

# Lecture 2 – Machine Learning, Probability Fundamentals

COSI 134



# Machine Learning

- Why Machine Learning?
  - Difficult to define some tasks, except by example
  - Hidden relationships in lots of data (data mining)
  - Rapidly adapt/update an existing system – on-the-job adjustments
  - Volume of knowledge perhaps too large for humans to encode
- Machine Learning sub-communities in many fields
  - Statistics
  - Brain modeling
  - Psychology
  - Control Theory – robotics
  - Artificial Intelligence – NLP!
  - Evolutionary control

# Types of Machine Learning

- Supervised Learning
  - Learn a function
- Unsupervised Learning
  - Cluster data points
- Reinforcement Learning
  - Learn a policy
- Explanation-based Learning
  - Speed-up learning
- Semi-supervised Learning
  - Learn a function; also consider unlabeled data

Machine Learning Problems:

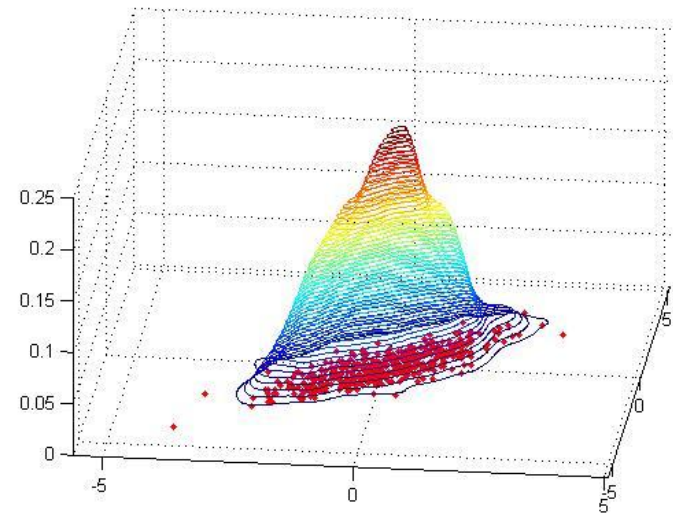
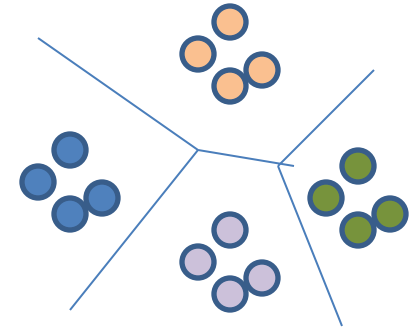
-Task

-Data

-Measure of Improvement

# Unsupervised Learning

- Cluster data points:  $D = \{x^{(1)}, \dots, x^{(n)}\}$ 
  - Data points could be documents in a vector space
  - Cluster documents together
- Density estimation (statistics)
- Dimensionality Reduction
  - Principal Component Analysis
  - Latent Semantic Analysis
- Generative Probabilistic Models
  - Bayesian Networks
  - Non-parametric Bayesian models



# Unsupervised Learning for NLP

- Advantages
  - No need to annotate data!
  - Rapidly adapt to new data
- Difficulties
  - Hard to bias/constrain learning
  - Getting good results is a “dark art”
- Generally unsupervised learning does not work as well as supervised learning with even small amounts of data
- But, interesting progress made in certain areas
  - Co-reference, information extraction, lexical semantics
- Semi-supervised learning
  - Improved adaptation
  - Require smaller amounts of fully labeled data

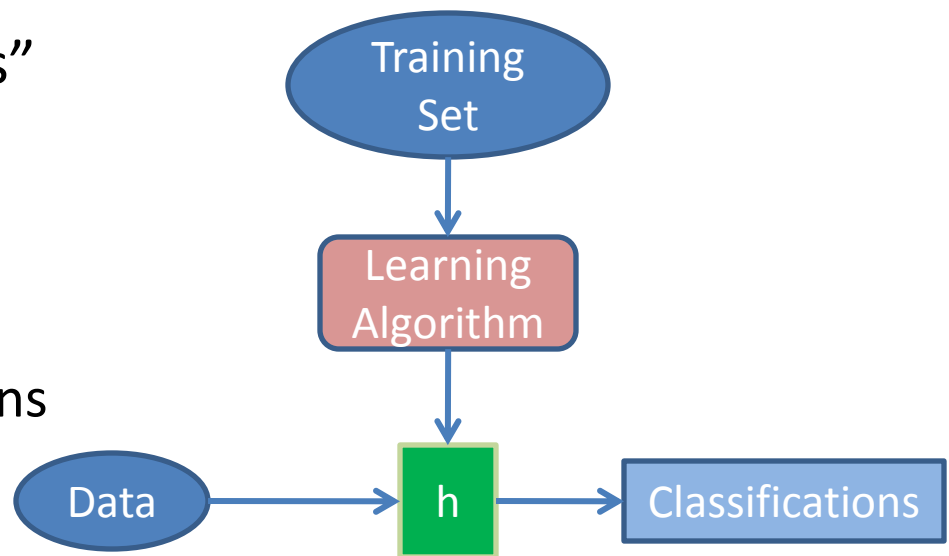
# Reinforcement Learning

- Set in a context of an AGENT
  - Agent is situated in a world
  - Agent has a set of actions available
  - State of the world changes of time
    - Through agent actions or environment
  - Agent may or may not have perfect knowledge of the state
- Aim of Reinforcement Learning is to learn a *policy*
  - A function that maps states (or observations) to actions
  - That maximizes reward, minimizes punishments
  - E.g. win or lose a game
- Credit-assignment problem



# Supervised Learning

- Let  $X$  be the space of inputs, usually  $X = \mathbf{R}^n$
- Let  $Y$  be the space of outputs,  $Y = \{-1,1\}$ ,  $Y = \mathbf{Z}^n$ ,  $Y = \mathbf{R}$
- $Y = \{-1,1\}$ , binary classification,  $Y = \mathbf{R}$  is regression
- Learn a function  $h: X \rightarrow Y$
- Referred to as the “hypothesis”
- Also:
  - Model – the parameters
  - Decoder – algorithm that uses model to arrive at classifications



# Machine Learning Representations

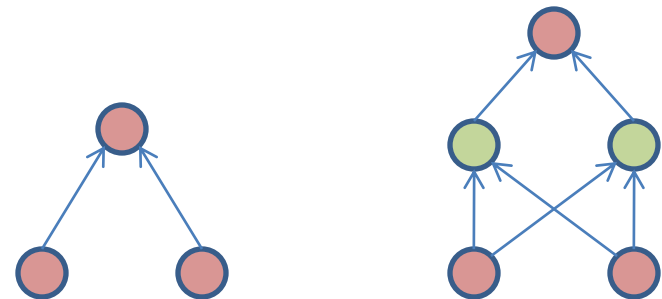
- Features

- Categorical features: Hair-color:{Blond, Black, Red,...}
- Binary features
  - Hair-is-blond:{true,false}
- Real-valued features
  - Wind-speed: 32.5

- Model structure

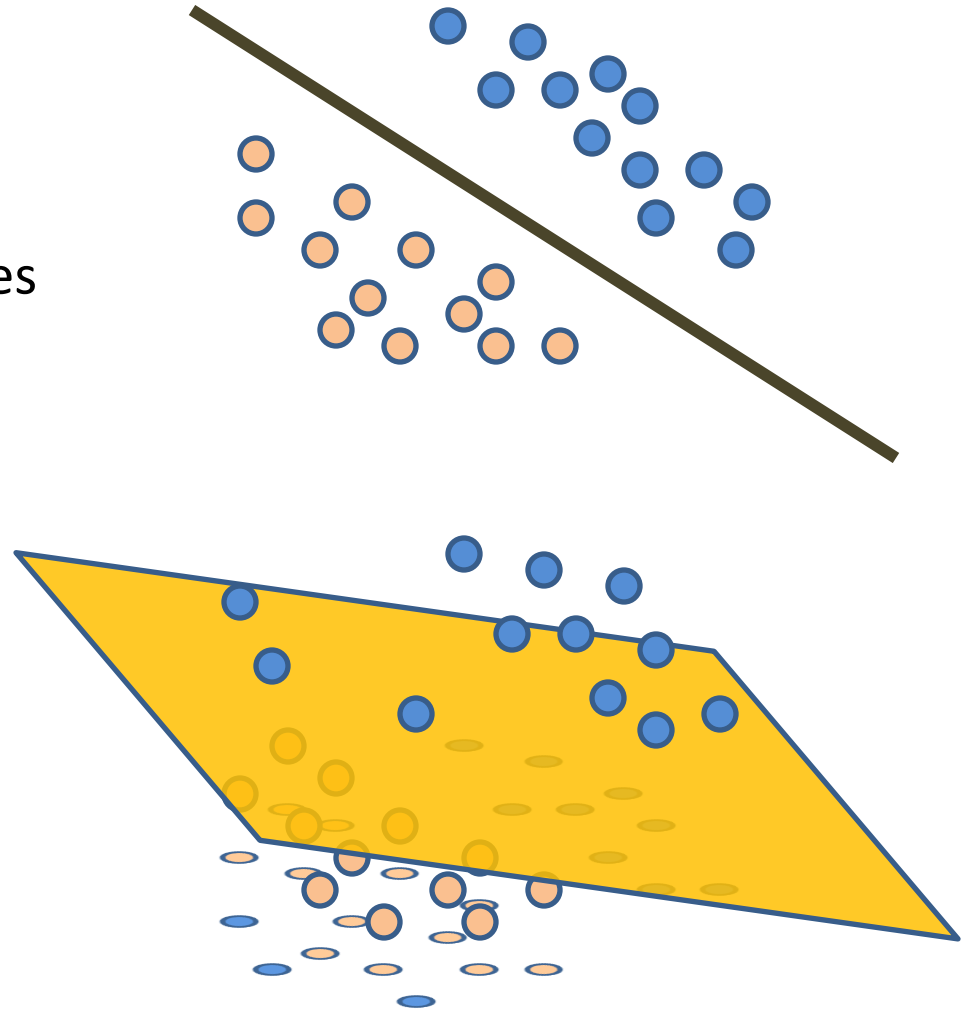
- Interactions among features
  - Dependencies
  - Conjunctions
- Hidden nodes, latent variables
- Relational (first-order) models vs propositional models

- Structure learning vs parameter estimation



# Supervised Learning Algorithms

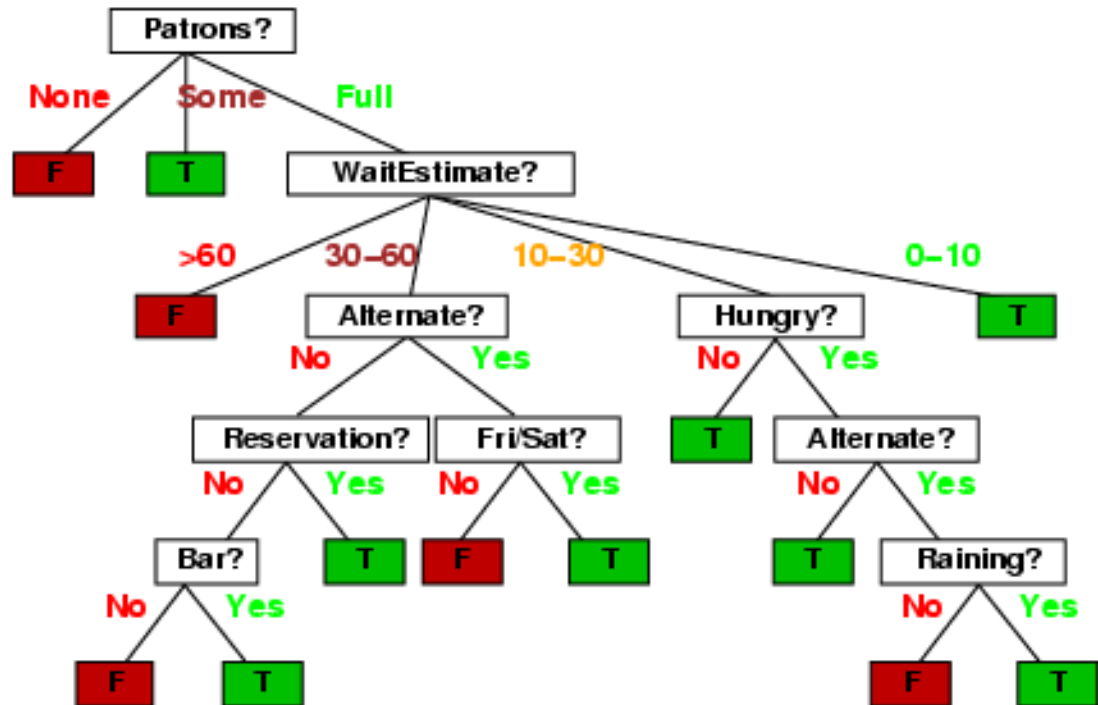
- Binary classification
  - Find hyper-plane that separates the data
- Perceptron algorithm
- Margin-based algorithms
  - Support Vector Machines
  - Kernels
- Neural Networks
- Curse/boon of dimensionality



# Categorical Supervised Learning

- Decision Trees

- Select conjunctions of attributes/features that lead to particular classifications
- Search for the right representation



# Probabilistic Classifiers

- Formulate classification problem as probabilistic inference

- $p(y = \text{"yes"} | \mathbf{x})$

- Advantages

- Classifiers produce not just output classification, but distribution over possible classifications

$$p(y = \text{"yes"} | \mathbf{x}) = 0.8$$

$$p(y = \text{"no"} | \mathbf{x}) = 0.2$$

- Solid mathematical underpinnings from statistics
- Frameworks for unifying supervised/unsupervised learning
- Graphical models for structured classification
- Uncertainty in the model is captured well
  - Probabilities usually more meaningful as “confidences” than with margin-based methods

# Learning with Dependencies

- In simplest case, each *instance* to classify is assumed to be independent of all other instance
- Instances are assumed to be identically distributed
  - They are “drawn” from the same distribution in the same manner
- These assumptions are violated in many real-world problems
- In NLP, we’re often interested in classifying elements **in sequence**
  - E.g. Part-of-speech tagging, named entity identification
- Sometimes more complicated structure
  - Parsing
  - Co-reference

# Quiz #1

- When: Sept. 28, 2010 - In class
- Topics (Tentative)
  - Mutual information, Entropy
  - Conditional Independence
  - Probability, linear algebra basics
  - VERY Basic notions of morphology, syntax, semantics
  - Naïve Bayes
  - General ideas of machine learning
    - Goal of supervised learning, unsupervised learning

# Readings, Papers

- Chapters 1-3 in Manning and Schütze
- Koller, Friedman Handout [Tuesday]
  - Naïve Bayes
- CoNLL 2003 Challenge on Named Entity
  - Pick any paper (and separately read the overview of task)
  - Discuss, hand in summary on 9/16
  - Summary includes:
    - ½ page (typed!) well-written description of the paper
    - At least 2-3 questions or things you didn't understand
    - OR – a coherent critique of the paper

# More Announcements

- Python tutorial – tentatively 9/7 (3:00-5:00)
  - Some focus on NLTK
- Weka tutorial
  - Later homeworks will use (or have an option for) Weka
  - Schedule sometime towards end of Sept.
    - 9/27 to 9/29?
- Office Hours
  - Ben: 4-5pm (Volen 256) on Tues/Thurs
  - Chen: 3-5pm (Volen 110) on Thurs
  - Yaqin: 3-5pm (Volen 110) on Tues

# **MATHEMATICAL FOUNDATIONS**

# Probability Theory

- How likely it is that something will happen
- Sample space  $\Omega$  is listing of all possible outcome of an experiment
- Event A is a subset of  $\Omega$
- Probability function (or distribution)

$$P : \Omega \rightarrow [0, 1]$$

- Prior probability: the probability before we consider any additional knowledge

$$P(A)$$

# Conditional probability

- Sometimes we have partial knowledge about the outcome of an experiment
- Conditional (or Posterior) Probability
- Suppose we know that event B is true
- The probability that A is true given the knowledge about B is expressed by

$$P(A | B)$$

# Conditional Probability

- Joint probability of A and B.
- 2-dimensional table with a value in every cell giving the probability of that specific state occurring

$$\begin{aligned}P(A,B) &= P(A|B)P(B) \\ &= P(B|A)P(A)\end{aligned}$$

$$P(A,B,C,D\dots) = P(A)P(B|A)P(C|A,B)P(D|A,B,C\dots)$$

$$P(A|B,C) = ?$$

# Reading Probabilities

likesCourse	Background	doesWell	
Y	Y	Y	0.35
Y	Y	N	0.05
Y	N	Y	0.35
Y	N	N	0.05
N	Y	Y	0.05
N	Y	N	0.05
N	N	Y	0.05
N	N	N	0.05

# (Conditional) Independence

- Two events  $A$ ,  $B$  are independent of each other if
- $P(A) = P(A | B)$
- Two events  $A$  and  $B$  are conditionally independent of each other given  $C$  if
- $P(A | C) = P(A | B, C)$

# Bayes' Theorem

- Bayes' Theorem lets us swap the order of dependence between events

- We saw that

$$P(A | B) = \frac{P(A, B)}{P(B)}$$

- Bayes' Theorem:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

# Example

- S:stiff neck, M: meningitis
- $P(S | M) = 0.5$ ,  $P(M) = 1/50,000$   $P(S) = 1/20$
- I have stiff neck, should I worry?

$$\begin{aligned} P(M | S) &= \frac{P(S | M)P(M)}{P(S)} \\ &= \frac{0.5 \times 1/50,000}{1/20} = 0.0002 \end{aligned}$$

# Random Variables

- Random variable (RV)  $X$  allow us to talk about the probabilities of numerical values that are related to the event space

$$X : \Omega \rightarrow \mathfrak{R}$$

$$X : \Omega \rightarrow S \subset \mathfrak{R}$$

$$X : \Omega \rightarrow \{0,1\}$$

- Examples

- Let sample space be the events from pairs of dice rolled

$$\Omega' = \{(1,1), (1,2), (2,1), \dots, (6,6)\}$$

- Let  $X$  be the sum of the two dice  $X : \Omega' \rightarrow \{2, \dots, 12\}$

- A RV has a probability mass function (discrete) or a density function (continuous)

$$P(X = 2) = \frac{1}{36}, P(X = 3) = \frac{1}{18}, \dots, P(X = 7) = \frac{1}{6}$$

# Expectation

$$p(x) = p(X = x) = p(A_x) \quad \sum_x p(x) = 1$$
$$A_x = \{\omega \in \Omega : X(\omega) = x\} \quad 0 \leq p(x) \leq 1$$

- **The Expectation is the mean or average of a RV**

$$E(x) = \sum_x xp(x) = \mu$$

- **Expectation of sum of RVs is sum of expectations**

$$E(x + y) = E(x) + E(y)$$

- **Expectation of product is product of expectations ONLY when independent**

$$E(xy) = E(x)E(y)$$

# Example of Expectations

- Let  $Y$  be the RV which is the value of rolling one die

$$E(Y) = \sum_{y=1}^6 yP(Y = y) = \frac{1}{6} \sum_{y=1}^6 y = \frac{21}{6} = 3\frac{1}{2}$$

- Let  $X$  be the RV for the sum of two dice
- What is the expected value of  $X$

$$E(X) = E(Y) + E(Y) = 7$$

# Variance

- The variance of a RV is a measure of whether the values of the RV tend to be consistent over trials or to vary a lot

$$\begin{aligned}\text{Var}(X) &= E((X - E(X))^2) \\ &= E(X^2) - E^2(X) = \sigma^2\end{aligned}$$

- $\sigma$  is the standard deviation

# Estimation of Probability Distributions

- Frequentist statistics
  - Avoid any prior beliefs, bias and subjectivity; more purely empiricist
  - Statistics is separate from probability
    - Probability is about reasoning and making inferences given a model
    - Statistics is about inferring a model and/or parameters from data
- Bayesian statistics
  - Treat statistical inference as probabilistic inference
  - Big point: view models and model parameters as random variables

$$P(\theta | \mathbf{D}) = \frac{P(\mathbf{D} | \theta)P(\theta)}{P(\mathbf{D})}$$

# Bayesian vs Frequentist Learning

- Calculate the probability of EACH hypothesis (e.g. each possible set of parameters) and make predictions
- Frequentist learning involves finding the single most likely or best hypothesis

- Bayesian prediction: 
$$P_D(Y | X) = \int_{\theta'} P(Y | X, \theta') P(\theta' | D)$$

- Frequentist prediction: 
$$P_D(Y | X) = P(Y | X, \theta_{Best})$$

# Likelihood

- Likelihood – How “likely” is the data given a specific model and set of parameters  $\theta$ 
  - (We will usually implicitly assume a given model distribution and refer just to the model parameters)
- The probability:  $P(\mathbf{D} | \theta)$
- Both Bayesian and frequentist approaches have to calculate likelihood values for different  $\theta$  (something in common)
- Frequentist approach involves finding the “point estimate” (i.e., single set of parameters or hypothesis) that maximizes the likelihood
  - But Bayesian ideas can influence this

# MAP and Maximum Likelihood

- Bayesian learning/prediction is often intractable
  - Requires complex summation/integration
- Rather than a distribution of models, take the mean or mode
  - Mean:  $\theta_{\text{BayesPoint}} = \int \theta P(\theta | \mathbf{X}) = \alpha \int \theta P(\mathbf{X} | \theta)P(\theta) d\theta$
  - Mode:  
$$\theta_{\text{MAP}} = \operatorname{argmax}_{\theta} P(\theta | \mathbf{X}) = \operatorname{argmax}_{\theta} P(\mathbf{X} | \theta)P(\theta)$$
  - This is called maximum a posteriori (MAP) estimation
- Maximum likelihood learning is MAP learning with a uniform prior

# Estimation of P

- Frequentist statistics
  - Parametric methods
    - Standard distributions:
      - Binomial distribution (discrete)
      - Normal (Gaussian) distribution (continuous)
        - Maximum likelihood
    - Non-parametric methods
  - Bayesian statistics
    - Bayesian statistics measures degrees of belief
    - Degrees are calculated by starting with *prior beliefs* and updating them in face of the evidence, using Bayes' theorem

# Frequentist vs. Bayesian

- Bayesian 
$$M^* = \operatorname{argmax}_M P(M) = \int_{\theta} P(D | M, \theta) P(\theta | M) d\theta$$

- Frequentist

$$\theta^* = \operatorname{argmax}_{\theta} P(D | M, \theta) \qquad M^* = \operatorname{argmax}_M P\left(D | M, \theta^*(M)\right)$$

$P(D | M, \theta)$  is the likelihood

$P(\theta | M)$  is the parameter prior

$P(M)$  is the model prior

# Bayesian Updating

- How to update  $P(M)$ ?
- We start with a priori probability distribution  $P(M)$ , and when a new datum comes in, we can update our beliefs by calculating the posterior probability  $P(M | D)$ .
- This then becomes the new prior and the process repeats on each new datum

# Bayesian Model Decision Theory

- Suppose we have 2 models  $M_1$  and  $M_2$ ; we want to evaluate which model better explains some new data.

$$\frac{P(M_1 | D)}{P(M_2 | D)} = \frac{P(D | M_1)P(M_1)}{P(D | M_2)P(M_2)}$$

$$\text{if } \frac{P(M_1 | D)}{P(M_2 | D)} \geq 1 \quad \text{i.e. } P(M_1 | D) \geq P(M_2 | D)$$

- $M_1$  is the most likely model, otherwise  $M_2$
- Bayesian **optimal** learning involves finding the best  $M$  (and parameters) from some defined space of models
  - E.g. Gaussian mixtures with # of mixture components between 1,10

# Parametric Methods

- Assume that some phenomenon (in language) is acceptably modeled by one of the well-known family of distributions (such binomial, normal, multinomial)
- We have an explicit probabilistic model of the process by which the data was generated, and determining a particular probability distribution within the family requires only the specification of a “few” parameters (less training data)

# Non-Parametric Methods

- No assumption about the underlying distribution of the data
- For ex, simply estimate  $P$  empirically by counting a large number of random events is a distribution-free method
- Less prior information, more training data needed

# Binomial Distribution (Parametric)

- Series of trials with only two outcomes, each trial being independent from all the others
- Number  $r$  of successes out of  $n$  trials given that the probability of success in any trial is  $p$ :
- Probability mass function: 
$$b(r; n, p) = \binom{n}{r} p^r (1-p)^{n-r}$$

# Normal (Gaussian) Distribution (Parametric)

- Continuous
- Two parameters: mean  $\mu$  and standard deviation  $\sigma$

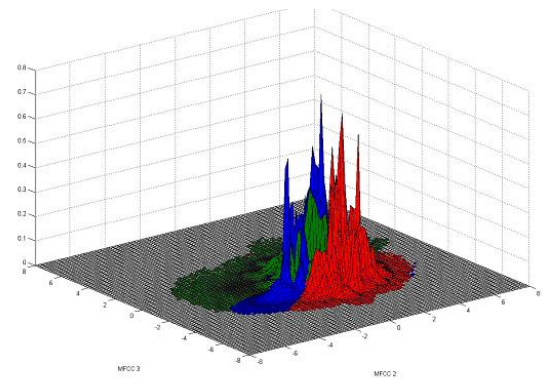
- Used in clustering

- Density function: 
$$n(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

# Gaussian Mixtures

- Let  $g_1 = \mu = 1, \sigma = 0.2$
- Let  $g_2 = \mu = 4, \sigma = 0.02$
- Let  $g_3 = \mu = 0, \sigma = 20$

$$n(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



- Introduce a set of  $k$  mixture weights,  $n_1, n_2, n_3$ 
  - A multinomial distribution
- Generate a point by selecting which Gaussian from the mixture and then generate via the Gaussian density

# Non-Parametric Example

- Kernel Density Estimation
- Let each kernel be a “little” Gaussian distribution around one or more points
- The number of kernels will be a function of the number of points
- There isn't a fixed set of parameters up front
  
- Examples
  - Support Vector Machines
  - K-Nearest Neighbor
  - Gaussian Processes

# (Non-)Parametric Methods

- Class will focus most on **parametric** methods
- Advantages
  - Known properties of distributions used
  - Fixed set of parameters (or parameter types)
- Some advantages of **non-parametric** methods
  - Can better fit the data when it doesn't match a known distribution
  - More flexibility with methods and loss functions
- Big disadvantage with non-parametric:
  - Complexity grows with size of the dataset (training set)

# Linear Algebra and Calculus Essentials

- Machine learning makes heavy use of linear algebra
- Statistics & Probability make use of discrete math, combinatorics
- Calculus required for in depth analysis of some methods
  - Most calculus for most learning algorithms is pretty basic
- Some methods based on mathematical programming
  - Linear programming
  - Quadratic programming
  - Integer programming
- *Getting the gist of the math, and being able to “translate” equations into pseudocode allows for a solid understanding of many statistical NLP methods*

# Basic Concepts, Notation

- Linear algebra provides compact ways to operate with systems of equations

$$3x_1 + 5x_2 = 10$$

$$-2x_1 + 3x_2 = 12$$

- Represent in matrix notation as  $Ax = b$

$$A = \begin{bmatrix} 3 & 5 \\ -2 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 10 \\ 12 \end{bmatrix}$$

- Matrix with  $m$  rows and  $n$  columns:  $A \in \mathbf{R}^{m \times n}$

- A vector with  $n$  entries:  $x \in \mathbf{R}^n$

- A matrix with  $n$  rows and 1 column

- A row vector:  $x^T$

- $i$ th element of a vector  $x$  is denoted  $x_i$

- Additional notations for vectors:

$$\vec{x} \quad \mathbf{x}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

# Matrices

- Use  $a_{ij}$  to denote entry of  $A$  in the  $i$ th row and  $j$ th column
- Matrix multiplication
  - Product of two matrices

$$A \in \mathbf{R}^{m \times n} \text{ and } B \in \mathbf{R}^{n \times p}$$

$$C = AB \in \mathbf{R}^{m \times p}$$

$$\text{where } C_{ij} = \sum_{k=1}^n A_{ik} B_{kj}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & \dots & a_{mn} \end{bmatrix}$$

- Example:

$$\begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 7 \\ 1 & 2 & 4 \end{bmatrix}$$

# Vectors and Matrices

- Inner product, dot product:

$$x^T y = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} = \sum_{i=1}^n x_i y_i$$

- Outer product

$$xy^T = \begin{bmatrix} x_1 \\ \dots \\ x_m \end{bmatrix} \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \dots & x_1 y_n \\ x_2 y_1 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ x_m y_1 & \dots & \dots & x_m y_n \end{bmatrix}$$

- Properties of matrices

- Associative
- Distributive
- Not commutative

# More Matrix, Vector Properties

- Transpose

- Result of “flipping” the rows and columns of a matrix

Given  $A \in \mathbf{R}^{m \times n}$ ,  $A^T \in \mathbf{R}^{n \times m}$  is the  $n \times m$  matrix with entries:  $(A^T)_{ij} = A_{ji}$

- Symmetric matrix  $A \in \mathbf{R}^{n \times n}$ ,  $A = A^T$

- Vector Norm

$$l_2 \longrightarrow \|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2} \quad l_1 \longrightarrow \|x\|_1 = \sum_{i=1}^n |x_i|$$