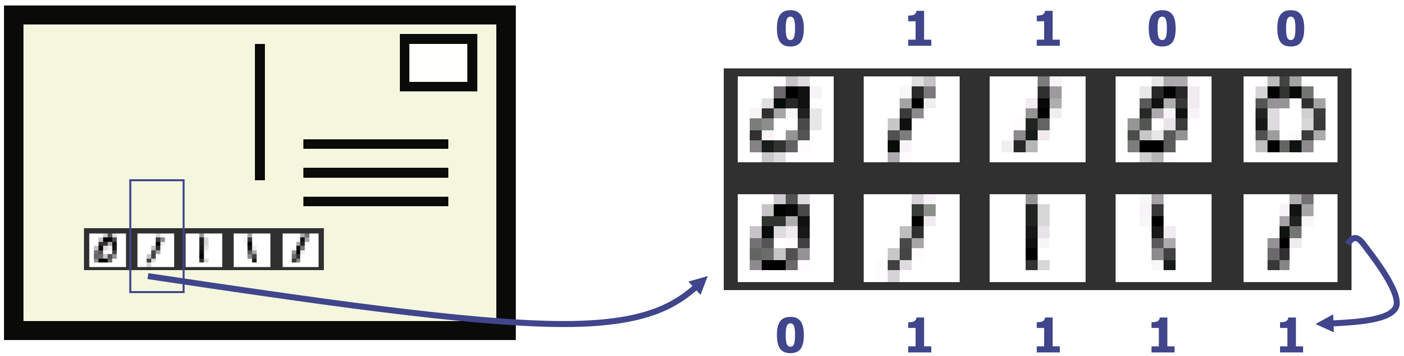
# ML Example: Digit Recognition

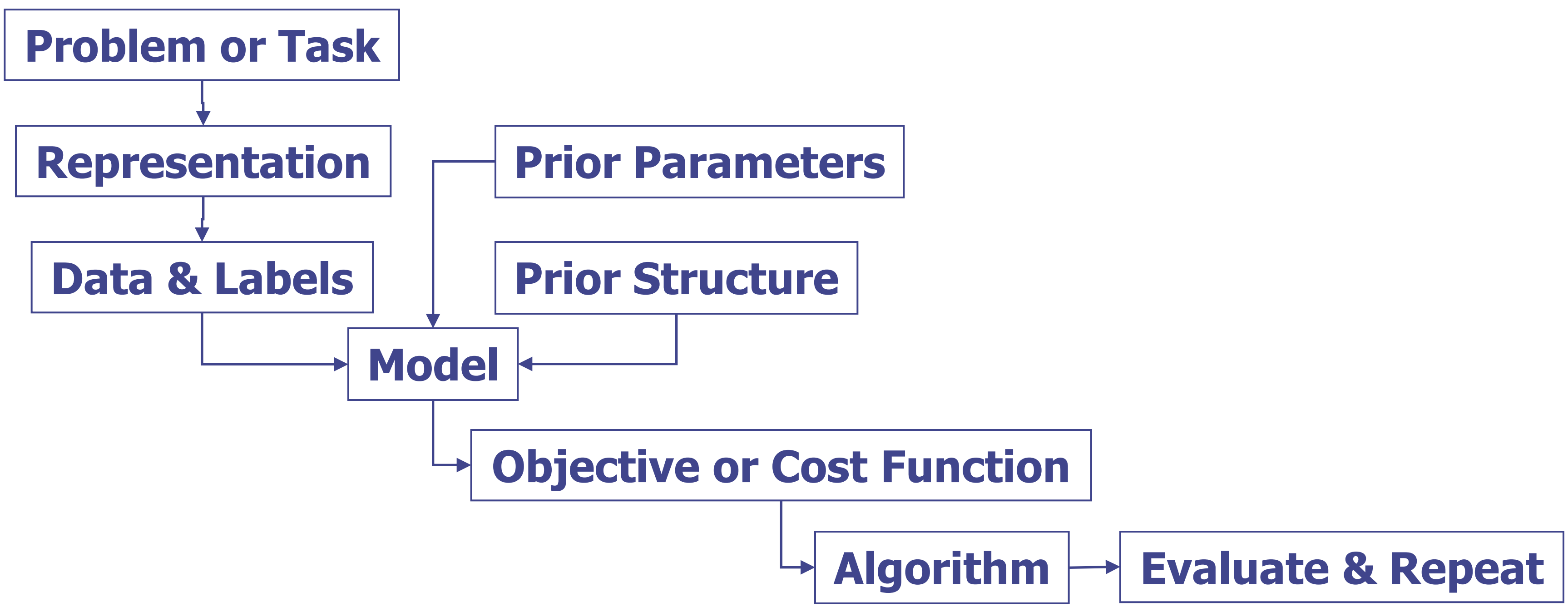



- Want to automate zipcode reading in post office
- Look at an image and say if it is a ‘1’ or ‘0’
- 8x8 pixels of gray-level (0.0=dark, 0.5=gray, 1.0=white)
- Learn from above labeled training images
- Predict labels on testing images
- Binary Classification [0,1]
- What to do?



# Ex: Setting up the Problem

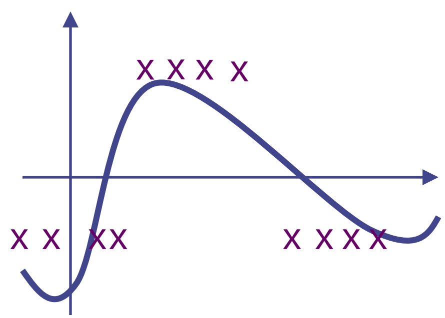


# Different Approaches

In ML, we will consider complementary approaches:

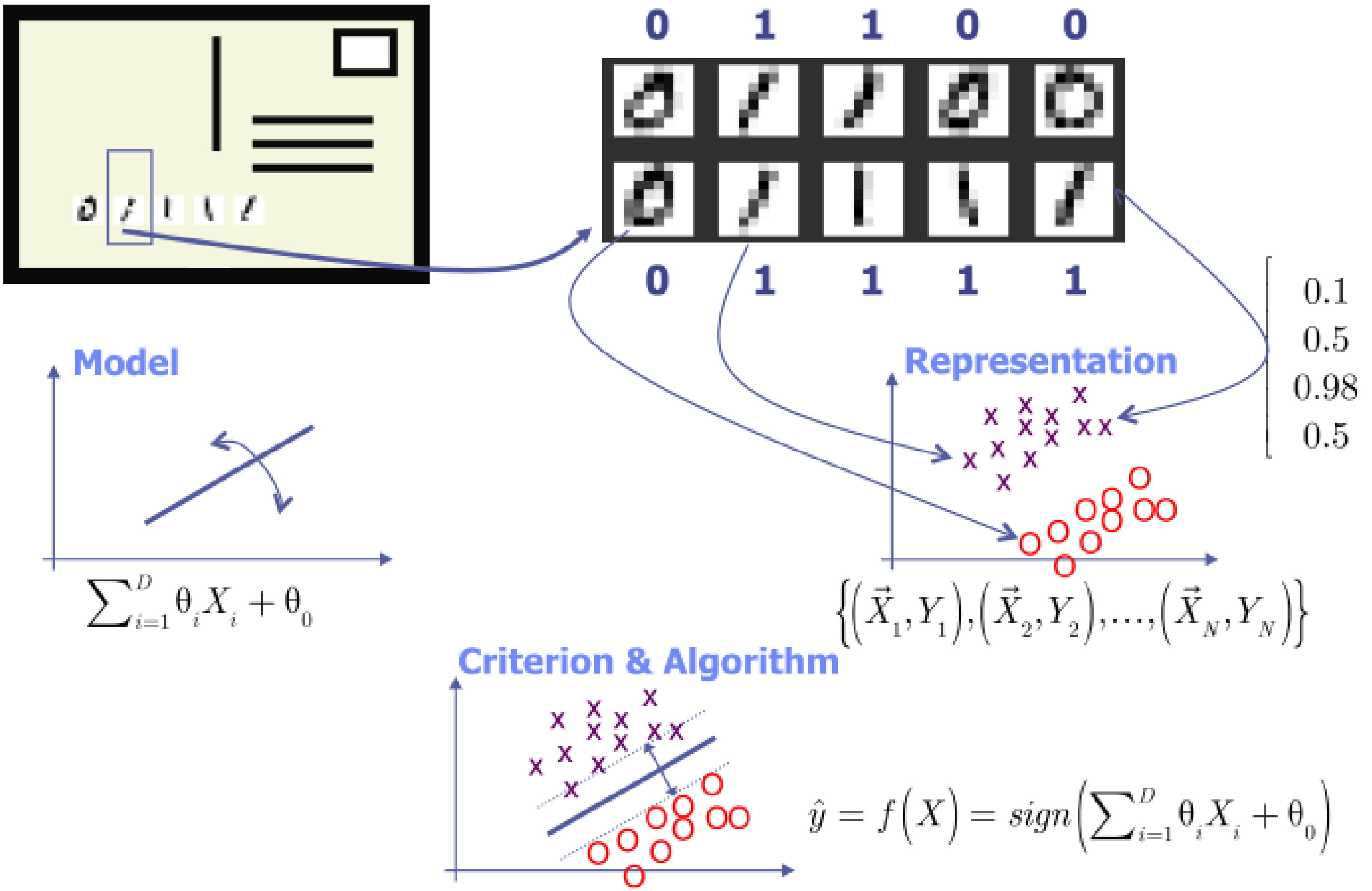
1) Deterministic:
   All variables/observables are treated as certain/exact
   Find/fit a function f(X) on an image X
   Output 0 or 1 depending on input
   Class label given by y=sign(f(X))/2 + 1/2




2) Probabilistic/Bayesian/Stochastic:
   Variables/observables are random (R.V.) and uncertain
   Probability image is a '0' digit: $p(y=0|X) = 0.43$
   Probability image is a '1' digit: $p(y=1|X) = 0.57$
   Output label with larger $p(y=0|\text{image})$ or $p(y=1|\text{image})$

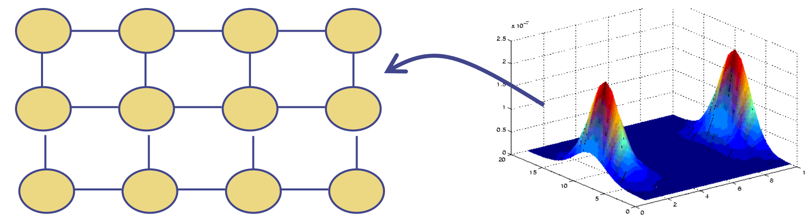
These are interconnected! Deterministic approaches can be generated from (more general) probabilistic approaches

# Ex: 1) Deterministic Approach

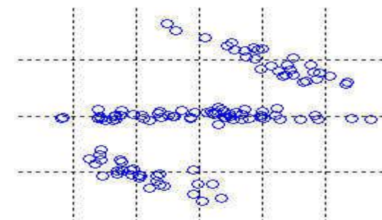


# Ex: 2) Probabilistic Approach

a) Provide Prior Model
Parameters & Structure
e.g. nearby pixels are
co-dependent

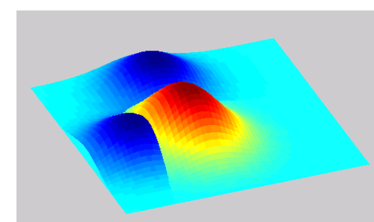


b) Obtain Data and Labels $\left\{\left(X_1, Y_1\right), \ldots, \left(X_T, Y_T\right)\right\}$



c) Learn a probability model with data
p(all system variables)

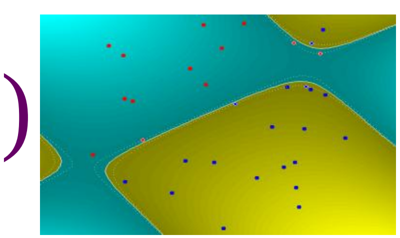
$p\left(X, Y\right)$



d) Use model for inference (classify/predict)

**Probability image is '0': p(y=0|X)**
**Probability image is '1': p(y=1|X)**
**Output: arg max$_i$ p(y=i|X)**

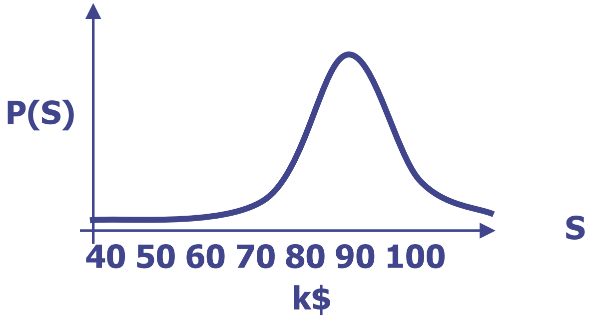
$p\left(Y \mid X\right)$



# Why Probabilistic Approach?

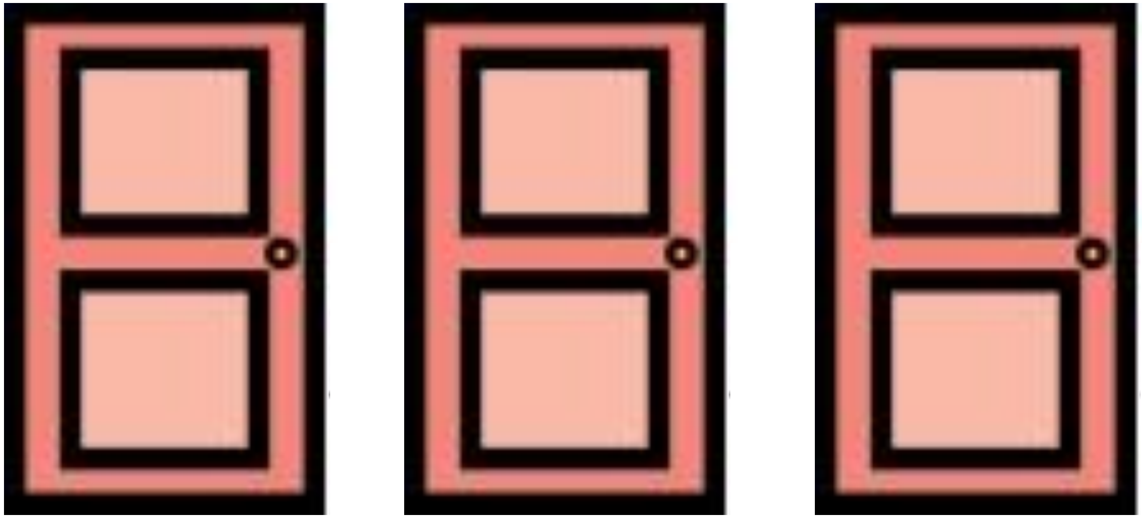
- Decision making often involves uncertainty
- Hidden variables, complexity, randomness in system
- Input data is noisy and uncertain
- Estimated model is noisy and uncertain
- Output data is uncertain (no single correct answer)

- Example: Predict your salary in the future
- Inputs: Field, Degree, University, City, IQ
- Output: $Amount
- There is uncertainty and hidden variables
- No one answer (I.e. $84K) is correct
- Answer = a distribution over salaries



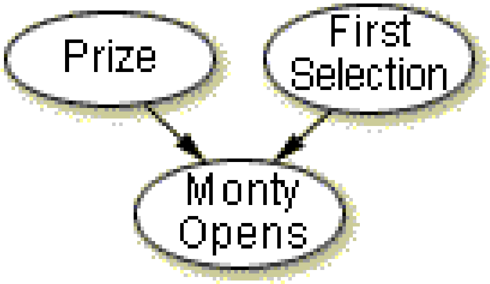
# Why Probabilistic? Monty Hall

- Behind one door is a prize (car? $1?)
- Pick a door



Door A Door B Door C

# Monty Hall Solution




**Probabilistic Graphical Model Bayesian Network**

Probabilistic Interpretation is Best

Bayesian Solution: Change your mind!

Assume we always start by picking A.

If prize behind A: Opens B/C → Change A to C/B → Lose

If prize behind B: Opens C → Change A to B → Win

If prize behind C: Opens B → Change A to C → Win

Probability of winning if change your mind = 66%
Probability of winning if stick to your guns = 33%

# Contrasting approaches

- Frequentist – Bayesian
- Discriminative – Generative
- Parametric - Nonparametric

# Ex: Is a coin fair?

A stranger tells you his coin is fair.

Let's assume tosses are iid with P(H)=p.

He tosses it 4 times, gets H H T H.

What can you say about p?