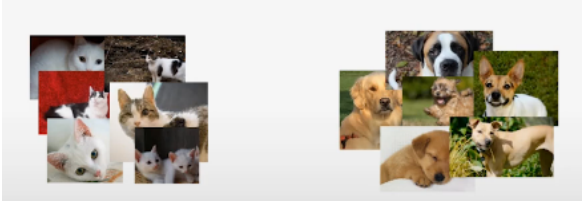
Introduction to Machine Learning

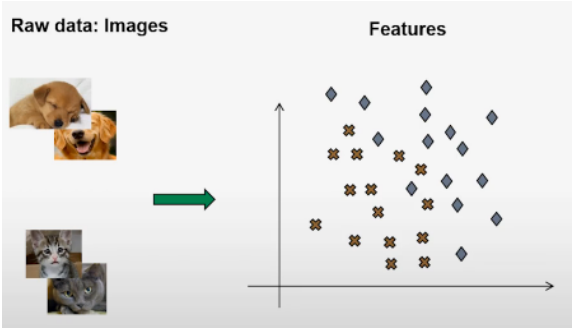
What is Machine Learning

- Many different definitions for “machine learning”
    - All involve learning by a machine (computer)
- Definition of learning in a typical dictionary: “the acquisition of knowledge or skills through experience, study, or by being taught”
    - Can machines be enabled to learn, without being explicitly programmed?
- Learning and Adaptation

An Illustrative Example

- Given some example pictures, how a computer can learning to differentiate dogs from cats?
    - 

Data Representation - Feature Extraction

- 

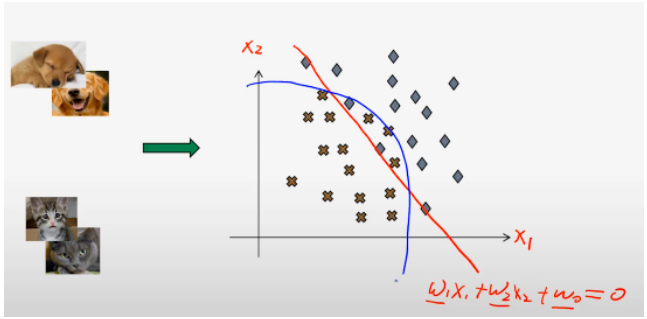
Different Types of Data Representations

- Numerical; Categorical; Ordinal
    - Univariate or multivariate
        - All could be represented by numbers
- Graphical representations in terms of notes and edges
    - E.g., Social Network Analysis

Preprocessing for Feature Extraction

- Segmentation
- Filtering
- Various transformations
    - All intended for facilitating feature extraction
- Good features should be *invariant* in some sense

Mathematical Models for Classification

- 

Importance of Statistical Modeling

- Why we often reply on statistical methods in machine learning?
- Data is noisy (measurement noise) → Features are often represented ransom variables/vectors
- Inaccuracy of the assumed model
- Inherent ambiguity of many real-world problems

Basic Machine Learning Paradigms

- Supervised Learning: the training samples have labels
- Unsupervised learning: the training set is not labeled
- Reinforcement learning: learning to take actions to maximize some notion of *reward*

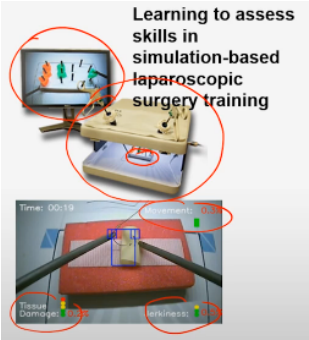
Knowledge Check

- Which of the following are considered supervised learning techniques?
    - Regression
        - Regression has both data and its corresponding label, so it is a supervised technique
    - Classification has both data and its corresponding label, so it is a supervised technique
- Which of the following are unsupervised learning problems?
    - Density Estimation
        - Density estimation task does not have any label information, so it is an unsupervised technique
    - Clustering
        - Clustering task does not have label information, so it is an unsupervised technique
- Which of the following is a numerical data type representation?
    - 2-D feature vectors
- The weather is labelled as cloudy, sunny, and rainy. Which data type would be most appropriate to represent the label?
    - Categorical

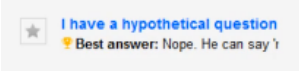
Introduction to Machine Learning: Applications

A few examples of machine learning (⅓):

- Learning to assess skills in simulation-based laparoscopic surgery training

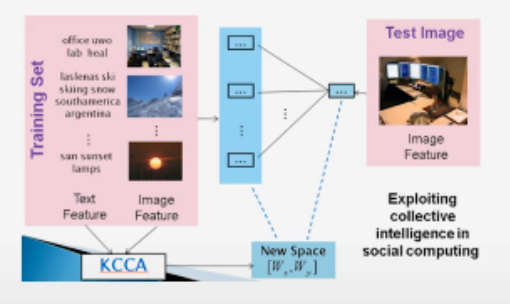


- Learning to predict best answers in a community Q&A

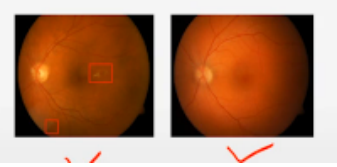


A few examples of machine learning (⅔):

- Tag prediction/recommendation

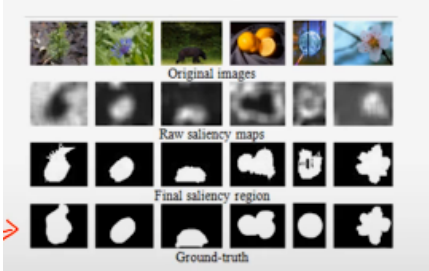


- Diabetic retinopathy detection



A few examples of machine learning (3/3):

- Visual saliency detection



- Computational visual aesthetics

Broad Applications of Machine Learning

- Computer vision
- Speech recognition; natural language processing
- Medical informatics
- Robotics
- Computational biology
- Information Technology
- Finance

Information Technology

- Spam detection
- Web image search
- Recommendation
- Information filtering
- Community detection
- Adaptive advertisement
- Sentiment analysis

Finance

- Credit risk assessment
- Fraud detection
- Stock market prediction
- Algorithmic trading
- Return forecasting

Knowledge Check

- If you want to solve a maze, which of the following learning method should you use?
    - Reinforcement learning

- Machine learning is related to which of the following fields?
    - Data Mining
    - Statistics
    - Artificial Intelligence
- Which of the following statements describes machine learning?
    - Building systems that can learn from data
    - The science of making computers better at some task with experience, given some performance measure
    - The science of getting computers to act without being explicitly programmed
- Which of the following problems are examples of unsupervised learning?
    - Grouping on-line users according to their similarity to each other
    - Recommending movies by using other users' view histories

Module 1 Quiz Questions

1. Players on a kids soccer team need to order new water bottles. There are different color choices. Which data type would be used to represent the colors of the water bottles?
    a. Categorical
2. Which of the following is a data type that is most useful for representing relationships among the users of a social network?
    a. Graphs with edges and nodes

Review of Mathematical Foundations Part 1

Basic Notations from Calculus (⅓)

- Derivative of f(x) with respect to x
- Partial derivative of a function f(x, y,...) with respect to x
    - Note: the function may be scalar-valued or vector-valued

Basic Notations from Calculus (⅔)

- R^d: d-dimensional Euclidean space
- Gradient operator in R^d: ∇

Basic Notations from Calculus (3/3)

- The integral of f(x) between a and b
- The argmin or argmax notation

Basic Notations from Set Theory (½)

- a set S is a collection of objects
    - $\emptyset$: the empty set (a special set that contains no object)
- Some basic relations and operations
    - $x \in A$: an object x is a member of a set A

- $A \subseteq B$: set A is a subset of $B \leftrightarrow x \in A \rightarrow x \in B$
- $B \subset B$: set B is a proper subset of C

Basic Notations from Set Theory (2/2)

- Some basic relations and operations
  - $A \cup B$: the union of A and B
  - $A \cap B$: the intersection of A and B (AB in shorthand)
  - Ac or $\overline{A}$: the complement of A
  - A and B are disjoint if $A \cap B = \emptyset$

Review of Mathematical Foundations Part 2

Linear Algebra: Basic Notations (¼)

- a d-dimensional column vector $\mathbf{x}$ and its transpose $\mathbf{x}^t$
- n by d matrix M and its d by n transpose M^t

Linear Algebra: Basic Notations (2/4)

- a square matrix M is symmetric if $M^t=M$
- multiplying a vector by a matrix: $M\mathbf{x}=y$
- multiplying two matrices M_1 **and** M_2

Linear Algebra: Basic Notations (3/4)

- the identity matrix $\mathbf{I}$ of d by d
- inner product of two vectors $\mathbf{x}^t\mathbf{y}$
- outer product of two vectors $\mathbf{x}\mathbf{y}^t$

Linear Algebra: Basic Notations (4/4)

- the length or Euclidean **norm** of a vector $\mathbf{x}$, denoted $||\mathbf{x}||$
- normalized vector, $||\mathbf{x}||=1$

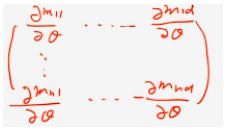
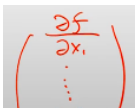
Matrix: Additional Definitions (½)

- Determinant of a matrix M: denoted |M| or det(M)
  - Look at size 2x2
  - What about size 3x3 and above?
- Trace of a matrix

Matrix: Additional Definitions (2/2)

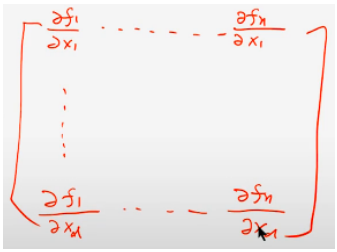
- matrix inversion M^{-1}
- eigenvectors and eigenvalues of M

Derivatives Involving Matrices (⅓)

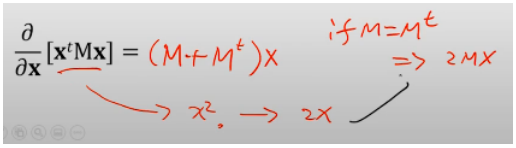
- If the entries of matrix M depend on a scalar parameter θ, we have $\frac{\partial M}{\partial \theta} =$ 
- Derivative of a scalar-valued function $f(\mathbf{x})$ of d variables x_i, i=1, ..., d, and $\mathbf{x}=(x_1,\ldots x_d)^t$, or the gradient w.r.t. $\mathbf{x}$ is 

$$\nabla f(x) = \frac{\partial f(x)}{\partial x} =$$

Derivatives Involved Matrices (⅔)

- If **f(x)** if n-dimensional vector-valued function of d variables x_i, i=1, ..., d, and **x**=$(x_i, ..., x_d)^t$, we have the derivative as*
- $\frac{\partial f(x)}{\partial x} =$ 

Derivatives Involving Matrices (3/3)

- Some useful results
  - $\frac{\partial}{\partial x}[Mx] = M^t$
  - $\frac{\partial}{\partial x}[y^t x] = y$
  - $\frac{\partial}{\partial x}[x^t Mx] = (M + M^t)x$
  - 

Knowledge Check

- Given $f(x) = 3x^2 - 2x^2 + 4x - 5$, what is the derivative of $f(x)$ with respect to x?
  - $9x^2 - 4x + 4$
- What does $A \cup B$ mean?
  - The union of A and B
- What is X^t if matrix $X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$?
  - $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$
- Which of the following matrices is a symmetric matrix? (Select all that apply)

- $\begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix}$
- $\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$
- $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

- Given matrix $X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, what is the value of *det(X)?*
  - -2

Review Probability Theory

Probability Space (½)

- A probability space is a triplet (Ω, B, P) that is used to model a process of an experiment with random outcomes.
  - The **sample space** Ω is the set of all possible outcomes of an experiment
    - Consider two different experiments
      (1) Tossing a coin
      (2) Tossing a die
- **Probability Space (2/2)**
  - B: a sigma algebra (or Borel field), or informally, a collection of subsets Ω, subject to some constraints (like containing the empty set, being closed under compliments and countable union)
  - P: a measure called **probability** defined on B, that satisfies
    - $P(A) \geq 0$ for all $A \in B$
    - $P(\Omega) = 1$
    - If $A_1, A_2, \ldots \in B$ are pairwise disjoint then $P(\cup A_i) = \Sigma P(A_i)$
      - (i.e.., $A_j A_k = \emptyset$, $\forall\ j \neq k$

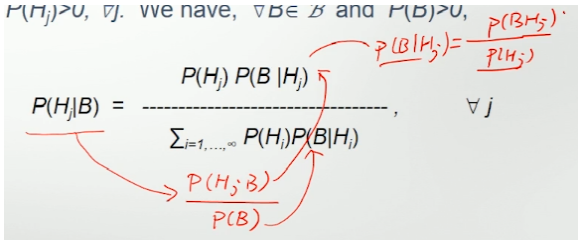
Conditional Probability

- Let (Ω, B, P) be a probability space and let $H \in B$ with $P(H)0$. For any $B \in B$, we define $P(B|H)=P(BH)/P(H)$ and call $P(B|H)$ the conditional probability of B given H

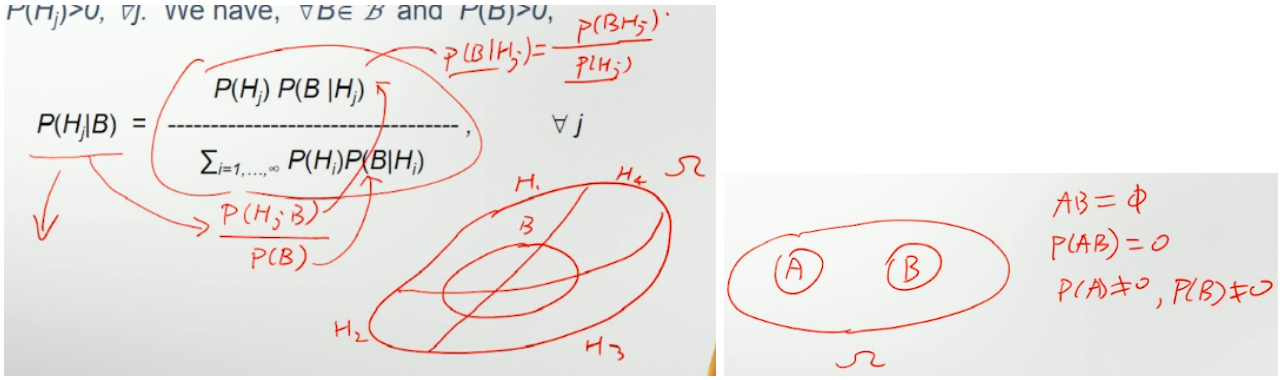
The Total Probability Rule

- Let (Ω, B, P) be a probability space, and let $\{H_j\}$ be pairwise disjoint events in B(i.e., $H_j H_k = \emptyset$, $\forall_j$, then $P(B) = \Sigma_{j=1,\ldots\infty} P(H_j)P(B|H_j)$
- Such $\{H_j\}$ is called a partition of Ω

The Bayes Rule

- Let (Ω, B, P) be a probability space, and let $\{H_j\}$ be pairwise disjoint events in B with $\cup_{j=1...\infty} H_j = \Omega$, and $P(H_j)>0$, $\forall j$. We have, $\forall B \in B$ and $P(B)>0$
- 

Independence of Events

- Let (Ω, B, P) be a probability space, $\forall A, B \in B$, we say A and B are independent if $P(AB) = P(A)P(B)$
- 

Knowledge Check

- The table below shows the purchase history of 10 customers from a set of zip codes that bought organic tea or organic coffee. Using this data, what is the estimated prior probability that a customer would buy organic tea?

CustomerID	Zipcode	Bought Organic Coffee	Bought Organic Tea
1	44005	yes	yes
2	44001	no	no
3	44001	yes	yes
4	44005	no	no
5	44003	yes	no
6	44005	no	yes
7	44005	no	no
8	44001	no	no
9	44005	yes	yes
10	44003	yes	yes

  - 0.5
    - In the last column "Bought Organic Tea", Yes appears 5 times, so the prior probability is 5/10 = 0.5.

- Given the same table as the previous question, what is the probability of a customer being from zip code 44005, given that the person bought organic tea?

CustomerID	Zipcode	Bought Organic Coffee	Bought Organic Tea
1	44005	yes	yes
2	44001	no	no
3	44001	yes	yes
4	44005	no	no
5	44003	yes	no
6	44005	no	yes
7	44005	no	no
8	44001	no	no
9	44005	yes	yes
10	44003	yes	yes

  - 0.6
    - (P(zip = 44005|T) = P(T,44005)/P(T) = 3/5 = 0.6)
- Given the same table as in the previous question, what is the probability that a person, who lives in the 44005 zip code and has bought organic coffee, will likely buy the organic tea?

CustomerID	Zipcode	Bought Organic Coffee	Bought Organic Tea
1	44005	yes	yes
2	44001	no	no
3	44001	yes	yes
4	44005	no	no
5	44003	yes	no
6	44005	no	yes
7	44005	no	no
8	44001	no	no
9	44005	yes	yes
10	44003	yes	yes

  - 1
    - (P(T|zip = 44005,C) = P(44005, C, T)/P(44005,C) = 1)

Review: Random Variables and their Distributions

Discrete Random Variables

- Let x be a discrete random variable that can take any of the m different in the set $V=\{v_1, v_2, ..., v_m\}$ with respective probabilities $\{p_1, p_2, ... p_m)$, i.e., $p_i=Prob[x = v_i]$
  - $p_i \geq 0, \Sigma_{j=1, ..., m} p_j = 1$
- Probability Mass Function $P(x)$ is used to represent the set of probabilities $\{p_1, p_2, ..., p_m\}$
  - $P(x) \geq 0, \Sigma_{x \text{ in } V} P(x) = 1$

Expected Value (Means) & Variance

- The expected value (mean) of *x*, E[x], often denoted μ
  - $\mu = E[x] = \Sigma_{x \text{ in } V} xP(x)$
- The expected value of a function f(x), E[f(x)],
  - $E[f(x)] = \Sigma_{x \text{ in } V} f(x)P(x)$
- E[ ] is linear when viewed as an operator
  - $E[\alpha f(x) + \beta g(x)]$
- The variance of x, Var[x], often denoted σ^2
  - $\Sigma^2 = Var(x) = E[x-\mu)^2] = \Sigma_{x \text{ in } V} (x-\mu)^2 P(x)$

Joint Distributions

- Consider a pair of discrete random variables, *x* and *y*, taking values in V-{v_1, v_2, ..., v_m} and W={w_1, w_2, ..., w_n} respectively
  - (x, y) to take a pair of values (v_i, w_j) with probability p_{ij}
  - Or, we can consider the joint probability mass function *P(x, y)*

Marginal Distributions

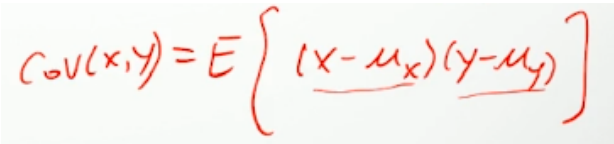
- Knowing *P*(x,), can we figure out $P_x(x)$ of $P_y(y)$
- The concept of marginal distribution for x and y respectively

Statistical Independence

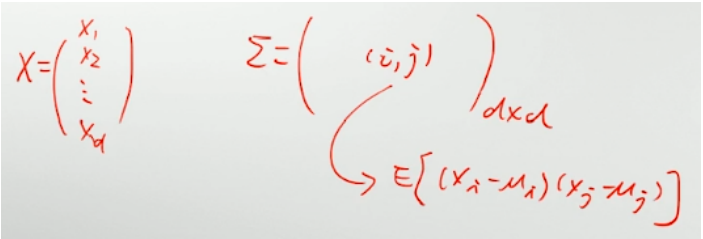
- Random variables *x* and *y* are said to be statistically independent if and only if $P(x, y) = P_x(x)\, P_y(y)$

Covariance

- Cov(x, y), often denoted σ_{xy}



- Covariance matrix Σ, $\Sigma = E[x-\mu)(x-\mu)^t]$



Conditional Density

- $P(x|y) = \frac{P(x, y)}{P(y)}$
- Similarly, we may write the Bayes Rule in terms of densities

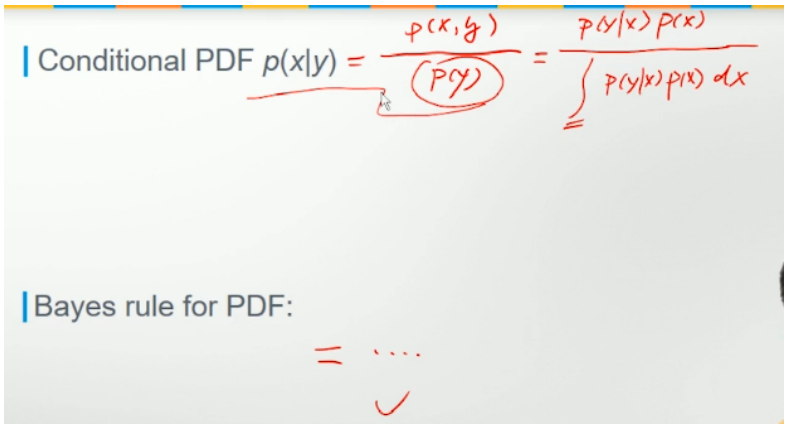
  $P(x|y) = \frac{P(x, y)}{P(y)} = \frac{P(x)P(x|y)}{\Sigma\, P(x)P(y|x)}$

How about continuous random variables?

- Instead of P(x), we have the probability density function (PDF)p(x)
- Some properties of p(x):
  - $P(x) \geq 0$
  - $\int_{-\infty}^{\infty} P(x)\, dx = 1$
- The cumulative distribution function (CDF) F(x):
  - $F(x) = \int_{-\infty}^{\infty} P(t)\, dt$

Continuous Random Variables

- Mean, variance, etc., are similarly defined, via integrals
  - $E(x) = \int x(P(x)\, dx$
  - $Var(x) = \int (x - \mu)P(x)\, dx$
- Joint PDF p(x, y) of two variables
  - Marginal PDFs for *x* and *y*
  - If $x \sim P_x(x)$ and $y \sim P_y(y)$ are independent $p(x, y) = p_x(x)\, p_y(y)$
- Conditional PDF $p(x|y) =$



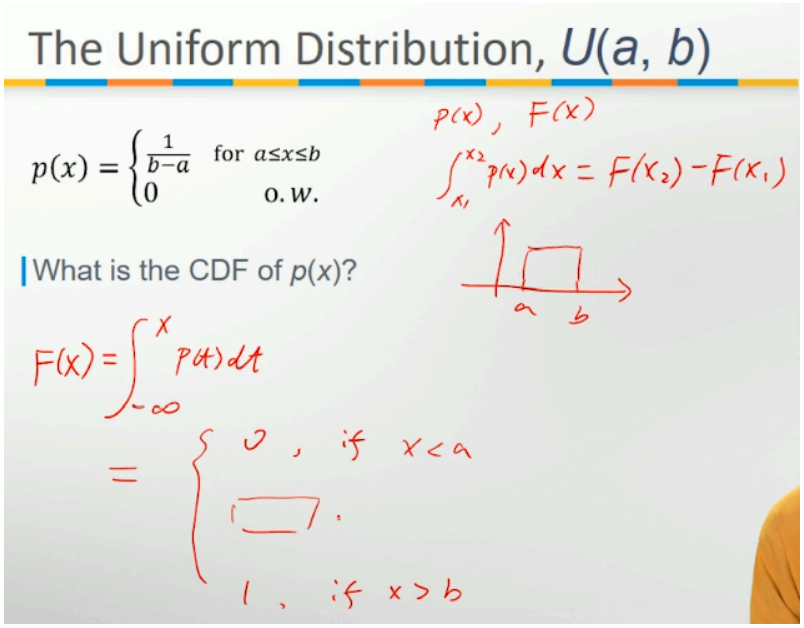
Common Densities in Machine Learning

Common Distributions

- Uniform Distribution
- Normal (Gaussian) Distribution

The Uniform Distribution, $\cup\ (a, b)$

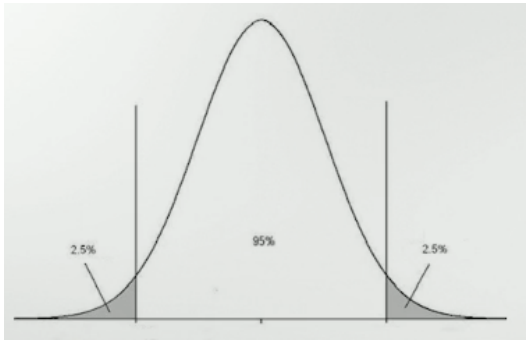
- 1-D example, with PDF
  - $p(x) = \left\{\frac{1}{b-a} \; for \; a \leq x \leq b\right\}, \{0 \; o.w.\}$
  - $$p(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{o.w.} \end{cases}$$
- What is the CDF of p(x)?



The Normal Distribution, $N(\mu, \sigma^2)$

- 1-D example, with PDF

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



- What is the mean and variance?

- $E[X] = \int x\, p(x)\, dx = \int x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \mu$

  $\sigma_X = Var(X) = \int (x-\mu)^2 p(x)\, dx = \dots =$

Standardized Normal Distribution

- 1-D example, with PDF $p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$
- What is the CDF?

  $\int_{-\infty}^{x} p(t)\, dt = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \dots$

  $= \frac{1}{2}\left[1 + \cdot\right.$
- The error function

  $erf(u) = \frac{2}{\sqrt{\pi}} \int_0^u e^{-x^2} dx$

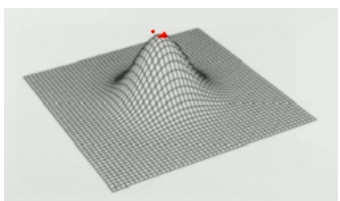
CDF for General Normal Distribution

- $p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- What is the CDF for $N(\mu, \sigma^2)$?

  $\int_{-\infty}^{x} p(t)\, dt = \dots = \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)\right]$

Multivariate Normal Distribution

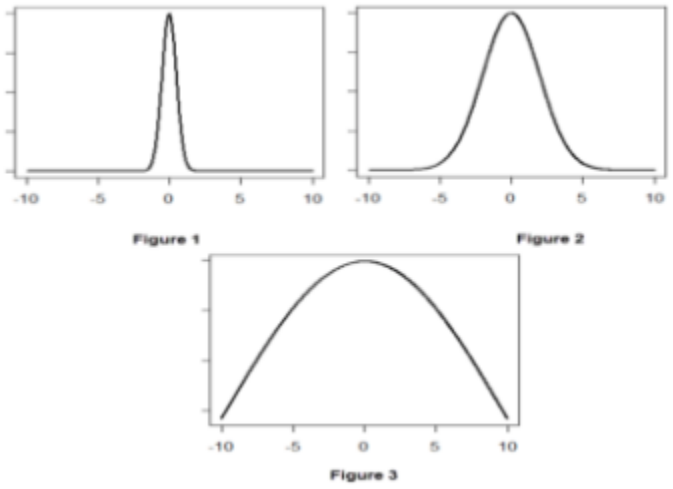
- *d*-dimensional vector **x** is said to be of multivariate normal distribution of its PDF is of the form
  - $p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp[-\frac{1}{2}(x - \mu)^t \Sigma^{-1}(x - \mu)]$
- Visualization of a 2-*d* example



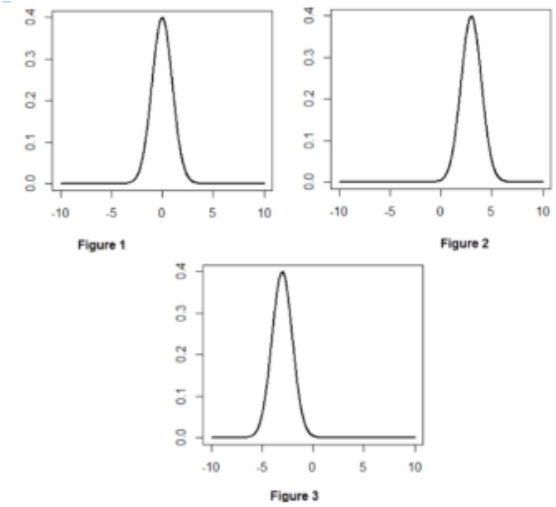
- Given some data **x** distributed according to the above density, we may apply some transformation to **x**, so that the covariance matrix of the transformed data is diagonal

Knowledge Check

- Assume that X is a uniformly distributed random variable that takes values from 1 to 20. What is the value of PMF(X=15)?
    - 1/20
        - (PMF(X = x) = 1/20)
- The following figures represent normal distributions with different standard deviations. Which figure represents the normal distribution with the smallest standard deviation?

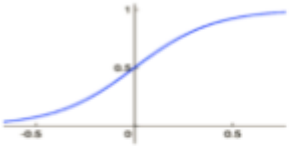


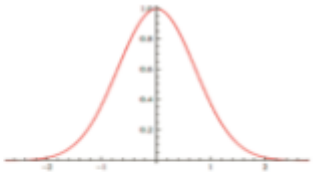
    - Figure 1
- The following figures represent normal distributions with different means. Which figure represents the normal distribution with the largest mean?

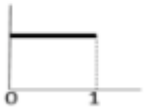


    - Figure 2

        - The more left the peak is, the smaller the mean is.
- Which of the following figures represents the PDF from a uniform distribution?



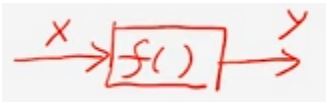




    - Figure 3
- Consider a wireless cell with four channels. Each channel is in one of two states: busy and available. Both states are equally probable, and each channel is independent of any other channel. Define a random variable X to be the number of channels in the available state. What is the value of E[X]?
    - 2
        - (p(x=available) = 0.5, E(X) = 1*P(1 channel is available)+ 2*P(2 channel is available)+ 3*P(3 channel is available)+ 4*P(4channel is available) = 1*4*0.5^4 +2*6* 0.5^4+3*4*0.5^4+4*1*0.5^4 = 2)
- If variables x and z are statistically independent, which of the following is correct? (select all that apply)
    - Var(x+z) = Var(x) + Var(z)
    - E(x+z) = E(x)+E(z)

Set-Up of Supervised Learning & Regression

Supervised Learning

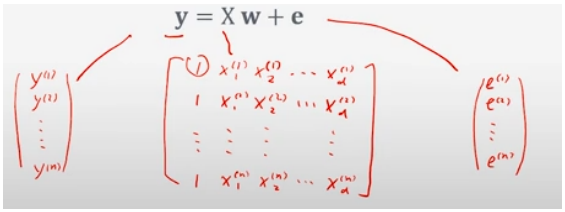
- The set-up: the given training data consist of <sample, label> pairs, or (**x**, y); the objective of learning is to figure out a way to predict label y for any new sample **x**
    - 

- Consider two types of problems:
    - Regression: y continuous
    - Classification: y is discrete, e.g., class labels

The Task of Regression

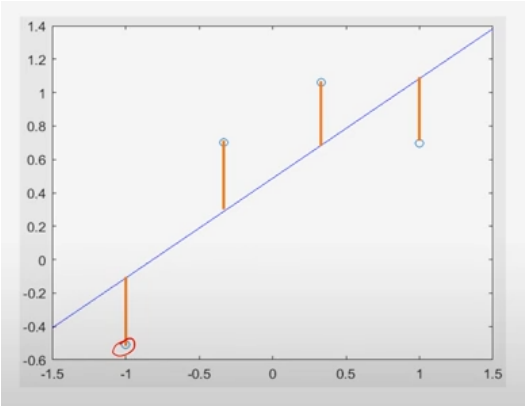
- Given: A training set of n samples $<\mathbf{x}^{(i)}, y^{(i)}$ is a continuous "babel" (or target value) for $\mathbf{x}^{(i)}$
- To learn a model for predicting y for any new sample **x**
- A simple model is linear regression: modeling the relation between y and **x** via a linear function
    - $y \approx w_0 + w_1 x_q + \ldots + w_d x_d = w^t x$

Linear Regression

- We can introduce an error term to capture the residual
  - $y = w^t x + e$
- Applying this to all n samples, we have:
  - $y = Xw + e$
  - 

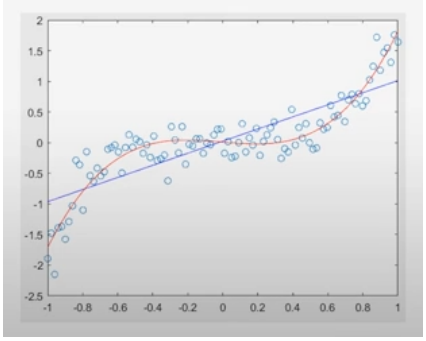
- Learning in this case is to figure out a good **w**
- Find an optimal **w** b minimizing the squared error
  - $||e||^2 = ||y - Xw||^2$
- The solution can be found to be:
  - $w = (X^t X)^{-1} X^t y$
- In practice, some iterative approaches may be used (e.g., gradient descent search).

A Simple Example



Generalizing Linear Regression

- Introducing some basis functions $\phi_j(x)$:
  - $y = w_0 + w_1 \phi_q(x) + \ldots + w_{M-1} \phi_{M-1}(x)$

- 
- Compare:
  - Blue: Linear Regression
  - Red: with $\phi_j(x) = x^j$

Regularized Least Squares

- E.g.m use a new error function:
  - $E_D(w) + \lambda E_w(w)$
  - λ is the regularization coefficient
  - $E_D(w)$ is the data dependent error
  - $E_w(w)$ is the regularization term, e.g., $E_w(w) = ||w||^q$
- Help to alleviate overfitting

Knowledge Check

- Given the regression equation as Y = 4X - 2. What will be the predicted Y for a X = 4?
  - 14
    - Y = 4X – 2=4*4-2 = 14
- Suppose it is possible to predict a person's score on Test B from the person's score on Test A. The regression equation is B = 2.3A + 10.5. What is a person's predicted score on Test B assuming this person got a 40 on Test A?
  - 102.5
    - B = 2.3A + 10.5 = 2.3*40+10.5 = 102.5
- Suppose a person got a score of 32.5 on Test A and a score of 95.25 on Test B. Using the same regression equation as in the previous problem (B = 2.3A + 10.5), what is the error of prediction for this person?
  - 10
    - Error = || B -2.3A + 10.5|| = ||95.25-32.5*2.3-10.5|| = 10
- Given a set of 2-D points, we want to fit a straight line to the points. Which of the following would be a useful criterion for choosing the best line?
  - The line should minimize the sum of square distances of the points to the line
- The mean of X is 4 and the mean of Y is 7. The regression line that predicts Y from X should go through the point (4,7).
  - True

- The goal of linear regression is to minimize the sum of squared distances of all the points to the line, so it must go through the point of $(\bar{x}, \bar{y})$

Classification & Density Estimation

Examples of Image Classification

- The MNIST training images of hand-written digits

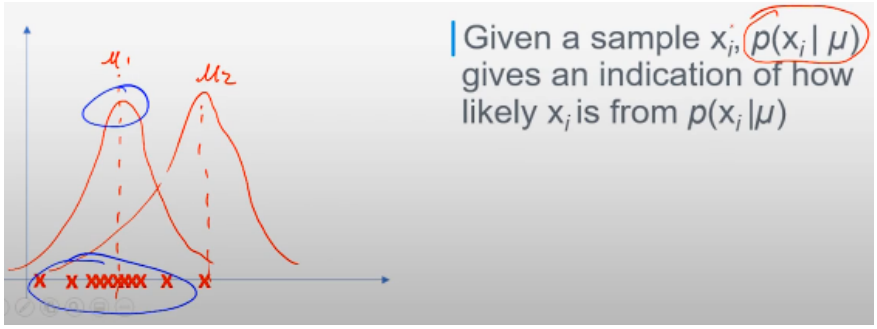
- The Extended Yale B Face Images

How do we model the training images?

- Parametric: each class of images (the feature vectors) ma be modeled by a density function $p_\theta(x)$ with parameter θ
  - To emphasize the density is for images from class/label y, we may write $p_\theta(x|y)$
  - We ma also use the notation $p(x|\theta)$ if the discussion is true for any y
- How to estimate θ from the training images?
- Note: We may also consider non-parametric approaches

MLE for Density Estimation (⅓)

- Given some training data; Assuming a parametric model $p(x|\theta)$; what specific θ will fit/explain the data best?
  - E.g., Consider a simple 1-D normal density with only a parameter μ (assuming the variance is known)



MLE for Density Estimation (2/3)

- The likelihood function: the density function $p(x|\theta)$ evaluated at the given data sample $\mathbf{x}_j$, and viewed as a function of the parameter θ
  - Assessing how likely the parameter θ (defining the corresponding $p(x|\theta)$ gives arise to the sample $\mathbf{x}_j$
  - We often $L(\theta)$ to denote the likelihood function, and $l(\theta)=\log(L(\theta))$ is called the log-likelihood
- Maximum Likelihood Estimation (MLE): Finding the parameter that maximizes the likelihood function
  - $\hat{\theta} = argmax_{\theta} p(x|\theta)$

MLE for Density Estimation (3/3)

- How to consider all the given samples $D=\{x_i, i=1, ..., n\}$?
- The concept of i.i.d. samples: the samples are assumed to be independent and identically distributed
- So, the data likelihood is give by
  - $L(\theta) = P(D|\theta) = \prod_{i=1}^{n} P(x_i|\theta)$

MLE Example 1

- Tossing a coin for n times, observing n_1 times for head
  - Estimate the probability θ for head
- The likelihood function is:
  - $L(\theta) = P(D|\theta) = \theta^{n_1}(1-0)^{n-n_1}$
- We want to find what θ maximizes this likelihood, or equivalently, the log-likelihood
  - $l(\theta) = logP(D|\theta) = log(\theta^{n_1}(1-0)^{n-n_1}) = n_1 \log\theta + (n-n_1)\log(1-\theta)$
  - Take the derivative and set to 0: $\frac{d}{d\theta} l(\theta) = 0$
  - This will give us: $\hat{\theta} = \frac{n_1}{n}$

MLE Example 2

- Given n i.i.d. samples $\{x_i\}$ from the 1-D normal distribution $N(\mu, \sigma^2)$ find the MLE for μ and σ^2

- The likelihood function is: $L(\mu,\ \sigma)\ =\ p(D|\mu,\sigma)\ =\ (\frac{1}{\sigma\sqrt{2\pi}})^{n}\ \prod_{i=1}^{n}\ e^{-\frac{(xi-\mu)^{2}}{2\sigma^{2}}}$
- The log-likelihood is:

$$\text{The log-likelihood is: } l(\mu,\sigma) = \log P(D|\mu,\sigma)$$
$$= \log\left(\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^{n}\prod_{i=1}^{n} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right)$$
$$= -n\log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}$$

- The MLE solution for μ
  - $$\hat{\mu} = \text{argmax}_{\mu} l(\mu,\sigma)$$
  $$= \text{argmax}_{\mu}\{-n\log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}\}$$
  - Set the derivative to 0: $\frac{\partial}{\partial\mu} l(\mu,\sigma) = 0$
  - The solution is: $\widehat{\sigma^2} = \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}$

Knowledge Check

- Assume that x_1, ..., x_n are i.i.d samples drawn from the same underlying distribution. Assume that the underlying distribution is Gaussian $N(\mu, \sigma^2)$. What is the MLE estimator of μ?
  - $\frac{1}{n}\sum_{i=1}^{n} = x_i$
    - $L(\mu,\sigma) = p(D|\mu,\sigma) = (\frac{1}{\sigma\sqrt{2\pi}})^{n}\prod_{i=1}^{n} = e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} = \log((\frac{1}{\sigma\sqrt{2\pi}})^{n}\prod_{i=1}^{n} e - \frac{(x_i-\mu)^2)}{2\sigma^2}) = -n\log(\sigma\sqrt{2\pi})$ $- \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}$

      The MLA solution for μ is:

      $$\hat{\mu} = argmax_{\mu}\ l(\mu,\ \sigma)\ = argmax_{\mu}\left\{-\ nlog(\sigma\sqrt{2\pi})\ -\ \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}\right\}$$

- If the true value of μ is known, then the MLE estimator of σ^2 is $\sigma^2_{MLE} = \frac{1}{N}\sum_{i-1}^{N}(X_i - \mu)^2$. Define the bias of the estimator as $E\left[\sigma^2 - \sigma^2_{MLE}\right]$. If the bias is 0, we say the estimator is unbiased. In this case, is σ^2_{MLE} unbiased?
  - The estimator is right, and it is unbiased
- Given a die with probability p of rolling an odd number, what is the likelihood function for rolling a particular sequence with α_0 odd numbers and α_e even number?

- $p^{\alpha_0}(1-p)^{\alpha_e}$

  The probability of α_0 odd numbers if p^{α_0}, and the probability of α_e even number is $(1-p)^{\alpha_e}$, so the likelihood function for rolling a particular sequence with α_0 odd numbers and p^{α_0} even number is $p^{\alpha_0}(1-p)^{\alpha_e}$.

- Suppose the scores of randomly selected students are normally distributed with unknown mean and standard deviation. A random sample of 10 students returns the following scores: [81, 71, 71, 74, 56, 92, 83, 74, 91, 66]. Which of the following is closest to the estimated σ by using maximum likelihood estimation?
  - 11
  - $\widehat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{81+71+71+74+58+92+83+73+91+66}{10} = 75.9$

    $\widehat{\alpha^2} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{(81-75.9)^2+(71-75.9)^2+(74-75.9)^2+(56-75.9)^2+(92-75.9)^2+(83-75.9)^2+(74-75.9)^2+(91-75.9)^2+(66-75.9)^2}{10}$

    $= 111.29,\ \widehat{\sigma} = \sqrt{\widehat{\sigma^2}} = 10.54 \approx 11$

Generative & Discriminative Models

Two Types of Models

- Generative Model
  - $P(y|x) \propto P(y)p(x|y)$
  - To learn $P(y)$ and $p(x|y)$
  - Bayesian learning, Bayes classifiers
  - Example: Naïve Bayes Classifier
- Discriminative Model
  - Directly learn $P(y|x)$
  - No assumption made on $p(x|y)$
  - Example: Logistic Regression

Practical Difficulty of Bayesian Learning

- Consider doing Bayesian learning without making simplifying assumptions
  - Given n training paris $< x^{(i)}, y^{(i)} >, i = 1, ..., n$. Each $x^{(i)}$ is d-dimensional
  - We need to learn $P(y)$ and $p(x|y)$
- $p(x|y)$ can be very difficult to estimate:
  - Consider a very simple case: binary features, and y is also binary. How many probabilities do we need to estimate?
    - 2^{d+1}

Knowledge Check

- Which of the following is true for generative or discriminative classifier models, considering input x and label y ? (Select all that apply)
    - A discriminative classifier models the posterior $p(x|y)$ directly
    - A generative classifier employs the prior $p(y)$
    - A generative classifier makes user of $p(x|y)$
- Which of the following is true for generative or discriminative classifier models, consider input x and label y? (Select all that apply)
    - A discriminative classifier models the posterior $p(x|y)$

Module 2 Quiz Questions

- Given two sets A=[1,2,3,4,5], B=[1,4,7,9,10]. What is A ∩ B?
    - {1, 4}
- Which of the following is always correct?
    - trace(X+Y)=trace(X)+trace(Y)
- If x ~ $p_x(x)$ and y ~ $p_y(y)$ are independent, what is p(x|y) =?
    - $p_x(x)$
- In a multivariate Gaussian distribution, if the "Σ" in the PDF is a diagonal matrix, what does it imply?
    - The features are statistically independent
- Consider a wireless cell with four identical channels. Each channel is one of two states: busy or available. Both states are equally probable and each channel is independent of any other channel. Define a random variable X to be the number of channels in the busy state. What is the value of E(X)?
    - 2
- Assume that X is uniformly distributed random variable that takes integer values from 1 to 40. What is the value of PMF(X=20)?
    - 1/40
- Consider a symmetric matrix A. What is $\frac{\partial(x^T Ax)}{\partial x}$?
    - $2Ax$
- The table below shows the purchase history of 10 customers from a set of zip codes that bought organic tea or organic coffee. Using Bayes's Rule, what is the probability that a person who lives in the 44005 zip code and bought organic coffee would likely not buy the organic tea?

CustomerID	Zipcode	Bought Organic Coffee	Bought Organic Tea
1	44005	yes	yes
2	44001	no	no
3	44001	yes	yes
4	44005	no	no
5	44003	yes	no
6	44005	no	yes
7	44005	no	no

8	44001	no	no
9	44005	yes	yes
10	44003	yes	yes

- 0

- The table below shows the purchase history of 10 customers from a set of zip codes that bought organic tea or organic coffee. What is the prior probability that a customer came from area with zip code 44001?

CustomerID	Zipcode	Bought Organic Coffee	Bought Organic Tea
1	44005	yes	yes
2	44001	no	no
3	44001	yes	yes
4	44005	no	no
5	44003	yes	no
6	44005	no	yes
7	44005	no	no
8	44001	no	no
9	44005	yes	yes
10	44003	yes	yes

- 0.3

Naive Bayes Classifier

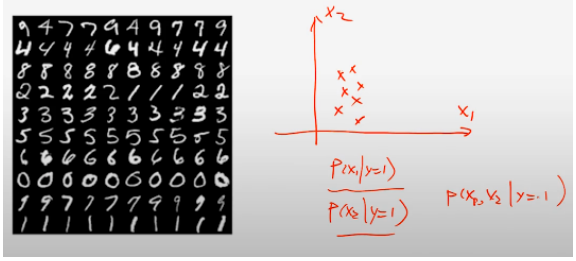
Naive Bayes Classifier

- The "naive" *conditional independence* assumption: each feature is (conditionally) independent of every other feature, given the label, i.e., $p(x_i|\{x_j \; for \; any \; j \neq i\}, \; y) = p(x_i|y)$
- How does this assumption simplify the problem?
  - Consider the previous example again: d-dimensional binary features, and y is also binary
    - 2^{d+1}
  - How many probabilities do we need to estimate now?
    - $4d$
- $p(\mathbf{x}|y) = p(x_1, x_2, \ldots, x_d|y) = \ldots \prod_{i=1}^{d} p(x_i|y)$
- The naive Bayes classifier: the predicted label is given by

$$\hat{y} = argmax \, P(y) \prod_{i=1}^{d} p(x_i|y)$$

- "Parameters" of the classifier:
  - $P(y)$
  - $p(x_i|y) \; for \; all \; i, \; y$
- E.g., estimating the parameters of the classifier
  - $P(y)$ & $p(x_i|y) \; for \; all \; i, \; y$

    for the following example:



Discrete Feature Example

- $x =< x_1, x_2, ..., x_d >$ where each x_i can take only a finite number of values from $\{v_1, v_2, ..., v_m\}$:
- In this case, the "parameters" of the classifier are
  - $P(y)$
  - $p(x_i = v_k|y)\ for\ all\ i,\ k,\ and\ y$
- Given: A training set of n labelled samples $< x^{(i)}, y^{(i)}, i = 1, ..., n >$
  - How to estimate the model parameters?
    - $P(y) = \frac{\#\ of\ samples\ with\ label\ y}{n}$
    - $p(x_i = v_k|y) = \frac{P(x_i=v_k,y)}{P(y)} = \frac{\frac{\#\ of\ samples\ with\ ith\ feature\ taking\ value\ v_k\ \&\ label\ y}{n}}{P(y)}$
- These are in fact the MLE solutions for the corresponding parameters

Knowledge Check

- For a classification problem using two features X and Y, the Naïve Bayes Model assumes that X and Y share the same distribution.
  - False
- Applying a Naïve Bayes Model for classifying features with normal densities, which of the following is always true?
  - Given the label, the features are assumed to be independent
- Given the following table, where X is the features and Y is the label, we want to train a binary classifier, with (1) the last column being the class label (i.e., whether to enjoy the sport); and (2) each column of X being a binary feature. How many independent parameters are there in the Naïve Bayes classifier?

	X					Y
Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
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

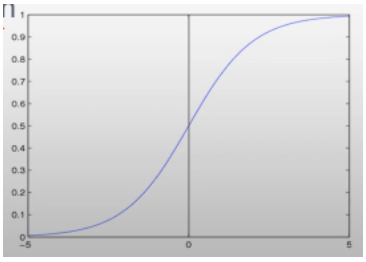
  - 13
    - $(6 * 2 + 1) = 13$
- Using the same format as the table, given a new (test) example x = (sunny;warm; high; strong; cool; change), what is $P(y) = Yes|x)$?

	X					Y
Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
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

- 1
  - The correct way of answering this problem is outline below:
  - P(Y|X) = P(X,Y)/P(X) = P(X|Y)P(Y)/P(X).
  - You may estimate P(Y) by looking at the last column.
  - So let's look at P(X|Y) and P(X)
  - P(X|Y): this is where we can use the Naïve Bayes assumption, so this can be factored into P(x1|Y) ...P(x6|Y).
  - P(X): this can be found by sum_over_Y {P(X,Y} = sum_over_Y{P(X|Y)P(Y)}. Now P(X|Y) and P(Y) are found above already. So the sum can be computed.
  - Plug everything back into P(X|Y)P(Y)/P(X), we can find the solution to be 1.
- What is your estimation for $P(x_5 = warm|y = Yes)$?
  - ⅔
    - P(x5=warm|y=Yes) = P(x5=warm,y=Yes)/P(y = Yes) = 2/3

Logistic Regression

Discriminative Model: Example



- Again, we are given a training set of n labelled samples $< x^{(i)}, y^{(i)} >$
- Why not directly model/learn $P(y|x)$?
  - Discriminative model
- Further assume $P(y|x)$ takes the form of a logistic sigmoid function
  - Logistic Regression

Logistic Regression

- Logistic regression: use the logistic function for modeling $P(y|x)$, considering only the case of $y \in \{0, 1\}$
  - $P(y = 0|x) = \frac{1}{1+exp(w_0+\sum_{i=1}^{d} w_i x_i)} = \frac{1}{1+e^{w^t x}} = 1 - \bar{v}(w^t x)$
  - $P(y = 1|x) = \frac{exp(w_0+\sum_{i=1}^{d} w_i x_i)}{1+exp(w_0+\sum_{i=1}^{d} w_i x_i)} = \frac{e^{w^t x}}{1+e^{w^t x}} = \bar{v}(w^t x)$
  - The logistic function:

    $\sigma(t) = \frac{1}{1+e^{-t}} = \frac{e^t}{1+3^t}$

Logistic Regression → Linear Classifier

- Given a sample x, we classify it as 0 (i.e., predicting $y = 0$) if:
  - $P(y = 0|x) \geq P(y = 1|x)$

    $\frac{1}{1+e^{w^t x}} \geq \frac{e^{w^t x}}{1+e^{w^t x}} = 1 \geq e^{w^t x}$

    $0 \geq w^t x \Leftrightarrow if\ w^t x > 0\ we\ classify\ x\ as\ 1$
- This is a linear classifier $w^t x$

The Parameters of the Model

- What are the model parameters in logistic regression?
- Given a parameter w, we have $P(y|x) = \bar{v}(w^t x)^y (1 - \bar{v}(w^t x))^{1-y}$
  - $\left[\bar{v}(w^t x)\right]^y \left[1 - \bar{v}(w^t x)\right]^{1-y}$
- Suppose we have two different sets of parameters, $w^{(1)}$ and $w^{(2)}$, whichever giving a larger $P(y|x)$ should be better parameter.

The Conditional Likelihood

- Given n training samples, $< x^{(i)} y^{(i)} >$, $i = 1, \ldots, n$, how can we use them to estimate the parameters?
  - For a given w, the probability of getting all those $y^{(1)}, y^{(2)}, \ldots, y^{(n)}$ from the corresponding data $x^{(i)}$, $i = 1, \ldots, n$, is
    - $P\left(y^{(1)}, y^{(2)}, \cdots, y^{(n)} \middle| x^{(1)}, x^{(2)}, \cdots, x^{(n)}, w\right) = \prod_{i=1}^{n} P(y^{(i)}|x^{(i)}, w)$

      $= \prod_{i=1}^{n} \left[\sigma(w^t x^{(i)})\right]^{y^{(i)}} \left[1 - \sigma(w^t x^{(i)})\right]^{1-y^{(i)}}$
  - Call this $L(w)$, the conditional likelihood

The Conditional Log-likelihood

$$l(w) = \log L(w) = \log \prod_{i=1}^{n} (\cdots)$$

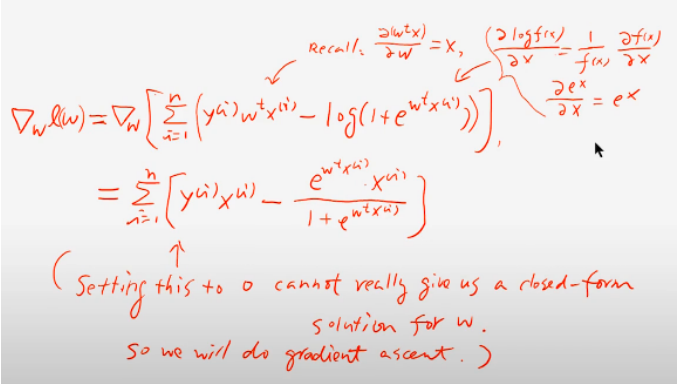
$$= \sum_{i=1}^{n} \log\left[\sigma(w^t x^{(i)})^{y^{(i)}} (1 - \sigma(w^t x^{(i)}))^{1-y^{(i)}}\right]$$

$$= \sum_{i=1}^{n} \left[\log\left(\sigma(w^t x^{(i)})^{y^{(i)}}\right) + \log\left((1 - \sigma(w^t x^{(i)}))^{1-y^{(i)}}\right)\right]$$

Maximizing Conditional Log Likelihood

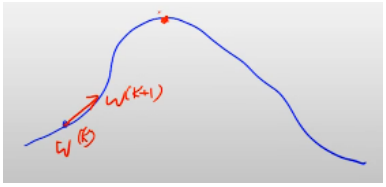
- Optimal parameters
  - $w^* = argmax_w l(w) = argmax_w \sum_{i=1}^{n} \left[y^{(i)} w^t x^{(i)} - log(1 + exp(w^t x^{(i)})) \right]$
- We cannot really solve for w^* analytically (no closed-form solution)
  - We can use a commonly-used optimization technique, gradient descent/ascent, to find a solution

Finding the gradient of $l(w)$



Gradient Ascent Algorithm

- The algorithm: iterate until coverage:
  - $w^{(k+1)} = w^{(k)} + \eta \nabla_{w^{(k)}} l(w)$
- $\eta > 0$ is a constant called the learning rate



Knowledge Check

- Which of the following is always true?
  - Naive Bayes estimates prior and conditional densities
    - Naïve Bayes estimates prior and conditional densities.
- Linear regression is a supervised learning technique.
  - True
- 2-class logistic regression is a non-linear classifier.
  - False
    - 2-class logistic regression is a linear classifier.
- For a given class C, which assumption about the attributes $A_1, A_2, \ldots A_n$ makes the following conditional joint probability equation hold? $P(A_1, A_2, \ldots A_n|C) = \prod_{t=1}^{n} P(A_i|C)$
  - $A_1, A_2, \ldots A_n$ are conditionally independent
    - $P(A_1, A_2, \ldots A_n|C) = \prod_{t=1}^{n} P(A_i|C)$ *implies* $A_1, A_2, \ldots A_n$ are conditionally independent

- Researchers at a medical center are interested in exploring the relationship between patient age (X11), weight (X22), and the presence (1) or absence (0) of a particular disease. If researchers decide to use Logistic Regression, which of the following would be interpreted as the probability that the disease is present?
    - $Probability = \frac{e^{\beta_0+\beta_1 r_1+\beta_2 r_2}}{1+e^{\beta_0+\beta_1 r_1+\beta_2 r_2}}$
        - $\sigma(t) = \frac{1}{1+e^{-t}} = \frac{e^{\beta_0+\beta_1 r_1+\beta_2 r_2}}{1+e^{\beta_0+\beta_1 r_1+\beta_2 r_2}}$

Linear Machines - Part 1: Basics

Revisiting Logistic Regression

- In logistic regression: given a training set of n labelled samples $< x^{(i)}, y^{(i)} >$, we learn $P(y|x)$ by assuming a logistic sigmoid function
    - We end up with a *linear classifier*

      $w^t x \leq 0, \ class\ 0$

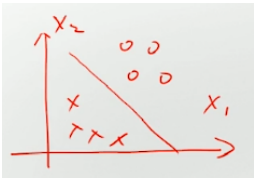
      $w^t x > 0, \ class\ 1$
    - $g(x) = w^t x$ is called the *discriminant function*

Linear Discriminant Functions

- In general, taking a discriminative approach, we can assume some form for the discriminant function that defines the classifier
    - The learning task is to use the training samples to estimate the parameters of the classifier

Linear Decision Boundaries

- Linear discriminant function give arise to liner decision boundaries
    - linear classifiers or linear machines
- We will use both notations:
    - $g(x) = w^t x$ or $g(x) = w^t x + w_0$



Linear Machines for C>2 Classes

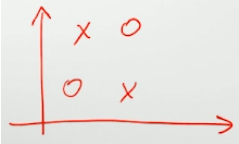
- We can define C linear discriminant functions:
    - $g_i(x) = w_i^t x, \ i = 1, 2, \ldots, C$
- What is the decision rule for the classifier?
    - $g_i(x) \geq g_j(x), \ \forall j \neq i$
        - classify x as class i

The Learning Task

- Finding w_i, $i = 1, 2, ..., C$
- Let's use the 2-class case as an example
    - For n samples $x_1, ..., x_n$, of 2 classes ω_1 and ω_2, if there exists a vector w such that $g(x) = w^t x$ classifies them all correctly → Finding w
    - i.e. finding w such that
        - $w^t x_i \geq 0$ for samples of ω_1 and
        - $w^t x_i < 0$ for samples of ω_2

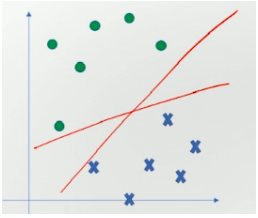
Linear Separability

- If we can find at least one vector w such that $g(x) = w^t x_i$ classifies all samples
    - We say the samples are linearly separable
- As example of not linearly separable in 2-D:



The Solution Region

- There may be many different weight vectors that can all be valid solutions for a given training set
    - The solution regions
- If the solution vector is not unique, which one is the best?



Solving for the Weight Vector

- Consider the following approach: finding a solution vector which optimizes some objective function
    - We may introduce additional constraints for a "good solution"
    - Solving a constrained optimization problem
- Theoretical: Lagrange or Karush-Kuhn-Tucker
- In practice: e.g., gradient-descent-based search

Gradient Descent Procedure

- Basic Idea:
    - Define a cost function $J(w)$

    - Starting from an initial weight vector $w(0)$
    - Update w by
        - $w(k+1) = w(k) - \eta(k)\nabla J(w(k))$
- $\eta > 0$ is the learning rate

Knowledge Check

- Logistic Regression is a linear classifier.
    - True
- Given a dataset of two classes, we can always find at least one vector w such that $g(x) = w^t x$ classifiers all samples correctly
    - False
- Given the objective function and additional constraints, how can we find the solution that optimizes the object function? (Select all that apply)
    - Find a solution to the Karush-Kuhn-Tucker condition
        - We can gradient-descent based method to update the object function, and using Lagrange multiplier to add the constraints, or find a solution to the Karush-Kuhn-tucker condition.
    - Using gradient-descent based method
        - We can gradient-descent based method to update the object function, and using Lagrange multiplier to add the constraints, or find a solution to the Karush-Kuhn-tucker condition.
    - Using Lagrange multiplier
        - We can gradient-descent based method to update the object function, and using Lagrange multiplier to add the constraints, or find a solution to the Karush-Kuhn-tucker condition.
-

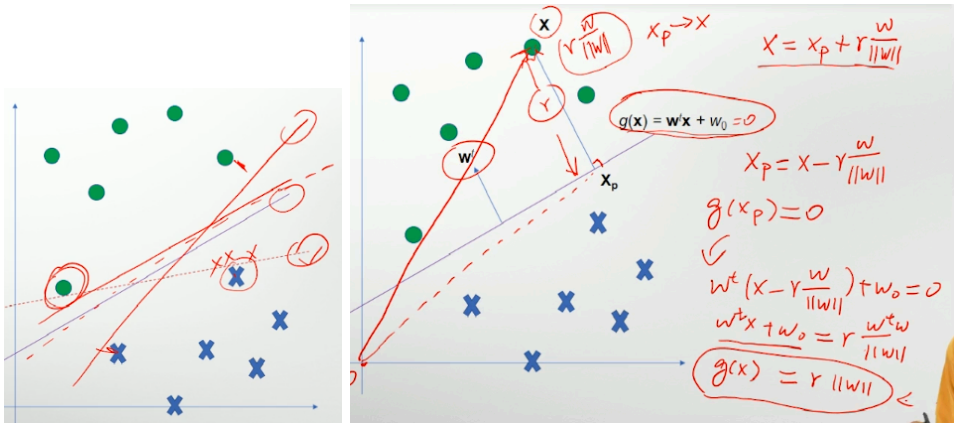
Linear Machines - Part 2: The Concepts of Margins

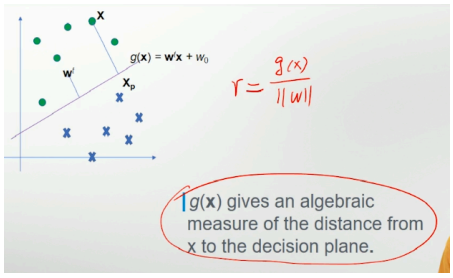
Illustrating Linear Boundaries

- The decision boundaries is given by the line $g(x) = 0$
    - For appreciating a geometric interpretation, we will write w_0 explicitly, i.e., we have
        - $g(x) = w^t x + w_0$
- The normal vector of the decision line/plane is ____

Which one is better?

- Consider the distances of the samples to the decision plane

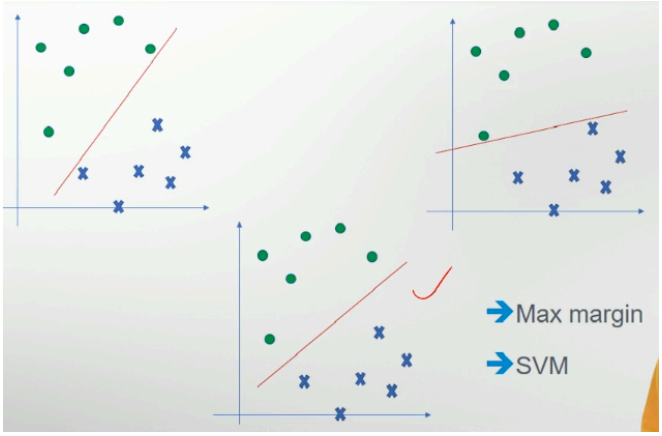
- 

- $g(x)$ gives an algebraic measure of the distance from x to the decision plane
  - 

The Concept of Margins

- Let $g(x) = 0$ be a decision plane
  - The margin of a sample x (w.r.t. The decision plane) is the distance from x to the plane
  - For a given set of sample S, the margin (w.r.t. a decision plane) is the smallest marin over all x in S
- For a given set, a classifier that gives rise to a larger margin will be better

Use Margins to Compare Solutions



Knowledge Check

- In the following figure, which one has the best margin?

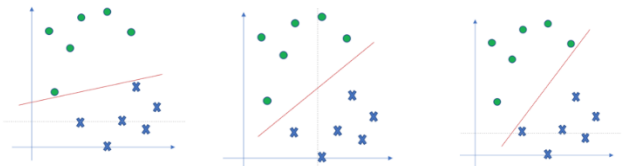


Fig.1 Fig.2 Fig.3

- Fig. 2

- For a given dataset, a classifier that gives rise to a larger margin is in general better than those with smaller margins?
  - True

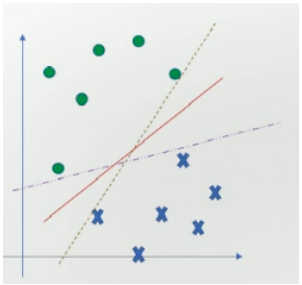
Module 3 Quiz Questions

- Let A_1, A_2, ..., A_n be mutually exclusive events that exhaust the probability space Y. Which of the following conditions is true?
  - $\sum_{i=1}^{n} P(A_i) = 1$
- How can Logistic Regression be used as a classifier?
  - Logistic Regression can be used as a classifier by using a threshold on the outcome of the Logistic function and using the threshold to classify the inputs.
- Select the answers that best complete the following statement. Unlike Naïve Bayes, which is a __ model, Logistic Regression is a __ model.
  - Generative, discriminative
- If the true value of μ is known, then the MLE estimator of σ^2 is $\sigma^2_{MLA} = \frac{1}{N}\sum_{i=1}^{n} (x_i - \mu)^2$ is the estimation of σ^2 unbiased?
  - No
- Which of the following is true for comparisons of Naïve Bayes and logistic regression? (Select all that apply)
  - Naive bayes is applicable only when the data are of normal distributions
  - They can both be used for supervised learning (selected)
  - Logistic Regression leads to a linear classifier (selected)
- Suppose the scores of randomly selected students are normally distributed with an unknown mean and a standard deviation. A random sample of 10 students returns the following scores [81, 71, 71, 74, 56, 92, 83, 74, 91, 66]. Estimate σ by using maximum likelihood Estimation (choose the closest value)
  - 11
- When Tossing a dice several times, let α_0 stand for the number of odd rolls and α_e stand for the number of even roles. Let p be the probability of getting an odd number. Using maximum likelihood estimation, how do you estimate p?

- $\hat{p} = \frac{\alpha_0}{\alpha_e + \alpha_0}$

- Which of the following is true for generative or discriminative classifier models that read the input x and the label y? (Select all that apply)
  - A generative classifier makes it's predictions by using Bayes' rule to calculate P(y|x) (selected)
  - A discriminative classifier models the posterior P(y|x) directly (selected)
  - A discriminative classifier learns a model of the joint probability p(x,y)
  - A generative classifier attempts to learn p(x|y) (selected)

SVM - Part 3: Linearly-Separable Case



Key Idea of Support Vector Machines

- For a given set, a classifier that gives rise to a larger margin will be better
- SVM: To find the decision boundary such that the margin is maximizes
  - $H: w^t x + b = 0$

Formulating the Problem (continued)

- Let $d_+ (d_-)$ be the shortest distance from the separating hyperplane to the closest positive (negative) examples
- These defines planes H_1 and H_2
- We can let $d_+ = d_- = d$
  - Find a solution maximizing 2d

Formulating the Margin

- Given separating plan: $H: w^t x + b = 0$ and distance d, what are the equations for H_1 and H_2?
- Consider the plane H* given by $w^t x + b = ||w||d$
  - Check its orientation
  - CHeckits distance to H
- H_1 is given by $w^t x + b = ||w||d$
  - Similarly, H_2 is given by $w^t x + b = - ||w||d$
- Note: for any p;lan equation, $w^t x + b = 0\{w, b\}$ is defined only up to an unknown scale:
  - $\{sw, sb\}$ is also a valid solution to the equation, for any constant s
- We can have the canonical formulation for all the plans as
  - $H: w^t x + b = 0$
  - $H_1: w^t x + b = 1$
  - $H_2: w^t x + b = - 1$
- The region between H_1 and H_2 is also called the margin, and its width is $\frac{2}{||w||}$

Formulating SVM

$\{w\ *,\ b\ *\} = argmin||w||\ or\ \{w\ *,\ b\ *\} = argmin\frac{1}{2}||w||^2$

- Subject to
  - $w^t x^{(i)} + b \geq 1\ for\ y^{(i)} = \ \ +1$
  - $w^t x^{(i)} + b \leq -1\ for\ y^{(i)} = \ \ -1$
- The constraints can be combined into:
  - $y^{(i)}(w^t x^{(i)} + b) - 1 \geq 0\ \forall i$
- A nonlinear (quadratic) optimization problem with linear inequality constraints

How to solve SVM? (Outline)

- Reformulate the problem using Lagrange multipliers α
  - Lagrangian Primal Problem
  - Lagrangian Dual Problem
- The Karush-Kuhn-Tucker Conditions
  - Necessary and sufficient for $q,\ b,\ \alpha$
  - Solving the SVM problem → finding a solution to the KKT conditions

SVM: Lagrangian Primal Formulation

- Define:
  - $L_p(w,\ b,\ \alpha) = \frac{1}{2}||w||^2 - \sum_i \alpha_i\left[y^{(i)}\left(w^t x^{(i)} + b\right) - 1\right]$
- then the SVM solution should satisfy
  -
- The final w is given by
  - $w = \sum_i \alpha_i y^{(i)} x^{(i)}$
  - and b is given by
    - $y^{(k)} - w^t x^{(k)}\ for\ any\ k\ such\ that\ \alpha_k > 0$

SVM: Lagrangian Dual Formulation

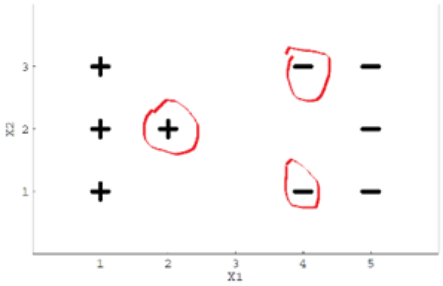
- The objective function is:
  - $L_D(w,\ b,\ \alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} \cdot x^{(j)}$
- The solution is the same as before. But there is an important observation
- Points for which $\alpha_i > 0$ are called support vectors

Knowledge Check

- Support vectors are the data points that lie closest to the decision surface.
  - True
- What is the goal in SVM?
  - To maximize the margin
- If we call the region between two separate planes $H_1 = w^t x + b = 1$ and $H_2 = w^t x + b = -1$, as the margin, what is the width of it?
  - $\frac{2}{||w||}$
    - $$H_1 - H_2 = w^t (x_1 - x_2) = 2 => the width is 2/||w||$$
- The linear SVM's are less effective when
  - The data is noisy and contains overlapping points
    - When the data has noise and overlapping points, there is a problem in drawing a clear hyperplane without misclassifying
- Given the objective function:

  $L_D(w, b, \alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} \cdot x^{(j)}$,

  Which points are the support vectors?
  - $x(i)\ for\ which\ \alpha_i > 0$
- Given this figure illustrating a feature space with two classes of samples (+ and -, or plus and minus), if any point circled in red is removed from the data, will the decision boundary change?



  - Removing any of them will change the decision boundary

Machines and SVM - Part 4: SVM for Non-Linearly-Separable Case

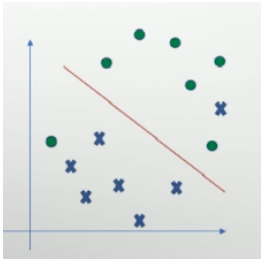
Linear Separability Violated

- Come samples will always be misclassified no matter what $\{w, b\}$ is used

Examining Misclassified Samples

- They will violate the constraints:
  - $w^t x^{(i)} + b \geq 1\ for\ y^{(i)} = +1$
  - $w^t x^{(i)} + b \leq -1\ for\ y^{(i)} = -1$

Relaxing Constraints



- Introducing non-negative slack variables ξ_i
  - $w^t x^{(i)} + b \geq 1 - \xi_i \; for \; y^{(i)} = +1$
  - $w^t x^{(i)} + b \leq -1 + \xi_i \; for \; y^{(i)} = -1$
- For an error to occur, the corresponding ξ_i must exceed unity
  - Hinge loss or soft margin
- $\Sigma_i \xi_i$ provides an upper bound on the number of training errors

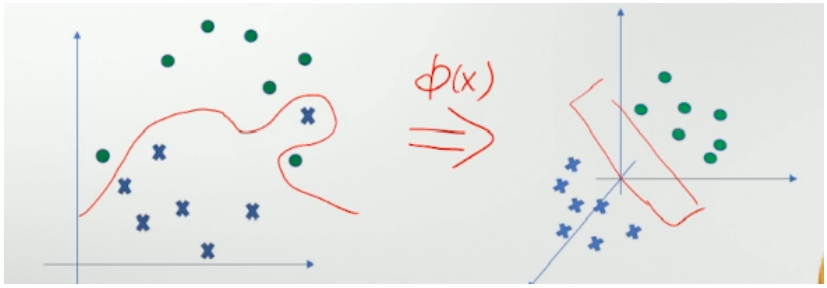
Updating the Formulation

$\{w^*, b^*\} = argmin \frac{1}{2}||w||^2 + C(\Sigma_i \xi_i)$

- Subject to
  - $w^t x^{(i)} + b \geq 1 - \xi_i \; for \; y^{(i)} = +1$
  - $w^t x^{(i)} + b \leq -1 + \xi_i \; for \; y^{(i)} = -1$
  - $\xi_i \geq 0, \; \forall i$
- C is a parameter to control how much penalty is assigned to errors

Are non-linear decision boundaries possible?

- Transform data to higher dimensions using a mapping
  - More freedom to position the samples
  - May make the samples linearly separable
  - Run linear SVM in the new space → may be equivalent to non-linear boundaries in the original space



- What mapping to use?

The Kernel Trick

- Revisit the Lagrange Dual Formulation for SVM
  - $L_D(w, \; b, \; \alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} \cdot x^{(j)}$
- Introduce a kernel function

- $L_D(w,\ b,\ \alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} K\left(x^{(i)},\ x^{\{j\}}\right)$

- Mercer's Theorem: for a symmetric, non-negative definite kernel function satisfying some minor conditions, there exists a mapping $\phi(x)$ such that
  - $K\left(x^{(i)},\ x^{\{j\}}\right) = \phi\left(x^{(i)}\right) \cdot \phi\left(x^{(j)}\right)$
- Using a kernel function in L_D can effectively defines an implicit mapping to a higher-dimensional space, where linear SVM was run
- The decision boundaries in the original space can be highly non-linear

Common Kernel Functions

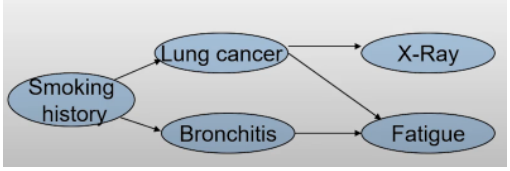
- Polynomials of degree d:
  - $K\left(x^{(i)},\ x^{(j)}\right) = \left[x^{(i)}, x^{(j)}\right]^d$
- Polynomials of degree up to d:
  - $K(x^{(I)},\ x^{(j)}) = \left(\left[x^{\{i)},\ x^{(j)}\right] + 1\right)^d$
- Gaussian Kernels:
  - $K(x^{(I)},\ x^{(j)}) = exp\left(-\frac{||x^{(i)} - x^{(j)}||^2}{2\sigma^2}\right)$
- Sigmoid Kernel:
  - $K(x^{(I)},\ x^{(j)}) = tanh\left(\eta\left[x^{(i)},\ x^{(j)}\right] + \nu\right)$

Knowledge Check

- What is the purpose of the kernel trick in SVMs?
  - Effectively transforming the data onto a higher dimensional feature space
- (In the context of this discussion of non-linearly-separable data) What would happen if you used very small slack variable ξ such that ξ is approximately 0?
  - Misclassification would occur
- Even if we use a kernel trick in SVM, the decision boundary in the original feature space should still be linear since SVM is a linear classifier.
  - False

Why do we use graphical models?

- In machine learning, we are often concerned with join distributions of many random variables
- A graph may provide an intuitive way of representing or visualizing the relationships of the variables
  - Making it easier for domain experts to build a model

- 

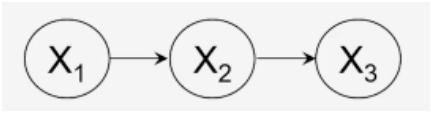
Graphical Models for Casual Relations

- Graphical models arise naturally from, often casual, independency relations of physical events

Bayesian Networks

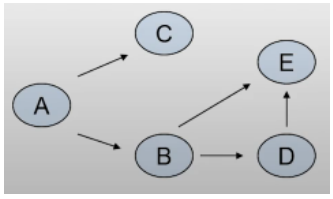
- A BN is directed acyclic graph (DAG), where
  - Nodes (vertices) represent random variables
  - Directed edges represent immediate dependence of nodes
- Other names: Belief networks, Bayes nets, etc.

Conditional Independence

- E.g., given the following graph, check the relationship between X_3 and X_1
  - 

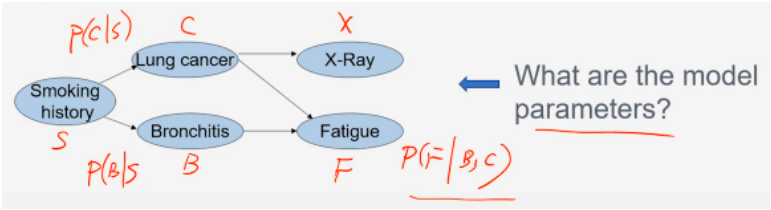
  - X_3 is dependent of X_2, and X_2 is dependent of X_1
  - Thus X_3 is dependent of X_1
  - But given X_2, X_3 is dependent of X_1
- Conditional Independence
  - $P(X_3|X_1, X_2) = P(X_3, X_2)$

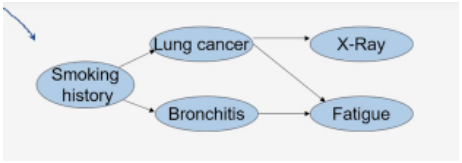
BN For General Conditional Dependency

- A BN can be used to model given conditional dependencies
  - For example, using the chain rule of probability, we have
  - $P(A, B, C, D, E) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)P(E|A, B, C, D)$
- If we know that, given A, C won't rely on B, and so forth, we may have
  - $P(A, B, C, D, E) = P(A)P(B|A)P(C|A)P(D|B)P(E|B, D)$
  - 

    - We could represent joint distributions more compactly in BN → Efficient computation

Inference In Bayesian Networks

- Given a model and some data ("evidence"), how to update our belief?
  - 

    - What are the model parameters?'
    - X = positive, negative
    - F = high, medium, low
    - C = Yes
    - $P(F = high|C = Yes, B = Yes)$
- E.g., for a patient with certain smoking history (non-smoker), whose X-ray result is positive, and who does not experience fatigue:
  - What is the probability of having lung cancer?
  - $P(C = Yes|C = Pos, S = Non)$
- In a simple BN like this, we can compute the exact probabilities
  - 

- In general, for a tree-structured BN, we may use belief propagation for the inference problem
- For general structures, sometimes it is possible to generalize the above method (e.g., the junction tree algorithm). More often, we must resort to approximation methods
  - E.g., Variational methods, Sampling (monte Carlo) methods

Learning in Bayesian Networks

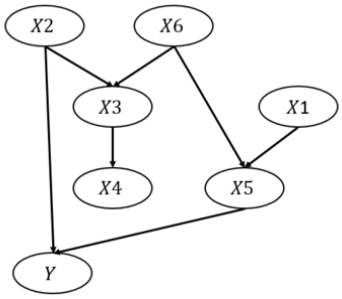
- Learning parameters (probabilities) for given BN (the graph is given)
  - Estimate the (conditional) probabilities from past data
- Learning both the structure and the parameters for a BN
  - A more challenging task beyond the scope of this discussion

Learning the Probabilities

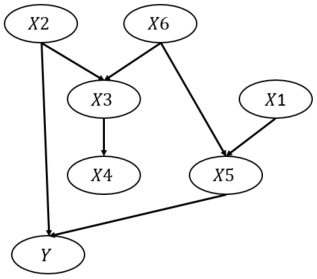
- Basic
  - Use relative frequency for estimating probability
  - A prior distribution is typically assumed
  - The prior is then updated by the data into posterior
  - Using the MLE principle
- The so-called "expectation-Maximization (EM) Algorithm" is often used
  - Iteratively update our guess for the parameter and each step attempts to apply the MLE principle

Knowledge Check

- Which of the following statements is true about graphical models? (Select all that apply.)
    - Graphical models provide an intuitive way for representing or visualizing the relationships of random variables
    - Graphical models arise naturally from, often casual, in-dependency relations of physical events
- Given this graphical model, which statement is true?



  Y depends on $X2$ and $X5$
- Given this graphical model, what is the joint distribution P(Y, X1, X2, X3, X4, X5, X6) ?



    - P(X2)P(X6)P(X1)p(X3|X2,X6)P(X4|X3)p(X5|X6,X1)p(Y,X2,X5)
- For inference with Bayesian network, what are the common techniques we use? (Select all that apply.)
    - Monte Carlo sampling
    - Variational inference
    - Belief propagation

Hidden Markov Models - Formulation

Hidden Markov Models

- Hidden Markov Models (HMMs) are a type of dynamic Bayesian Network
    - Modeling a process indexed by time
- "Hidden": the observations are due to some underlying (hidden) states not directly observable
- "Markov": the state transitions are governed by a Markov process

Discrete Markov Process

- Consider a system which may be described at any time as being in one of a set of N distinct states, S_1, ..., S_N

- At time instances $t = 1, 2, 3$, the system changes its state according to certain probability. The full description requires us to know $P(s^t = S_j | s^{t-1} = S_i, s^{t-2} = S_k, ..., s^1 = S_m)$ for all $t, i, k, ..., m$, where s^t stands for the state of the system at time t
    - For a first-order Markov chain, we need to consider only $P(s^t = S_j | s^{t-1} = S_j)$
    - Further assume Ps are "stationary": $a_{ij} = P(s^t = S_j | s^{t-1} = S_j), 1 \leq i, j \leq N, for\ any\ t$

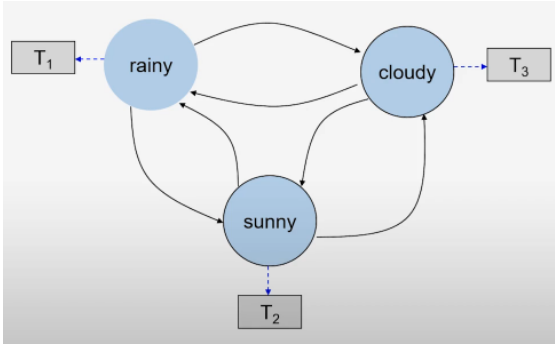
A Simple Example

- Assume one of the three states for each day:
    - S_1 - rainy
    - S_2 - cloudy
    - S_3 - sunny
- Assume the transition probability matrix
    - $A = \{a_{ij}\} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$
- Many questions we may ask, based on this model
    - E.g., Given today is cloudy, what is the probability it remains to be cloudy for the next 5 days

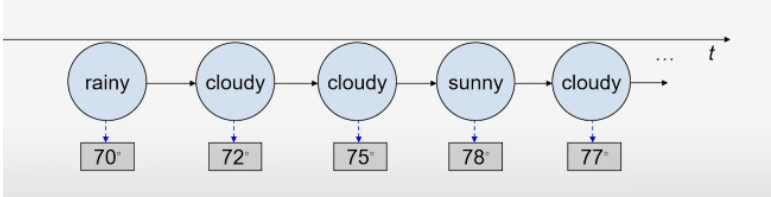
Extending to "Hidden" States

- The previous example is an "observable" Markov model: the output of the system/process is the states of interest
- Now assume that we can only measure the (average) temperature of a day
    - Further assume this measurement is useful for predicting the weather states (rainy, cloudy, sunny)
    - We can view the temperature values as being produced by the *hidden states* of interest, i.e., the weather

A Simple HMM



A Specific Process from the Model



Specifying an HMM

- Θ: the set of hidden states
- The state transition probabilities $a_{ij} = P(s^t = S_j | s^{t-1} = S_j),\ 1 \le i, j \le N$
  - Let $A = \{a_{ij}\}$ be the transition probability matrix
- Ω: the set of outputs (observations)
- The observation probabilities: $P[o^t | s^t]$, where o^t stands for the observation at time t, given the state s^t. This is also called the emission probability
  - For discrete observation space, we can design $B = \{b_{jk}\} = P[o^t = v_k \ at\ t | s^t = S_j)$ as the emission probability matrix, where v_k is the kth symbol in Ω
- The initial state distribution $\pi = \{\pi_i\},\ \pi_i = P(s^1 = S_i)$
  - Sometimes we are given an initial state, i.e., $P(s^1 = S_i) = 1$ for certain i

Basic Problems in HMM

- For a given HMM $\Lambda = \{\Theta, \Omega, A, B, \pi\}$
  - Problem 1: Given an observation (sequence) $O = \{o^1, o^2, ..., o^k\}$, what is the most likely state sequence $S = \{s^1, s^2, ..., s^k\}$ that has produced O?
  - Problem 2: How likely is an observation O (i.e., what is $P(O)$)?
  - Problem 3: How to estimate the model parameters (A, B, π)?

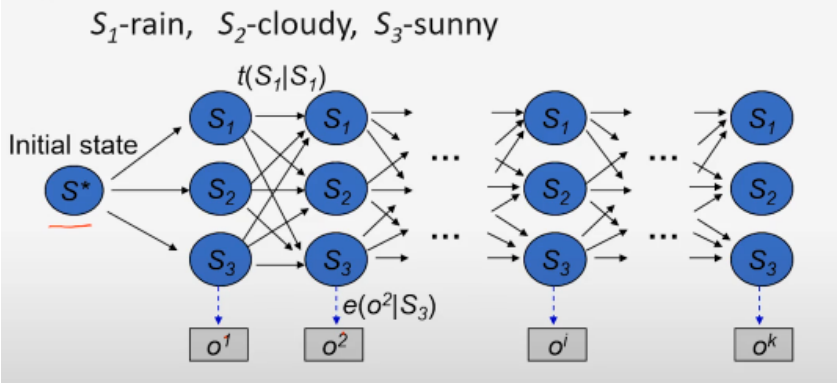
Hidden Markov Models - Learning Inference

Problem 1: State Estimation

- Problem 1: Given an observation (sequence) $O = \{o^1, o^2, ..., o^k\}$, what is the most likely state sequence $S = \{s^1, s^2, ..., s^k\}$ that has produced O?
  - Formally, we need to solve $argmax P(S|O)$
  - Or, equivalently, $argmax \frac{P(S, O)}{P(O)} = argmax P(S,\ O)$
- For a given HMM, we may simplify $P(S, O)$ as
  - $P(S,\ O) = P(O|SP(S)$

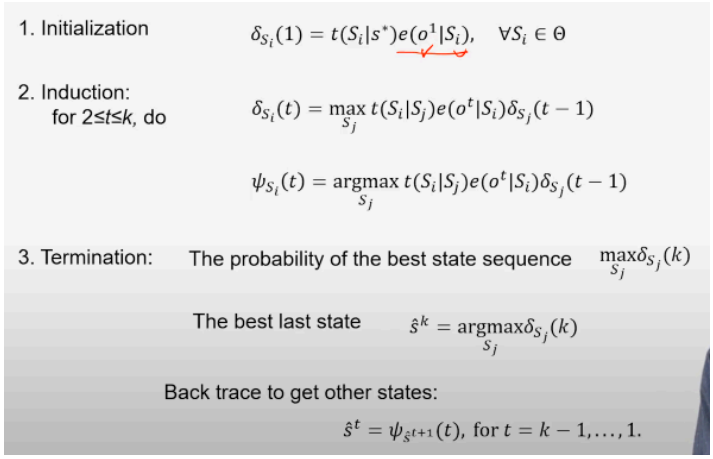
- $$= P(o^1 \dots o^k | s^1 \dots s^k) \prod_{j=1}^{\kappa} P(s^j | s^1 \dots s^{j-1})$$
$$\simeq P(o^1 \dots o^k | s^1 \dots s^k) \prod_{j=1}^{k} P(s^j | s^{j-1})$$
$$= \prod_{i=1}^{k} P(o^i | o^1 \dots o^{i-1}, s^1 \dots s^i) \prod_{j=1}^{k} P(s^j | s^{j-1})$$
$$\simeq \prod_{i=1}^{k} P(o^i | s^i) \prod_{j=1}^{k} P(s^j | s^{j-1}) = \prod_{i=1}^{k} P(o^i | s^i) P(s^i | s^{i-1})$$

The "Weather" Example

- Let's expand the state space as a trellis, for the earlier example:
- 

- –t(.|.) is the transition probability and e(.|.) the emission probability
  - To identify a path for which the product of t's and the e's is maximized

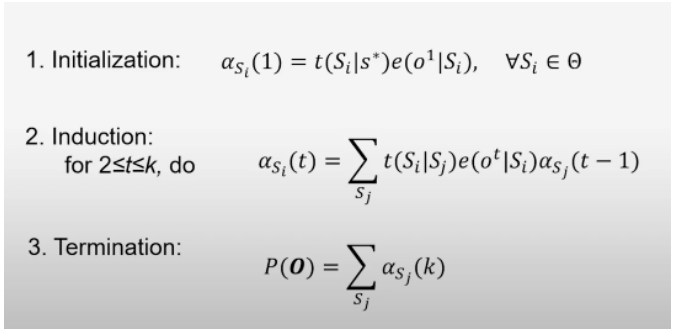
Viterbi Algorithm for Problem 1

- A dynamic programming solution
  - For each state in the trellis, we record:
    - $\delta_{si}(t)$ is the probability of taking the maximal path up to time $t-1$ ending at state S_i at time t and while generating $o^1 \dots o^t$
    - $\psi_{si}(t)$ is the state sequence that resulted in the maximal probability up to state S_i at time t
- 

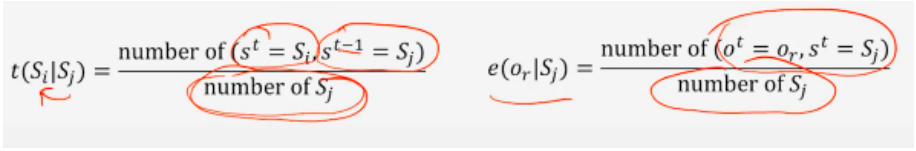
Problem 2: Evaluate P[O]

- To evaluate $P(O)$, we can do $P(O) = \sum_S P(S, O)$
- From the trellis, a solution can be found by summing the probabilities of all paths generating the given observation sequence
- A dynamic programming solution: the forward algorithm or the backward algorithm

The Forward Algorithm

- Define the forward probability $a_{si}(t)$, which is the probability for all paths up to time $t - 1$ ending at state S_i at time t and generating $o^1 \ldots, o^t$
  - 

Problem 3: Parameter Learning

- Case 1: we have a set of labeled data - sequences in which we have the <state, observation> information
  - Use relative frequency for estimating the probabilities
  - the MLE solution:
    - 

- Case 2: we have only the observation sequence
  - The Forward-Backward Algorithm (a.k.a. Baum-Welch Algorithm): An EM approach

Knowledge Check

- For the following data types, which may be appropriately modeled by using HMM? (Select all that apply)
  - Stock market price date
    - Some temporal structure exists
  - Daily precipitation data in Tempe
    - A temporal structure exists.
- Suppose that we use four distinct words to write a paragraph with 100 segments, and we treat each word in the paragraph as a segment. We want to infer by HMM three possible class labels of all the segments in this

paragraph, including a) location b) person name, and c) background. What is the size of the state transition probability matrix in our HMM model?

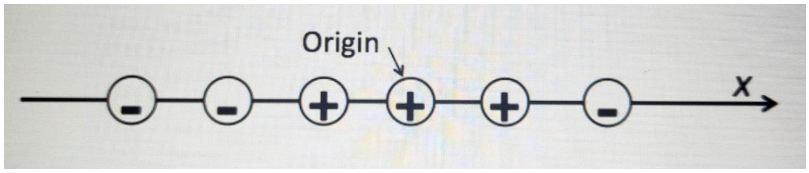
- 3*3
  - Null

- Suppose that we use four distinct words to write a paragraph with 100 segments, and we treat each word in the paragraph as a segment. We want to infer by HMM three possible class labels of all the segments in this paragraph, including a) location b) person name, and c) background. If we start randomly with three possible class labels, what is the π?
  - $\pi 1 = 1/3, \pi 2 = 1/3, \pi 3 = 1/3$
- Suppose that we use four distinct words to write a paragraph with 100 segments, and we treat each word in the paragraph as a segment. We want to infer by HMM three possible class labels of all the segments in this paragraph, including a) location b) person name, and c) background. What is the size of the observation probability matrix?
  - 3*4
- Suppose that we use four distinct words to write a paragraph with 100 segments, and we treat each word in the paragraph as a segment. We want to infer by HMM three possible class labels of all the segments in this paragraph, including a) location b) person name, and c) background. With a given paragraph, how many observations do you see?
  - 100
- Suppose that we use four distinct words to write a paragraph with 100 segments, and we treat each word in the paragraph as a segment. We want to infer by HMM three possible class labels of all the segments in this paragraph, including a) location b) person name, and c) background. What is the length of the path of the state ?
  - 100
- Suppose that we use four distinct words to write a paragraph with 100 segments, and we treat each word in the paragraph as a segment. We want to infer by HMM three possible class labels of all the segments in this paragraph, including a) location b) person name, and c) background. Suppose that the first state is about 'background', how many different possible state paths are there in total?
  - 3^{99}

Module 4 Quiz Questions

- Which of the following is true for SVMs? (Select all that apply)
  - The support vectors alone define the decision boundary in SVM
  - For linearly separable data for two classes, there's only one hyperplane that can correctly classify all samples
  - SVMs use the kernel trick to effectively transform the problem into a space where the data is linearly separable
  - Selected 1 and 3

- Given the following dataset in the 1-d space, which consists of 3 positive data points {-1, 0, 1} and 3 negative data points {3, -2, -2}

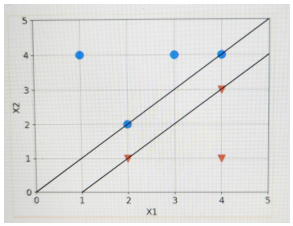


Are they linearly separable in the original 1-d space?

  - No

- Consider the figure provided.

(A plot of 7 points belonging to two classes, are being separated by two canonical hyperplasia. The points of Class blue are at (1,4), (2,2), (3,4), (4,4) and the points of Class red are (2,1), (4,1), (4,3). The First hyperplasia is passing through (2,2) and (4,4) and the second hyperplasia is passing through (2,1) and (4,3).



What is the classification rule of the maximal margin classifier that classifies the points in blue (the circles)?

  - $1-x_1+x_2>0$

- Which of the following dataset sets require a kernel transformation to transform data into higher dimensions where it can be separated with a hyperplane?

Three figures: Figure 1 has points of two classes completely mixed and there is no separation. Figure 2: half points of each class are mixed and the other half are separated. Figure 3: the two classes are clearly separated. (Select all that apply)

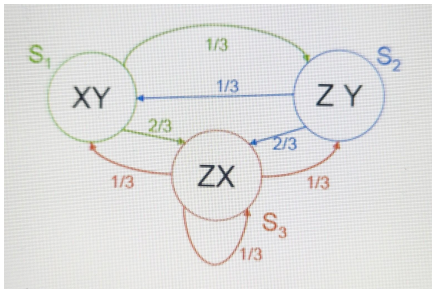
  - Figure 1
  - Figure 2
  - Figure 3
  - Selected Figure 1 and 2

- In the following application the SVM classifier is applied to a non-linearly separable data. For which point is xi(Greek alphabet letter) ≠0

A plot of 9 points, 5 of Class blue and 4 of Class three. The blue points are at (1,4), (2,4), (3,4), (2,3), and (2.5, 2.5). The red points are at (1.5, 2), (2,1), (4,1), and (4,3). A dotted line is passing through (2,3) and (3,4). A solid line is passing through original and (2.5, 2.5) and a dotted line is passing through (2,1) and (4,3)

  - Point B

- Given the following HMM model where symbols X, Y, and Z represent the possible observations in the states. In a circle representing a state, the symbols are equally possible observations in that state.



There are 3 nodes XY, ZY, ZX with the following configuration: XY is in a state S1 and has a directed edge from XY to ZY with weight 1/3; there is another directed edge from XY to ZX with weight 2/3; ZY is in a state S2 and has a directed edge from ZY to ZX with weight 2/3; there is a directed edge from ZY to XY with weight 1/3; ZX is in state S3 and there is a self loop at ZX with weight 1/3; there is a directed edge from ZX to ZY with weight 1/3; there is a directed edge from ZX to XY with weight 1/3

What is the value of a_33 in the state transition probability matrix?

- 1/3

- Referencing again to the HMM model: Suppose pi_2=1/2, what is the value of P(Q), Q= S_2×S_3×S_3 ?
  - 1/9
- Referencing again to the HMM model: How much is P(O1=Y, O2=X, O3=Z|S2S3S3)?
  - 1/8

Set-Up of the Unsupervised Learning Problem

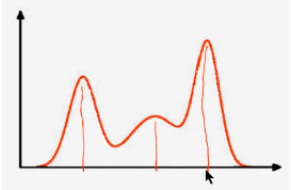
Learning from Unlabeled Data

- Given a training set of n unlabeled samples $\{x^{(i)}\}$
- What can we learn from the samples?
  - We could estimate the overall distribution of the data without knowing their label
  - We could figure out the groupings of the samples (if any)

An Example

- Illustrating structures/groupings of unlabeled samples may relate to the (unknown) labels of the samples
  - If we know the labels, we may find the densities of the classes
  - What may we see if we have no label for the sata samples?

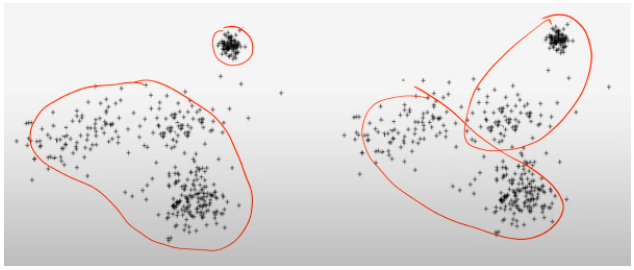
Another Example:

- A density estimated from unlabeled samples may help us to identify densities of different classes
- If we know there are three classes in the data, each having a normal distribution
  - 

A Mixture-Density Model

- Assume a parametric model like this:
  - The samples come from C classes
  - The prior probabilities $P(\omega_j)$ for each class are known for, $j = 1, ..., C$
  - The form of $p(x|\omega_j, \theta_j)$ $(j = 1, ..., C)$ are known
  - The C parameter vectors $\theta_1, \theta_2, ..., \theta_C$ are unknown
- Samples from this distribution are given, but the labels of the training samples are unknown.
- What is the PDF of the unlabeled samples?
  - $p(x|\theta) = \sum_{j=1}^{C} p(x|\omega_j, \theta_j)P(\omega_j)$ where $\theta = (\theta_1, \theta_2, ..., \theta_C)$
- Can we learning θ from unlabeled samples from this mixture density?

Illustrating Mixture-Density Model

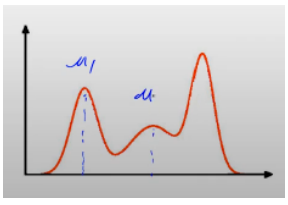
- An example: with the assumption of 4 classes
- 

The Question of Identifiability

- Can we learning a unique θ from a set of unlabeled samples from a mixture density?
  - For continuous features (with PDFs), the answer is often "Yes"
- An example in discrete case (with PMF)
  - Two coins with P(head) being p & q respectively
  - Randomly pick one and toss it; Record the outcome
  - With only the outcomes of N tosses, but not knowing which coin was used each time (→ unsupervised), can we figure out p and q?

The Gaussian Mixture Models

- The mixture model: $p(x|\theta) = \sum_{j=1}^{C} p(x|\omega_j, \theta_j)P(\omega_j)$
- GMM: each component density is a Gaussian distribution
  - Can be a good approximation to many real data distributions

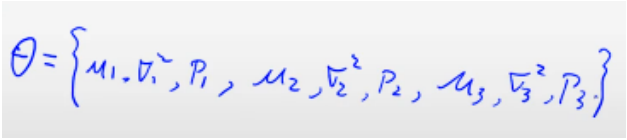


If we do have labels...

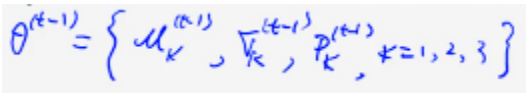
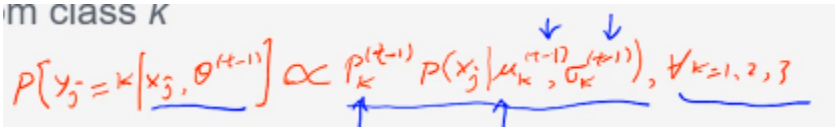
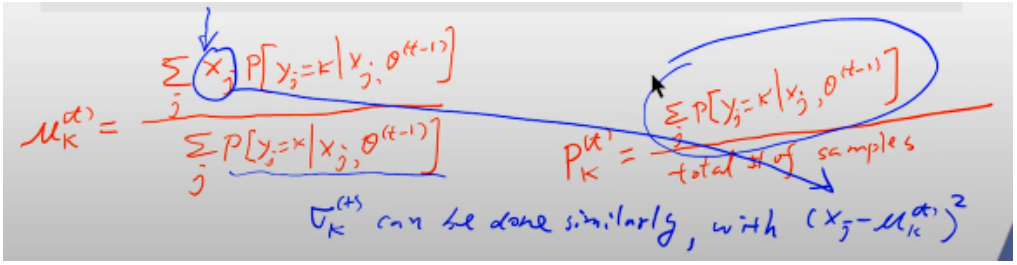
$$p(x|\theta) = \sum_{j=1}^{C} p(x|\omega_j, \theta_j)P(\omega_j)$$

- This becomes supervised learning for each component (class)
  - $\mu_1 = \frac{1}{n_1}\Sigma x_i^{(1)}$, $\sigma_1^2 = \frac{1}{n_i}\Sigma\left(x_i^{(1)} - \mu_1\right)^2$
- It is more difficult without labels

Unsupervised Case

- Consider an iterative method using the maximum likelihood estimation concept
- Consider a 3-component 1-d example
- What are the parameters in this case?
  - 
- We might have some initial (imprecise) guesses for the parameter, e.g., vs the k-means algorithms

An Example of Expectation-Maximization Algorithm:

- Iterate on t
  - Given parameter estimates at iteration $t - 1$
    
  - Step 1: for a sample j, compute its probability of being from class k
    
  - Step 2: Update the estimates of the parameters
    

Knowledge Check

- Given a dataset of unlabeled samples, what can we learn from the samples? (Select all that apply.)
  - Identify features that may be more important than others
    - E.g., a feature that stay the same more or less the same for all samples may be not important at all.
  - Figure out potential groupings of the samples

    - Estimate the overall distribution of the data
- Which of the following are the roles of clustering? (Select all that apply)
    - Finding coherent groups of data
    - Visualizing big data
    - Figuring out structures of the feature space
- Gaussian Mixture Model is a good approximate of many real data distributions: True or False?
    - True
- To learn a GMM, the number of components should be given: True or False?
    - True
- GMM always assign hard class membership: True or False?
    - False
- Using EM algorithm to learn a GMM can always guarantee a global optimal solution: True or False:
    - False

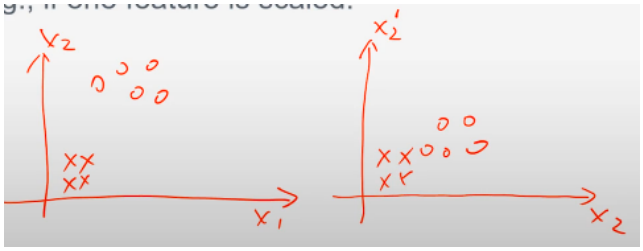
Unsupervised Learning - Part 3: The k-Means Algorithm

Finding the clusters/grouping of the samples

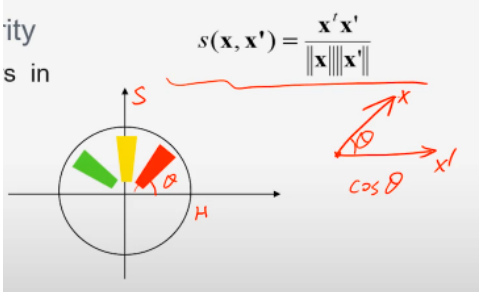
- A few basic questions to answer
    - How to represent the clusters?
        - We will use the centroid to represent a cluster
    - Which cluster a sample should be assigned to (e.g., membership)?
        - We will use the similarity to the centroid to determine the membership
    - What similarity measure to use?
        - E.g., Euclidean distance

More on Similarity Measures

- If we use Euclidean distance as the measure:
    - It is invariant to translations & rotations of the feature space
    - But not to more general transformations
- E.g., if one feature is scaled



- Other types of similarity measures
- E.g., cosine similarity
    - For clustering colors in the gue-saturation space
- E.g., distance on a graph, like shortest path



Clustering as Optimization

- The sum-of-squared-error criterion/cost
  - Let D_i be the subset of samples from class i
  - Let n_i ne the number of samples in D_i, and m_i the mean of those samples
    - $m_i = \frac{1}{n_i} \sum_{x \epsilon D_i} x$
  - The sum of squared error is: $J_e = \sum_{i=1}^{C} \sum_{x=D_i} ||x - m_i||^2$
- Well-separated, compact data "clouds" tend to give small errors when the clusters coincide with the clouds
- An optimization problem to solve for finding a "good" clustering: to find the partition of the data that minimizes J_e
- If the membership of a sample is determined by the distance to the means m_i
  - Then the task is to find the optimal set of $\{m_i\}$
  - The problem is NP-hard

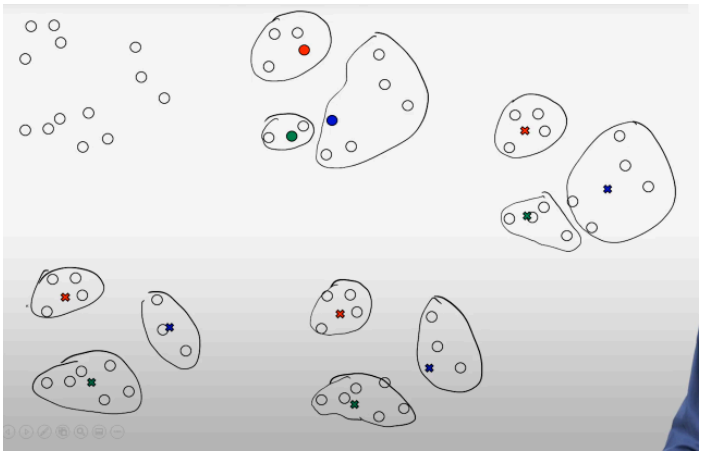
k-Means Clustering

- Input: Given n data samples
- Goal: Partition them into k clusters/sets D_i, with respective center/mean vectors $\mu_1, \mu_2, ..., \mu_k$, so as to minimize
  - $\sum_{i=1}^{k} \sum_{x \epsilon D_i} ||x - \mu_1||^2$
- Comparing with the mixture models:
  - Here we do "hard" assignment of the membership to a samples (simply based on its distance to the cluster center)

The Basic k-Means Algorithm

```
Given: n samples, a number k.
Begin
   initialize μ1, μ2, …, μk (randomly selected)
      do classify n samples according to nearest μi
         recompute μi
      until no change in μi
   return μ1, μ2, …, μk
End
```

Illustrating the Algorithm

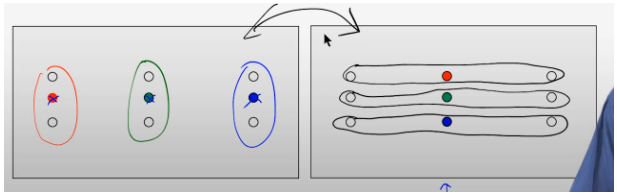


Knowledge Check

- The K-means algorithm can always converge with finite steps: True or False?
  - True
- What can the K-means algorithm be used for?
  - Clustering
- Which of the following statements is true about K-means? (Select all that apply.)
  - K-means assigns hard class membership
  - The choice of initial points may have a large influence on the results
  - K-means finds a local optimum
- Which of the following contributes to achieving good clustering? (Select all that apply.)
  - Minimizing the sum of distances within clusters
  - Using a good similarity measure meaningful for a given problem

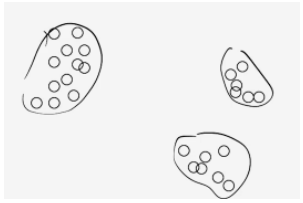
Unsupervised Learning Part 4

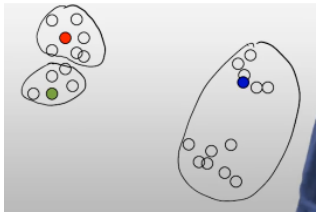
Properties of the k-Means Algorithm

- The algorithm will converge when the cluster centers no longer change
  - Sensitivity to initialization
- But the results may not be an optimal solution
  - 

Another Example

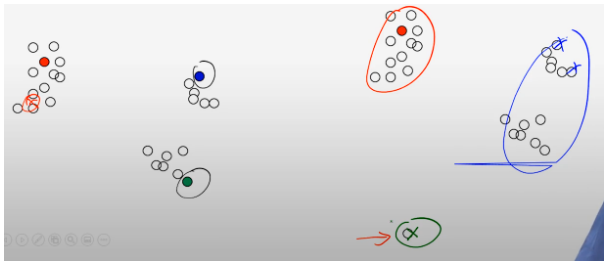
- The natural grouping seems to be so well defined

- 

- For $k = 3$, what will be the clusters?
  - 
- What can we do to improve?

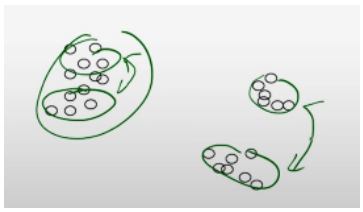
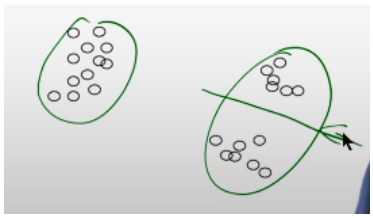
A Few Common "Tricks"

- Multiple runs with different initial centers
- Choosing the point furthest from the previous centers
  - Drawback: might be sensitive to "outliers"



Other Variants of Basic k-Means

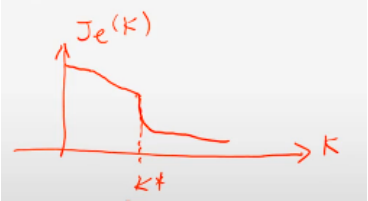
- k-Means++:
  - New centers are chosen with probabilities (as a function of distance to closest prior centers)
  - Kind of between "random prior centers)
  - Kind of between "random" and "furthest point" techniques
- Hierarchical approaches
  - Agglomerative vs. divisive

The QUestion of Choosing k

- Two trivial extremes
  - If $k = 1$, the error is the variance of the samples

    - If $k = n$, the error can become 0
- What is a proper $1 < k < n$ for capturing the structure of the samples?
- Some tricks
    - Trick 1: Will the cost function drop dramatically at some point?
    - Trick 2: Cross-validation (on, e.g., a classification task)



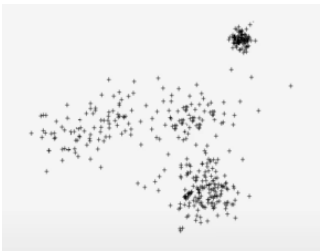
Knowledge Check

- Select the answer that correctly completes the following statement.
- The goal for K-means cost function is to ___ squared error function where error function represents distance between data points and cluster centroids.
    - minimize
- Which of the following strategies may improve the performance of K-means? (Select all that apply.)
    - Choose the point furthest from the previous centers
    - Multiple runs
- Which of the following statements is true about GMM and K-Means?
    - Both methods are unsupervised approaches

Spectral Clustering: Introduction

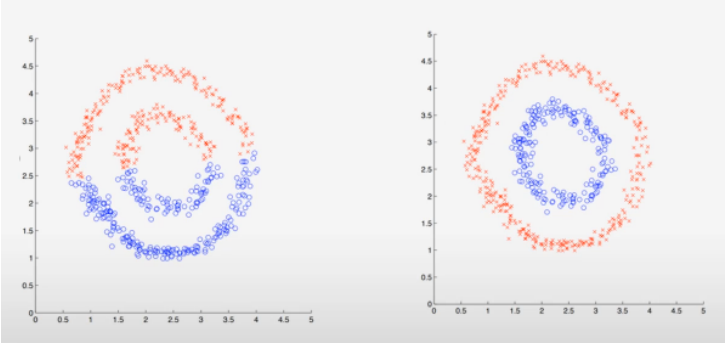
Revisiting k-means & mixture models

- K-means use "hard" membership while mixture models allow "soft" membership
- Both use feature/vector representation of the data as input → E.g., Euclidean distance is one neutral (dis)similarity measure
    - What is the input data is NOT represented in feature/vector, format?
- E.g., graph data
- E.g., object with only pair-wise similarities (like individuals on a social network → community detection)
- In both k-means and mixture models, we look for compact clustering structures



- In some cases, connected-component structures may be more desirable

Example

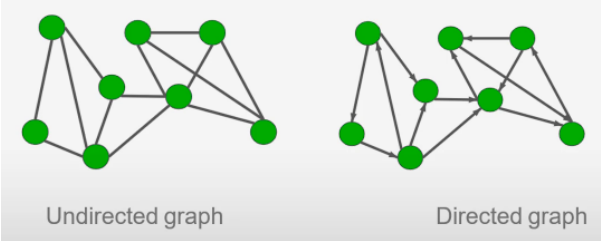


Spectral Clustering

- A family of methods for finding such similarity-based clusters
  - "Spectral": for using the eigenvalues (spectrum) of the similarity matrix of the data
  - Graph clustering, similarity-based clustering
- The objects to be clustered are not in a vector space
  - The primary feature is the similarity between objects
  - For any pair of objects i and j, we have a value $s(i,\ j)$ measuring their similarity; all such values form the similarity matrix
- Graphs are intuitive for representing/visualizing such data

Graph Representation

- Definition: A graph $G = (V,\ E)$ is defined by V, a set of N vertices, and E, a set of edges



- In spectral clustering, we consider undirected graphs

Graph Representation (¼)

- Adjacency matrix W of undirected graph
  - $N \times N$ symmetric binary matrix
  - The row and columns are indexed by the vertices and the entries represent the edges of the graph
  - $\omega_{i,j} = 0\ if\ vertices\ i,\ j\ are\ not\ connected$
  - $\omega_{i,j} = 1\ if\ vertices\ i,\ j\ are\ connected$
- Simple graph = zero diagonal

Graph Representation (2/4)

- Weighted adjacency matrix (sometimes called affinity matrix)

    - Allow values other than 0 or 1
    - Each edge is weighted by pairwise similarity
    - $\omega_{i,j} = 0 \; if \; i, j \; are \; not \; connected$
    - $\omega_{i,j} = s(i, \; j) \; if \; i, j \; are \; connected$
- $\omega_{i,j}$ may be defined through some kernel functions

Graph Representation (¾)

- Degree matrix D of undirected graph
- $N \times N$ diagonal matrix that contains information about the degree of each vertex
- Degree $d(v_i)$of a vertex v_i: number of edges incident ot the vertex
    - Extended to sum of weights from edges incident to the vertex
- So, we have:

$$\mathbf{D} = \begin{bmatrix} d(v_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d(v_N) \end{bmatrix}$$

Graph Representation (4/4)

- Laplacian matrix L of undirected graph
- $L = D - W$ (Degree -Affinity) (Unnormalized)
- L is symmetric and positive semi-definite
- N non-negative real-values eigenvalues
- The smallest eigen-value is 0, the corresponding eigenvector is the 1-vector (all elements being 1)
- The smallest non-zero eigenvalue of L is called the spectral gap

Knowledge Check

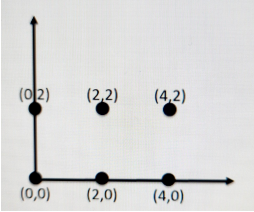
- Given an adjacency matrix W, if $\omega_{i,j} = 1$, what does that mean?
    - Node i and Node j are connected
- What is the spectral gap?
    - The smallest non-zero eigenvalue of L
- Which of the following statement is true? (Select all that apply.)
    - Laplacian matrix of undirected graph is a positive semi-definite matrix
    - Adjacency matrix of undirected graph is a symmetric matrix

Module 5 Quiz Questions

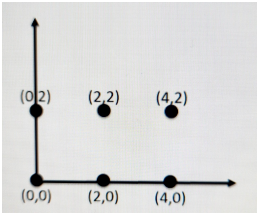
- Which of the following defines hard clustering?
    - Every data points falls into one cluster and one only
- Which of the following can NOT be used as a stopping criteria for K-means algorithm? (Select all that apply):
    - The sum of squared error reaches a local maximum

    - There are points that belong to more than one cluster
- Which of the following should be taken into consideration when using the K-means algorithm to cluster a new dataset? (Select all that apply):
    - Selection of the initial cluster centers
    - Existence of the outliers (so different initialization strategy may be used)
    - Selection of the number of clusters
- Which of the following is not a step in K-means algorithm implementation?
    - Use the same cluster center for each iteration until a stopping criterion is met
- Which of the following is correct for the K-means algorithm? (Select all that apply):
    - On every iteration of K-means, the cost function should either stay the same or decrease (in particular, it should not increase)
    - The value of K is selected manually
- You are given 5 data points in a 1-D space: x_1=-5, x_2=-4, x_3=0, x_4=2, and x_5=3. Suppose K=2 and the initial cluster centers are μ_1=0, and μ_2=1. When running K-means, what is the cluster assignment for each data point after convergence?
    - μ_1: x_1, x_2, x_3

      μ_2: x_4, x_5
- Which of the following is true for K-Means Clustering (Select all that apply):
    - There is no guarantee that it reaches the global optimum
    - Completely different clusters may arise from small changes in the initial random choice
    - The overall goal is to minimize the total squared distances from all points of their cluster centers
- Suppose we find K-Means on the following dataset with six data points to find two clusters.

  (There are a set of points in a 2 dimensional space with the following coordinates: point P1 has coordinates (0,0), points P2 has coordinates (0,2), points P3 has coordinates (2,0), points P4 has coordinates (2,2), points P5 has coordinates (4,0), and points P6 has coordinates (4,2).)
  Suppose the initial cluster centers are (0,0) and (5,0). How many iterations does the algorithm take until convergence?
    - Fewer than 3 iterations
- Suppose we run K-means on the following dataset with six data points to find two clusters. If the initial cluster centers are (2,0) and (2,2), what is the cluster assignment for each data point after Step1?



(There are a set of points in a 2 dimensional space with the following coordinates: point P1 has coordinates (0,0), points P2 has coordinates (0,2), points P3 has coordinates (2,0), points P4 has coordinates (2,2), points P5 has coordinates (4,0), and points P6 has coordinates (4,2).)

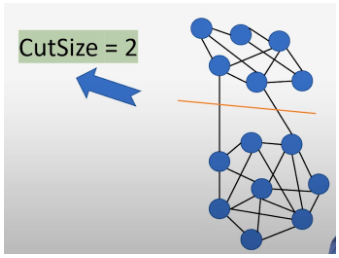
- bottom three belong to the first cluster, and the upper three belong to the second cluster

- K-Means algorithm can be used to solve _ problems (fill in the blank)
  - Clustering

A Graph Cut Formulation

Clustering as Graph Partition/Cut

- Find a partition of a graph such that the edges between different groups have a very low weight while the edges within a group have high weight
- E.g., minimum cut
- More general, consider weighted edges



2-way Spectral Graph Partitioning

- Weighted adjacency matrix W
  - $w_{i,j}$: the weight between two vertices i and j
- (Cluster) Membership vector q
  - $q_i = \{1 \;\; i \in Cluster\ A, \;\; -1 \;\; i \in Cluster\ B$
  - $q = argmin\ Cutsize,\ q \in [-1,\ 1]^n$
  - $Cutsize = J = \frac{1}{4}\sum_{i,j}(q_i - q_j)^2 w_{i,j}$

Solving the Optimization Problem

$$q = argmin \frac{1}{4}\sum_{i,j}\left(q_i - q_j\right)^2 w_{i,j}$$

- Directly solving the above problem requires combinational search → exponential complexity

- How to reduce the computational complexity?

Relaxation Approach

- Key difficulty: q_i has to either $-1, 1$
  - Relax q_i to be any real number
  - Impose Constraint: $\sum_{i=1}^{n} q_i^2 = n$
    - $J = \frac{1}{4}\sum_{i,j}\left(q_i - q_j\right)^2 w_{i,j} = \frac{1}{4}\sum_{i,j}\left(q_i^2 - 2q_iq_j + q_j^2\right)w_{i,j}$
    - $= \frac{1}{4}\sum_{i} 2q_i^2\left(\sum_j w_{i,j}\right) - \frac{1}{4}\sum_{i,j} 2q_iq_jw_{i,j}$
    - $= \frac{1}{2}\sum_{i} q_i^2 d_i - \frac{1}{2}\sum_{i,j} q_i\left(d_i\delta_{i,j} - w_{i,j}\right)q_j$
    - where $d_i = \sum_j w_{i,j}$ and $D \equiv \left[d_i\delta_{i,j}\right]$
    - $\Rightarrow j = \frac{1}{2}q^T(D - W)q$
- The final problem formulation:
  - $q = argmin\, J = argmin\, q^T(D - W)q$
  - $$\mathbf{q} = \underset{\mathbf{q}}{\operatorname{argmin}}\ J = \underset{\mathbf{q}}{\operatorname{argmin}}\, \mathbf{q}^T(\mathbf{D} - \mathbf{W})\mathbf{q}$$
  - subject to $\sum_{i=1}^{n} q_i^2 = n$
- Solution: the second minimum eigenvector for $D - W$
  - $(D - W)q = \lambda_2 q$

Graph Laplacian

- $L = D - W$
- L is semi-positive definitieve matrix
  - For any x, we have $x^T Lx \geq 0$. (Why?)
- Minimum eigenvalue $\lambda_1 = 0$ (what is the eigenvector?)
  - $0 = \lambda_1 < \lambda_2 < \lambda_3 \ldots < \lambda_k$
- The eigenvector that corresponds to the second minimum eigenvalue λ_2 gives the best bipartite graph partition

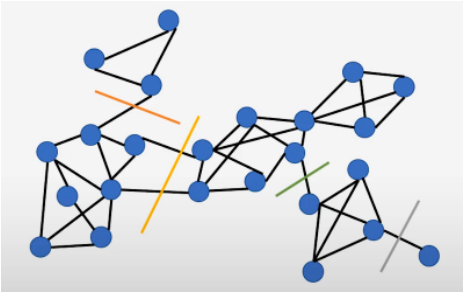
Recovering the Partitions

- Due to the relaxation, q can be any number _not just -1 and 1)

- How to construct the partition based on the eigenvector?
- A simple strategy:
  - $A = \{i | q_i < 0\}, B = \{i | q_i \geq 0\}$

One Obvious Drawback

- Minimum cut does not balance the size of bipartite graphs



- How should we consider other factors like the sizes of the partitions?

Knowledge Check

- If the dataset has only pair-wise similarity, and we want to cluster on it, which approach should we choose?
  - MinCut
- In MinCut algorithm, we only need to consider the CutSize: True or False?
  - True
- To obtain 2-way partition, we should pick the eigenvector that is corresponding to which eigenvalue?
  - The second smallest

Beyond MinCut

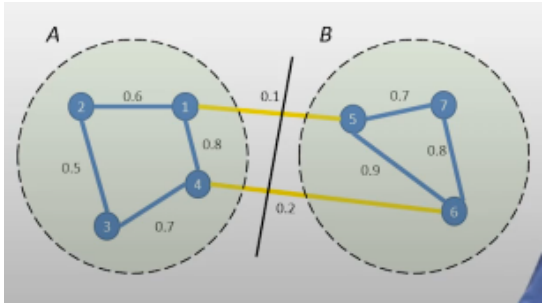
MinCut

- In MinCut, we used the following objective function:
  - $J_{MinCut} = Cut(A, B)$
- We noted one drawback of MinCut: the sizes of the partitions are not considered
- A few extensions exist

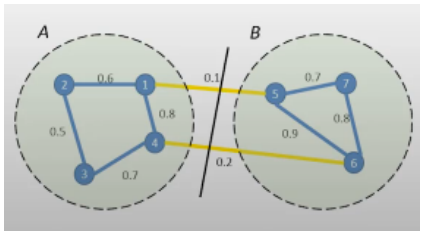
Characterizing Graph Cut

- $Cut(A, B) = \Sigma_{i \epsilon A, j \epsilon B} w_{i,j}$ $e.g.,$ $Cut(A, B) = 0.3$
- $Cut(A, A) = \Sigma_{i \epsilon A, j \epsilon A} w_{i,j}$ $e.g.,$ $Cut(A, A) = 2.6$
- $Cut(B, B) = \Sigma_{i \epsilon B, j \epsilon B} w_{i,j}$ $e.g.,$ $Cut(B, B) = 2.4$
- $Vol(A) = \Sigma_{i \epsilon A} \sum_{i=1}^{n} w_{i,j}$ $e.g.,$ $Vol(A) = 5.5$
- $Vol(B) = \Sigma_{i \epsilon B} \sum_{i=1}^{n} w_{i,j}$ $e.g.,$ $Vol(B) = 5.1$

- $|A| = 4,\ |B| = 3$



The Ratio Cut Method

- The Objective function:
  - $J_{RatioCut}(A, B) = Cut(A, B)\left(\frac{1}{|A|} + \frac{1}{|B|}\right)$
- Attempts to produce balanced clusters
  - Example: $J_{RatioCut}(A, B) = \frac{7}{40}$
  - 

- Similar to MinCut, the solution can be found by the following generalized eigenvalue problem:
  - $(D - W)q = \lambda Dq$
  - $Lq = \lambda Dq$

Normalized Cut (NCut)

- In Ratio Cut, the balance of the partitions is defined based on the number of vertices
- We may consider the "size" of a set based on weights of its edges → Ncut
- The objective function is:
  - $J_{NCut}(A, B) = Cut(A, B)\left(\frac{1}{Vol(A)} + \frac{1}{Vol(B)}\right)$
- Example: $J_{NCut}(A, B) = 0.1134$

Additional Considerations

- In clustering, we should also consider within-cluster connections
- A good partition should consider
  - Inter-cluster connections, and
  - Intra-cluster connections

MinMaxCut

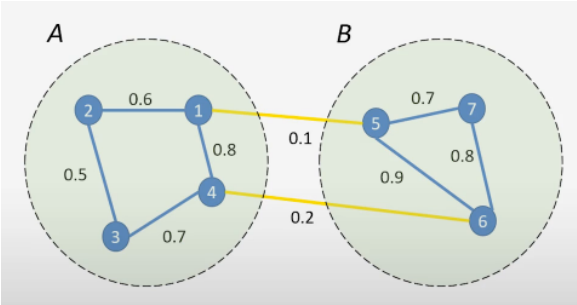
- 1st constraint: inter-connection should be minimized: $MinCut(A, B)$

- 2nd constraint: intra-connection should be maximized: $MaxCut(A, A)\ and\ MaxCut(B, B)$
- These requirements may be simultaneously satisfied by minimizing the objective function:
  - $J_{MinMaxCut}(A, B) = Cut(A, B)\left(\frac{1}{Cut(A,A)} + \frac{1}{Cut(B,B)}\right)$
- Example: $J_{MinMaxCut}(A, B) = 0.240$

Normalized and MinMaxCut methods

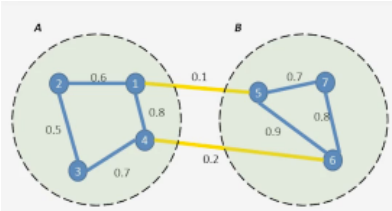
- Similar to before, we may relax the indicator vector q to real values
- For both NCut and MinMaxCut, the solution may be found by solving generalized eigenvalue problems

An Illustrative Example



- For the following graph, what is Cut(A, B)?
  - 0.3
    - 0.1+0.2=0.3

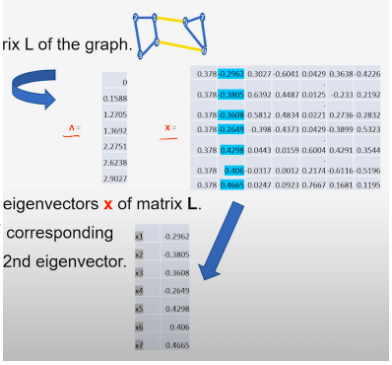
Graph and Similarity Matrix



	x1	x2	x3	x4	x5	x6	x7
x1	1.5	-0.6	0	-0.8	-0.1	0	0
x2	-0.6	1.1	-0.5	0	0	0	0
x3	0	-0.5	1.2	-0.7	0	0	0
x4	-0.8	0	-0.7	1.7	0	-0.2	0
x5	-0.1	0	0	0	1.7	-0.9	-0.7
x6	0	0	0	-0.2	-0.9	1.9	-0.8
x7	0	0	0	0	-0.7	-0.8	1.5

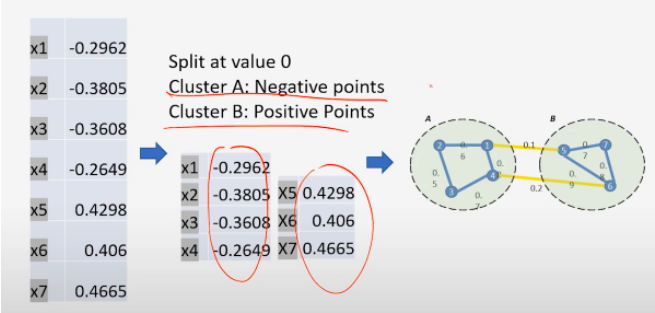
Solve Eigen Problem

- Pre-processing
  - Build Laplacian matrix L of the graph
- Find
  - Eigenvalues Λ and eigenvectors x of matrix L
  - Map vertices to the corresponding components of the 2nd eigenvector



Spectral Clustering

- Splot at value 0
- Cluster A: Negative points
- Cluster B: Positive Points



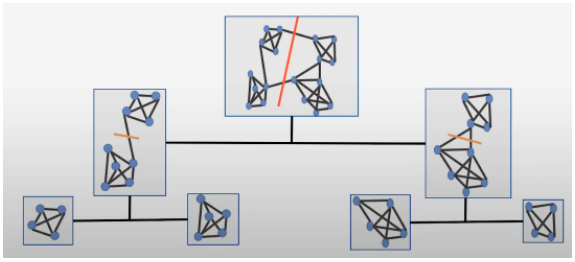
Knowledge Check

- Which of the following methods considers the "size" (by any measure) of the partitions? Select all that apply.
  - RatioCut
  - NCut
  - MinMaxCut
- Which of the following considers the size of a set based on weights of its edges?
  - NCut
- Spectral clustering does not make strong assumptions on the forms of the cluster: True or False
  - True

Practical Considerations in Implementation

Recursive bi-partitioning

- Recursively apply bi-partitioning algorithm in a hierarchical divisive manner
- Disadvantages: inefficient, stability issues



K-way Graph Cuts

- Generalizing the 2-way objective functions:

- $J_{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A_i})}{|A_i|}$
- $J_{NCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A_i})}{Vol(A_i)}$
- $J_{MinMaxCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A_i})}{Cut(A_i, \overline{A_i})}$

Implementation Considerations (1/4)

- Preprocessing: spectral clustering method can be interpreted as tools for analysis of the block structure of the similarity matrix
    - Building such matrices may certainly ameliorate the results
- When building graphs from real data
    - Calculation of the similarity matrix is not evident
    - Choosing the similarity function can highly affect the results of the following steps
    - A Gaussian kernel is often chosen, but other similarities like cosine similarity might be proper for specific applications

Implementation Considerations (2/4)

- Graph and similarity matrix constructionL Laplacian matrices are generally chosen to be positive and semi-definite thus their eigenvalues will be non-negatives
    - A few variants

Unnormalized	$L = D - W$
Symmetric	$L_{sy} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$
Asymmetric	$L_{As} = D^{-1} L = I - D^{-1} W$

Implementation Considerations (3/4)

- Computing the eigenvectors
    - Efficient method exist for sparse matrices
- Different ways of building the similarity graphs
    - ε-neighborhood graph
    - k-nearest neighbor graph
    - fully connected graph

Implementation Considerations (4/4)

- Choosing k:
    - Similar to k-means, there are many heuristics to use
    - The eigengap heuristic: to choose a k such that first k eigenvalues are very small but the $(k + 1)th$ one is relatively large

- Clustering: simple algorithms other than k-means can be used in the last stage, such as simple linkage, *k*-lines, elongated k-means, mixture model, etc.

Recap: Pros and Cons of Spectral Clustering

- Advantages:
    - Does not make strong assumptions on the forms of the clusters
    - Easy to implement, and can be implemented efficiently even for large data sets as long as the similarity graph is sparse
    - Good clustering results
    - Reasonably fast for sparse data sets of several thousand elements
- Disadvantages:
    - May be sensitive to choice of parameters for neighborhood graph
    - Computationally expensive for large datasets

Knowledge Check

- What can you do to obtain K-way partition? (Select all that apply.)
    - Generalizing the 2-way objective functions to consider k clusters directly
        - We can also apply 2-way partition recursively.
    - Apply 2-way partition recursively
        - We can also formulate an objective function to reflect k-way partition directly as shown in the slides.
- How can we build the similarity graphs?
    - Consider l-nearest neighbor
    - Consider ε-neighborhood
    - Consider all nodes in the graph
- Which statement is true about spectral clustering? Select all that apply.
    - Spectral clustering may be sensitive to the choice of parameters for neighborhood graph
    - Spectral clustering does not make strong assumptions on the forms of the clusters

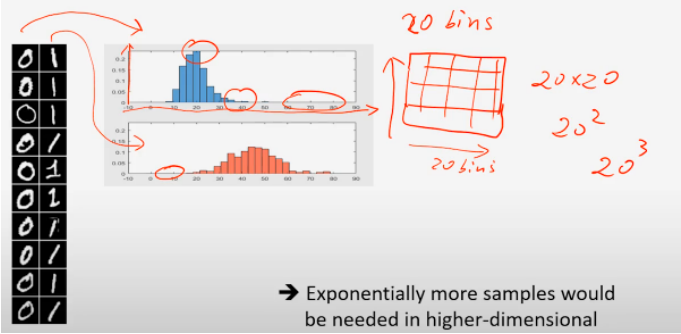
Introduction to the Problem of Dimensionality Reduction

What is Dimensionality Reduction?

- We have *N* data points in a high-dimensional space
    - e.g., in the order of tens of thousands of dimensions
- We want to project them into some low-dimensional space
    - e.g., in the order of tens of dimensions
- Why dimensionality reduction?
    - a key technique to mitigate curse of dimensionality

The Curse of Dimensionality

- Consider histogram as density estimator



- Exponentially more samples would be needed in higher dimensional spaces for the same "resolution"

Many Techniques for Dimensionality Reduction

- Many ways for going from a higher-dimensional space to a lower-dimensional space
    - Feature Selection achieves this by keeping only a subset of the original features/dimension
- There are many other techniques, employing a feature mapping/projection approach
    - New features are generated (instead of selecting only from the original features)
    - The underlying assumptions and/or goals of the techniques are often different

Examples of Feature Mapping

- Linear Discriminant Analysis (LDA)
- Independent Component Analysis (ICA)
- Non-negative Matrix Factorization (NMF)
- Auto-encoder
- Self-organizing maps
- Principal component analysis (and its variants)

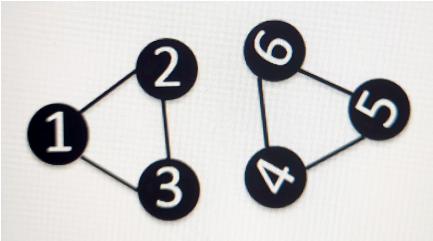
Knowledge Check

- Why do we need dimensionality reduction? (Select all that apply.)
    - Reduce the competition and storage space required
    - Mitigate the curse of dimensionality
    - Easy to visualize the data
- Which of the following is a dimensionality reduction technique? (Select all that apply.)
    - Non-negative matrix factorization (NMF)
    - Principal Component Analysis (PCA)
    - Independent Component Analysis (ICA)
- Which of the following is a way to find out whether the dimensionality reduction algorithm performs well?
    - The reconstruction error (the smaller the better)

Module 6 Quiz Questions

- Which of the following dataset sets have a natural graph structure? (Select all that apply):

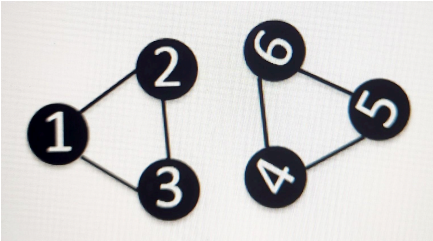
    - 1. A set of dog images
    - 2. Web pages and the hyperlink
    - 3. Social network
    - Selected 2 & 3
- Which of the following statements are true about the L matrix? (Select all that apply):
    - 1. The smallest eigenvalue of L is greater than 0
    - 2. L is a symmetric matrix
    - 3. L is a Semi-positive definitive matrix
    - Selected 2 and 3
- Which algorithm would we use to get the partition shown by the curve in the following figure? (There are a set of 6 red colored points and 5 blue colored points on the side of the plane. There is one blue colored point on the other plane.)
    - Ncut
- Which of the following approach considers neither inter-cluster nor intra-cluster similarity?
    - MinCut
- True or False: using 2-way partitioning recursive to get k-way partition is inefficient and may cause stability issues
    - True
- What is the spectral gap?
    - The smallest non-zero eigenvalue of L
- Compared to K-means, which of the following is an advantage of spectral clustering?
    - Does not make strong assumptions on the forms of the clusters
- Given a graph with 6 nodes (i.e. data points) in the following figure, we want to run the spectral clustering for MinCut to find two clusters.



(There is a graph with the following configuration: there is an edge connecting Node 1 and Node 2, there is an edge connecting Node 2 and Node 3, there is an edge connecting Node 1 and Node 3, there is an edge connecting Node 4 and Node 5, there is an edge connecting Node 5 and Node 6, there is an edge connecting Node 4 and Node 6).
What is the adjacency matrix?

    - $$W = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- Given a graph with 6 nodes (i.e. data points) in the following figure, we want to run the spectral clustering for MinCut to find two clusters.

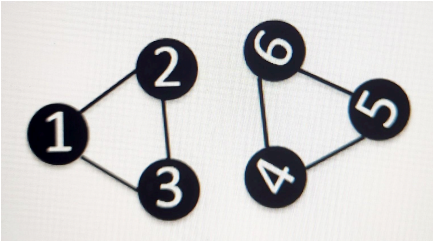


(There is a graph with the following configuration: there is an edge connecting Node 1 and Node 2, there is an edge connecting Node 2 and Node 3, there is an edge connecting Node 1 and Node 3, there is an edge connecting Node 4 and Node 5, there is an edge connecting Node 5 and Node 6, there is an edge connecting Node 4 and Node 6).

What is the degree matrix?

  - $$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

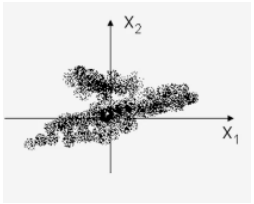
- Given a graph with 6 nodes (i.e. data points) in the following figure, if we run the spectral clustering for MinCut to find two clusters given as {1,2,3} and {4,5,6}.



- (There is a graph with the following configuration: there is an edge connecting Node 1 and Node 2, there is an edge connecting Node 2 and Node 3, there is an edge connecting Node 1 and Node 3, there is an edge connecting Node 4 and Node 5, there is an edge connecting Node 5 and Node 6, there is an edge connecting Node 4 and Node 6).

  What is the cut size?
  - 0

Principal Component Analysis - Basic Idea



- Look at a simple 2-D to 1-D example: we want to use a single feature to describe the 2-D
- Consider these possibilities
  - Naive: randomly discard one dimension
  - Better: discard the less-descriptive one (x_2 in the figure)
  - Much better: project the data to a most-descriptive direction and use the projections

How to formulate this idea?

- "Most descriptive" ≈ Largest "variance"
- So the problem is to find the direction of the largest variance
- Problem: Given n samples $D = \{x_1, x_2, ..., x_n\}$ in d-dimensional space, find a direction e_1, such that the projection of D onto e_1 gives the largest variance (compared with any other dimension)
- e_1 is a d-dimensional vector with unit norm

Find e_1

- Let's compute the variance of the projected data on a given direction e
  - The n projected samples are given as $i = 1, ..., n$
    - $y_i = x_i \cdot e$
  - The mean of the projections
    - $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n} x_i \cdot e = \bar{x} \cdot e$
  - Thus the (sample) variance of the projections:
    - $\sigma^2(e) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \bar{y})^2 = \frac{1}{n}\sum_{i=1}^{n} \left[\left(x_i - \bar{x}\right) \cdot e\right]^2$
- $\sigma^2(e) = \sum_{j=1}^{d}\sum_{k=1}^{d} e_j e_k \left[\frac{1}{n}\sum_{i=1}^{n}\left(x_{i,j} - \bar{x}_{i,j}\right)\left(x_{i,k} - \bar{x}_{i,k}\right)\right] = \sum_{j=1}^{d}\sum_{k=1}^{d} e_j e_k C_{jk} = e^t C e$
- To find e_1, we can do
  - $e_1 = argmax\sigma^2(e)$ subject to $||e|| = 1$
- Constrained maximization: use Lagrange multiplier method
  - $maximize\, F(e) = e^t C e - \lambda\left(e^t e - 1\right)$
- Set the partial derivative to 0, we have
  - $\frac{\partial F}{\partial e} = 2Ce - 2\lambda e = 0$
  - $\rightarrow Ce = \lambda e$
- The solution is an eigenvector of C, with eigenvalue λ, which is also the variance under e:
  - $\sigma^2(e) = e^t C e = \lambda$
- We should set e_1 to be the eigenvector corresponding to the largest eigenvalue λ_1

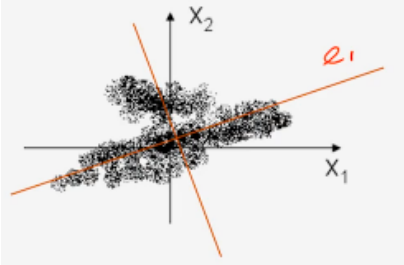
Recap of the Key Idea

- We want to project the given samples to certain direction so that the variance is maximized, compared with any other direction

Principle Component Analysis - The Algorithm and Important Properties

Principle Components

- We found e_1, which gives the direction of the largest variance after projection
    - The first principal component
- The process can be continued in the subspace orthogonal to e_1, and so on and so forth
    - Obtaining other principal components: e_2, e_3, etc., corresponding to other eigenvectors of C, ordered by the corresponding eigenvalues λ_i
- The principal components are orthogonal to each other $\rightarrow \{e_i\}$ forms an orthonormal basis in the d-dimensional space



- The total variance is given by the sum of variances of the projections
    - $\sigma^2 = \sum_{j=1}^{d} \lambda_j$

How Many Principal Components to Keep?

- To reduce dimensions, we will need to keep only $d' << d$ projections
- We can measure how much of the total variance of a d'-dimensional subspace captures, by the ratio
    - $\frac{\sum_{j=1}^{d'} \lambda_j}{\sum_{j=1}^{d} \lambda_j}$
- Variance may be related to the "energy" of a signal: how accurately we want to represent the data
    - The ratio can be used to guide in choosing in proper d' for desired accuracy

The PCA Algorithm

1. Compute the dxd sample covariance matrix C
2. Find the eigenvalues and the corresponding eigenvectors of C
3. Project the original data onto the space spanned by the eigenvalues
    a. The projection may be done onto a d'-dimensional subspace spanned by the first d' eigenvectors (ordered by the eigenvalue in descending order)
        i. d' is determined by the desired accuracy

Important Properties of PCA

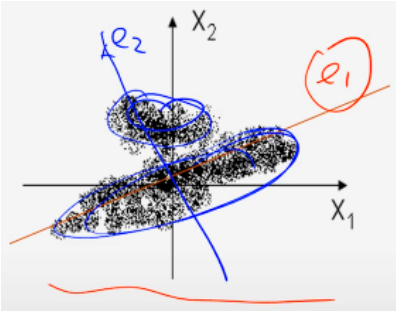
- PCA represents the data in a new space, in which the components of the data is ordered by their "significance"
    - Dimension reduction can be done by simply discarding less significant dimensions
- Linearity assumption → extensions exist
- "Variance ≈ Importance" is meaningful only under large *signal-to-noise ratio*

PCA as Feature Mapping

- When we use only d' dimensions from PCA (with original dimension $d > d'$), this may look like feature section
    - But in general they are different approaches
- PCA
    - Unsupervised (in general)
    - Generates new features (linear combination of original ones)
- Feature Selection
    - Supervised (in general)
    - Selects a few original features (e.g., for better classification)

Can PCA help classification?

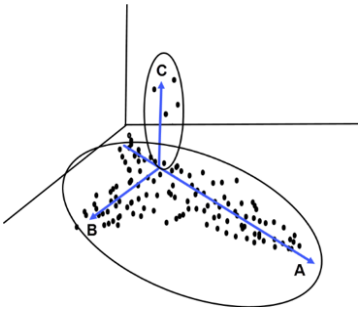
- Can we do better classification in a lower-dimensional space from d' principal components given by PCA?
    - Not necessarily
- LDA may be better posed for such a task



Knowledge Check

- What is the primary purpose of PCA?
    - To reduce the number of input features
- Which of the following is true about PCA? (Select all that apply.)
    - It searches for the directions that data have the largest variance
    - All principal components are orthogonal to each other
    - PCA is an unsupervised approach
- When eigenvalues are roughly equal, which of the following is true?
    - PCA will not be very useful for dimensionality reduction
        - PCA cannot be very useful since any direction is about equally important.

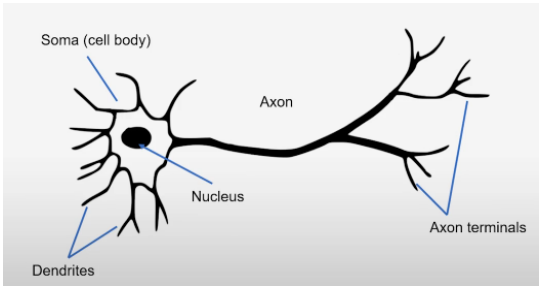
- Given a data matrix whose rows are data samples and whose columns represent feature dimensions, which of the following techniques would perform better for reducing dimensions of the data set?
    - Removing a column that contains identical entries
- What happens when you obtain features in lower dimensions using PCA?
    - A feature is in general mapped from original features, and thus cannot be easily linked to any single dimension of the original features
- Which of the following option(s) is / are true?
    - For a given a date set, you can run PCA without knowing labels of the samples
- In the following figure, which vector is the best candidate for the second principal component?



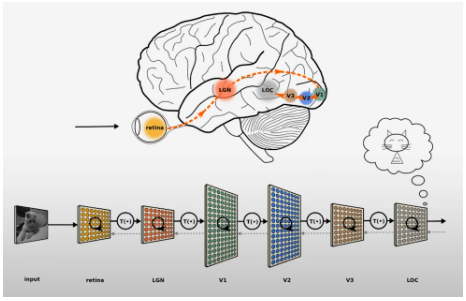
    - Vector B

Neural Networks and Deep Learning - Part 1

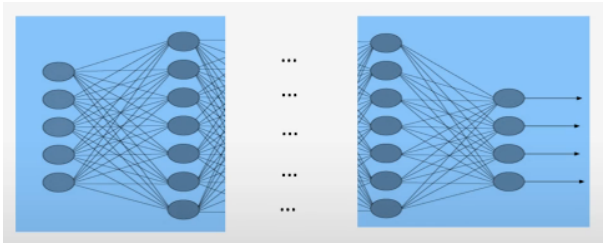
Illustrating a Biological Neuron



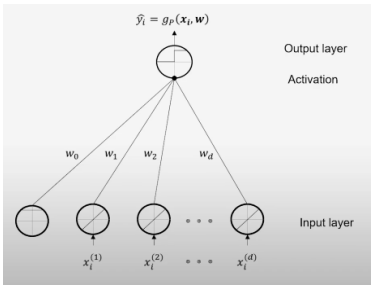
Neural Network



Artificial Neural Networks

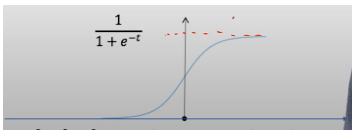


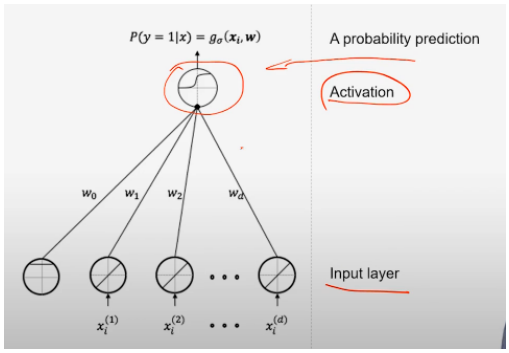
Building Artificial Neural Networks



- What does this "neuron" do?
    - The Perceptron model

Logistic Neuron

- In Perceptron: $g_P(x_i, w) = \{1, if\ w^T x_i > 0\}, \{0, otherwise\}$
- If we let: $g_\sigma(x_i, w) = \frac{1}{1+e^{-w^T x_i}}$
- The logistic function: $\frac{1}{1+e^{-t}}$ 

- We have:
    - 

Neural Networks and Deep Learning - Part 2

Learning in the Perceptron

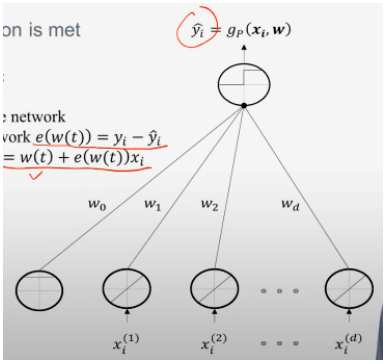
- "Learning": how does the neuron adapt its weights in response to the inputs?

The Perceptron Learning Algorithm

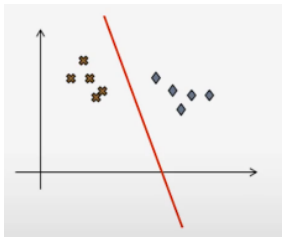
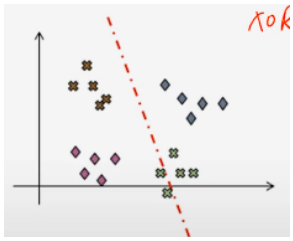
- Input
    - Training set
    - $D = \{(x_i, y_i), i \in [1, 2, ..., n]\}\ y_i = [0, 1]$
- Initialization
    - Initialize the weights $w(0)$ (and some thresholds)
    - Weights may be set to 0 or small random values

- Iterate for t until a stop criterion is met

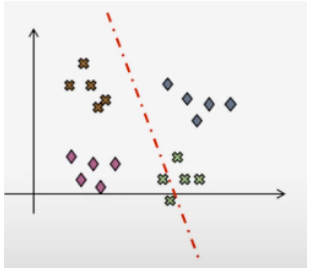
```
{
   For each sample x_i with label y_i:
   {
      compute the output ȳ_i of the network
      estimate the error of the network e(w(t)) = y_i − ȳ_i
      update the weight w(t + 1) = w(t) + e(w(t))x_i
   }
   t++
}
```



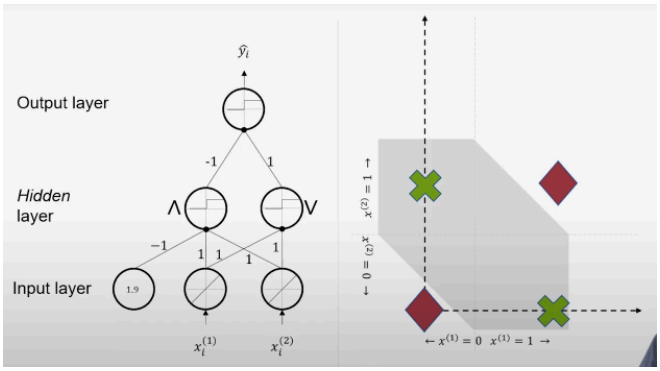
The Need for Multiple Layers

- This is easy can can be learned by the Perceptron Algorithm
  - $w_1 x_1 + w_2 x_2 + w_0 = 0$
  - 
- But how about this?
  - 

Extending to Multi-layer Neural Networks

-  → the XOR Problem
- Question: Can a multi-layer version of the Perceptron (MLP) help solving the XOR problem?

An MLP Solving the XOR Problem



Neural Networks and Deep Learning - Part 3

The Question of Learning

- How can the network learning proper parameters form the given samples?
    - Can the Perceptron algorithm be used?

Difficulty in Learning for MLP

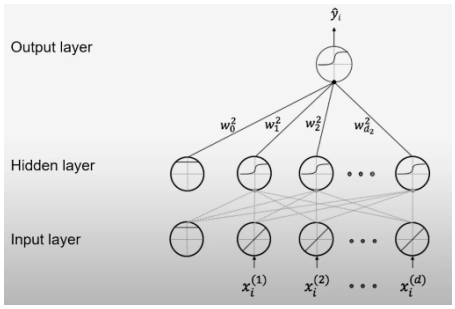
- Perceptron Learning Algorithm
    - The weight update of the neuron is proportional to the "error" computed as $y_i - \overline{y}_i$
        - This requires us to know the target output
- Multi-layer Perceptron
    - Except for the neurons on the output layer, other neurons (on the *hidden* layers) do not really have a target output given

Back-Propagation (BP) Learning for MLP

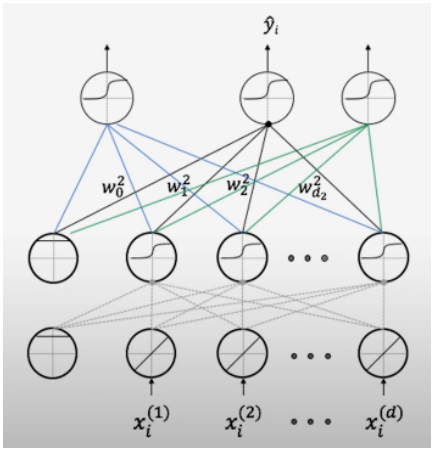
- The key: properly distribute error computed from output layer back to earlier layers to allow their weights to be updated in a way that reduce the error
    - The basic philosophy of the BP algorithm
- Differentiable activation functions
    - We can use - e.g.,
        - the logistic neurons
        - neurons with sigmoid activation
        - or its variants

A Multi-Layer Neural Network

- Using Logistic Neurons

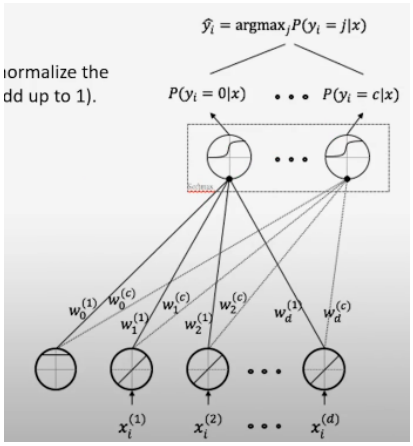


Handling Multiple (>2) Classes



Softmax for Handling Multiple Classes

- Using softmax to normalize the outputs (so they add up to 1)



How to compute "errors" in this case?

- Consider the cross-entropy as a loss function:
  - $l(W) = \sum_{i=1}^{n} \sum_{j=1}^{C} \Pi_j(y_i) log P(y_i = j|x_i)$
  - $\Pi_j(y_i) = \{1, if\ y_i - j\}, \{0, otherwise\}$

Knowledge Check

- Under which of the following conditions does a neuron fire its own signal?
    - When it receives a sufficient number of signals from other neurons within a timeframe
- An MLP neural network architecture contains which of the following? (Select all that may apply)
    - Hidden layer
    - Output layer
    - Input layer
- This figure shows a simple neural network. An observation with two variables is presented to the network, as shown below. What is the updated weight if the linear function y=x is used as the activation function and the Mean Squared Error is used as the error function?

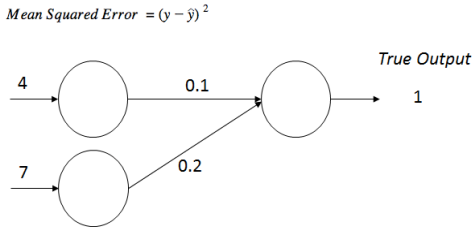


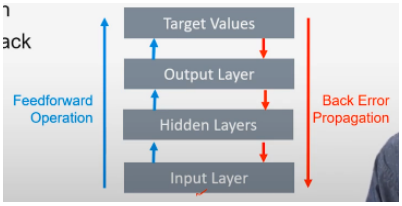
Diagram illustrating mean square error calculation: sum of squared differences between predicted and actual values, divided by the number of data points.

    - (2.66, 4.68)
        - The updated weights are (2.66, 4.68)
- What does the backpropagation algorithm do?
    - To distribute the output error proportionally to the nodes in the hidden layer

Key Techniques Enabling Deep Learning

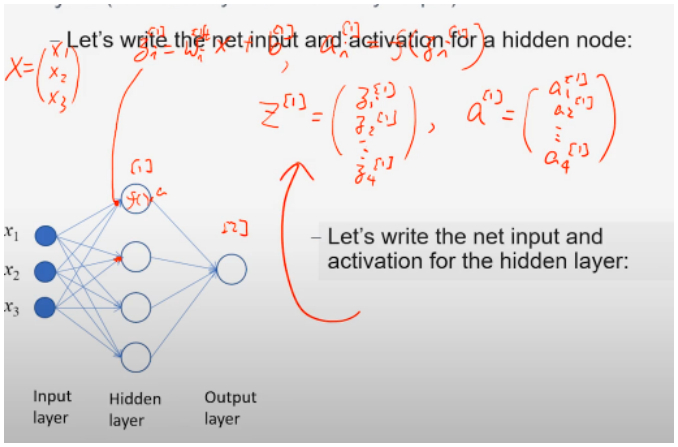
Back Propagation (BP) Algorithm

- Simple Perceptron algorithm illustrates a path to learning by iterative optimization
    - Updating weights based on network errors under current weights, and optimal weights are obtained when errors become 0 (or small enough)
- Gradient descent is a general approach to iterative optimization
    - Design a loss function J
    - Iteratively update the weights W according to the gradient of J with respect to W
- $W \leftarrow W - \eta \nabla J(W)$
    - W is the parameter of the network; J is the objective function
- Generalizes/Implements the idea for multi-layer networks
    - Gradient descent for updating weights in optimizing a loss function
    - Propagating gradients back through layers
        - hidden layer weights are linked to loss gradient at output layer



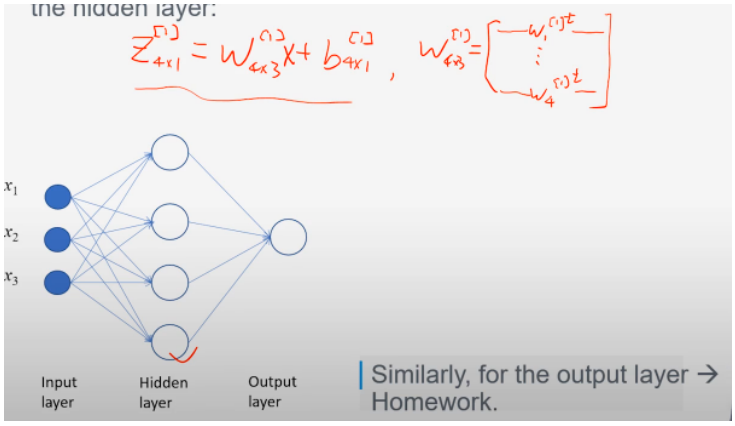
Illustrating the BP Algorithm 1/6

Let's consider a simple neural network with a single hidden layer (We will only outline the key steps)



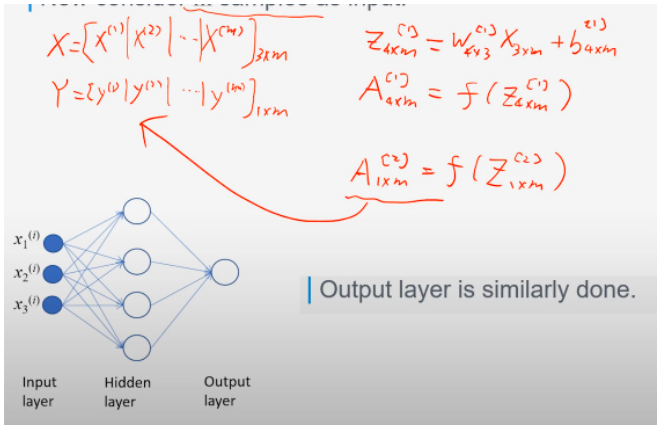
Illustrating the BP Algorithm 2/6

Using matrix/vector notations, for the hidden layer:



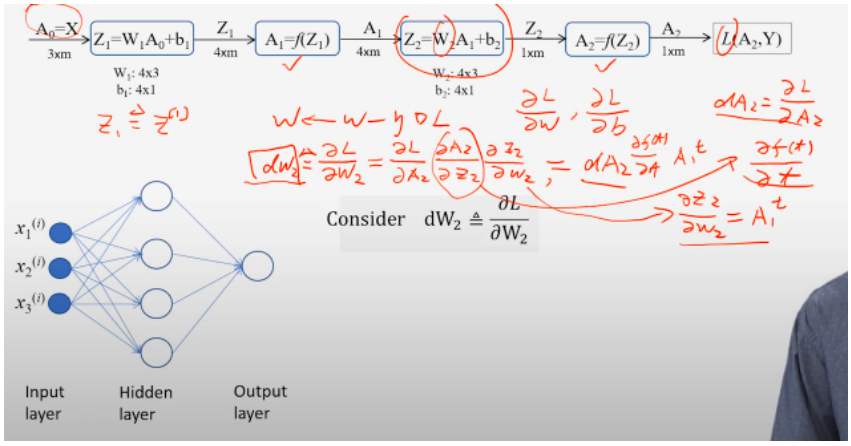
Illustrating the BP Algorithm 3/6

Now consider m samples as input



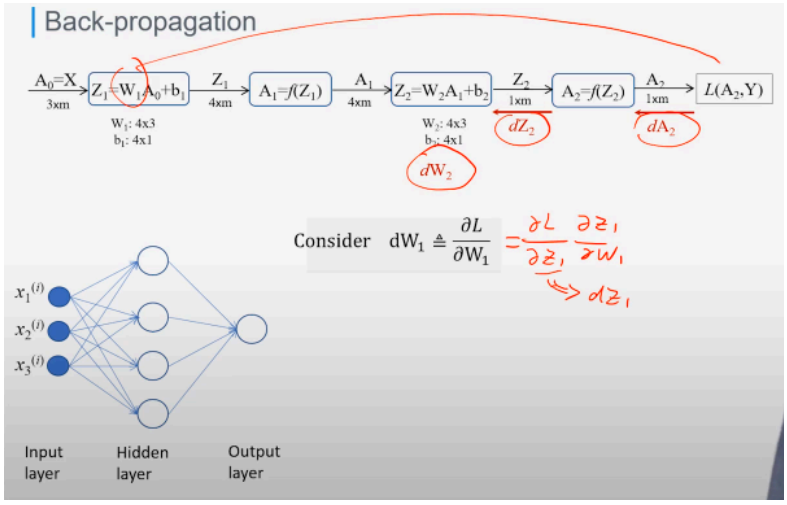
Illustrating the BP Algorithm 4/6

Overall we have this flow of *feedforward* processing (not the notation change for simplicity: subscripts are for layers):



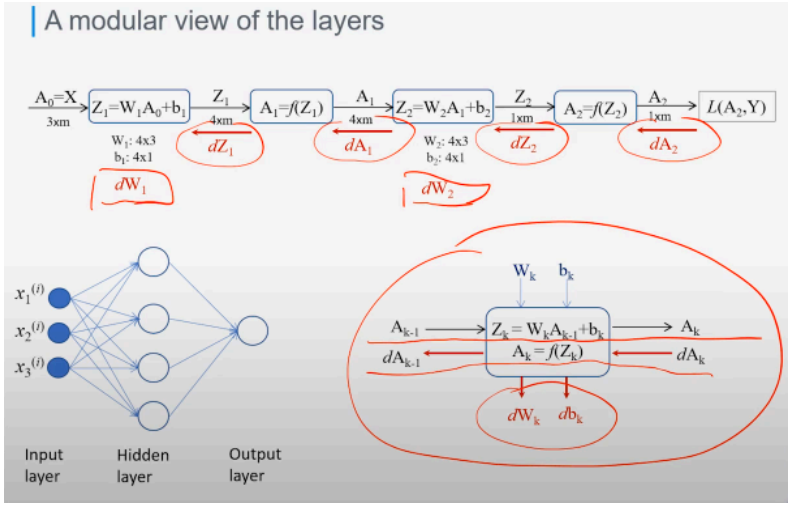
Illustrating the BP Algorithm 5/6

Back-propagation



Illustrating the BP Algorithm 6/6

A modular view of the layers



BP Algorithm Recap

- The feedforward process: ultimately produce $A^{[K]}$ that leads to the prediction for Y
- The backpropagation process:
    - First compute the loss
    - Then compute the gradients via back-propagation through layers
    - Key: use the chain rule of differentiation
- Essential to deep networks
- Suffers from several practical limitations
    - gradient exploding
    - gradient vanishing etc.

Activation Functions: Importance

- Provides non-linearity
- Functional unity of input-output mapping
- Its form impacts on gradients in BP algorithm

ReLU and Some Variants

- $a_{ReLU}(x) = max(0, x)$
- $a_s(x) = log(1 + e^x)$
- $a_n = max(0, x + \varphi)$, **with** $\varphi \sim N(0, \sigma(x))$
- $a_L(x) = \{x, if\ x > 0\}, \{\delta x, otherwise\}$
    - with δ a small positive number

The Importance of Regularization

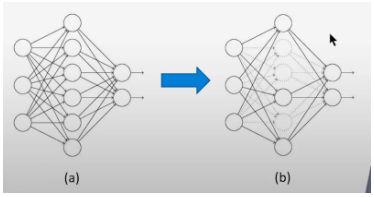
- The parameter space is huge, if there is no constraint in search for a solution, the algorithm may converge to poor solutions
- Overfitting is a typical problem
    - Converging to local minimum goof only for the training data

Some Ideas for Regularization

- Favoring a network with small weights
    - achieved by adding a term of L2-norm of the weights to original loss function
- Preventing neurons from "co-adaption" → Drop-out
- Making the network less sensitive to initialization/learning rate etc. → Batch normalization
- Such regularization techniques have been found to be not only helpful but sometimes critical to learning in deep networks

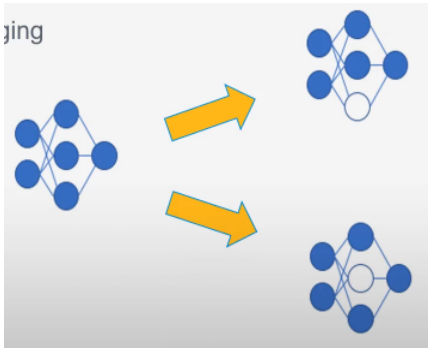
Drop-out

1. Obtain (b) by randomly deactivate some hidden nodes in (a)
2. For input x, calculate output y by using the activated nodes ONLY
3. Use BP to update weights (which connect to the captivated nodes) of network
4. Activate all nodes
5. Go back to first step



Why Drop-out?

- Reducing co-adaptation of neuron
- Model averaging



Batch Normalization (BN)

- Inputs to network layers are of varying distributions, the so-called internal coverage shift [Ioffe and Szegedy, 2015]
    - Careful parameter initialization and low learning rate are required
- BN was developed to solve this problem by normalizing layer inputs of a batch

How is BN Used in Learning?

- Define two parameters β and γ so that the output of the BN layer can be calculated as:
    - $y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$
- Parameters β and γ can be learned by minimizing the loss function via gradient descent
- Usually used right before the activation functions
-

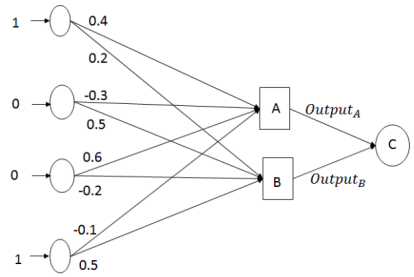
Other Regularization Techniques

- Weight sharing
- Training data conditioning
- Sparsity constraints
- Ensemble methods (committee of networks)

- Come of these will be discussed in later examples of networks

Knowledge Check

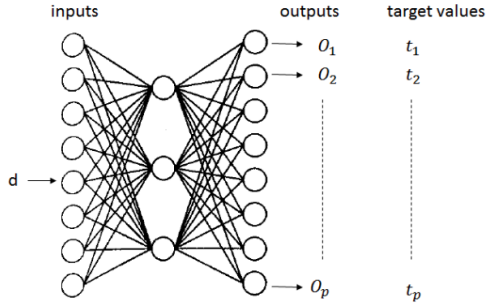
- The figure below shows a simple neural network. An observation with four variables (1,0,0,1) is presented to the network as shown. What is the output of hidden node A, outputA, if sigmoid activation function is used?



A network diagram showcasing two nodes and two outputs, illustrating the connectivity and flow within the system.

  - $\frac{1}{1+e^{-0.3}}$
    - Calculate the value of $\Sigma_i w_i x_i$. Then use the sigmoid function

- A network with multiple output nodes is given below. The target values, t_i, are the true values of the network and outputs, o_i, are the computed values.



For input d and weight vector w, the error function is defined as

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{k \in outputs} (t_k - o_k)^2$$

What is the value of the following expression?

$$\frac{\partial E}{\partial o_j}$$

  - $o_j - t_j$
    - Taking the partial derivative with respect to oj returns the value of $o_j - t_j$

- Which of the following improves the vanishing gradient problem?
  - The use of the ReLU function
    - The ReLU function was found to alleviate solves the vanishing gradient problem.
- Which of the following is NOT true for the drop-out regularization technique?

- The drop-out technique uses the backpropagation algorithm to update the weights of all nodes, including the deactivated nodes
    - The backpropagation algorithm updates the weights of active nodes only.

Some Basic Deep Architectures - Part 1

Overview

- Convolutional Neural Network (CNN)
    - will be given the most attention, for its wide range of application
- Auto-encoder
- Recurrent Neural Networks (RNN)

Convolutional Neural Network (CNN)

- Most useful for input data defined on grid-like structures, like images or audio
- Built upon concept “convolution” for signal/image filtering
- Invokes other concepts like pooling, weight-sharing, and (visual) receptive field, etc.

Image Filtering via Convolution - 1 of 5

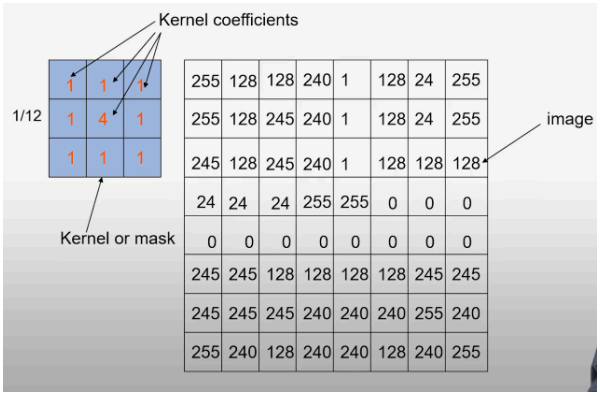


Image Filtering via Convolution - 2 of 5

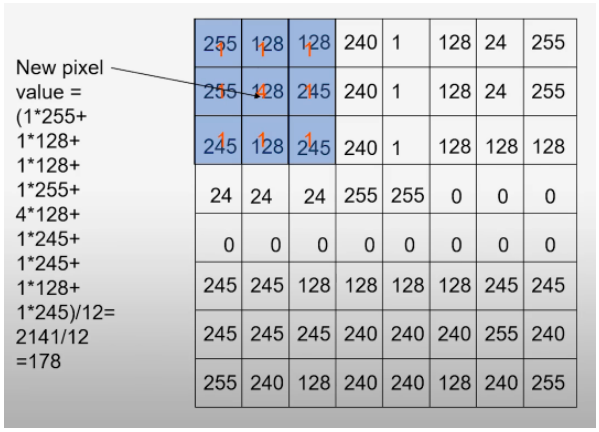


Image Filtering via Convolution - 3 of 5

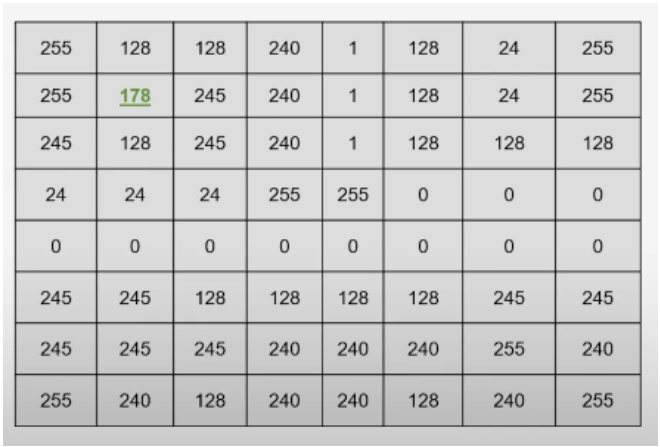


Image Filtering via Convolution - 4 of 5

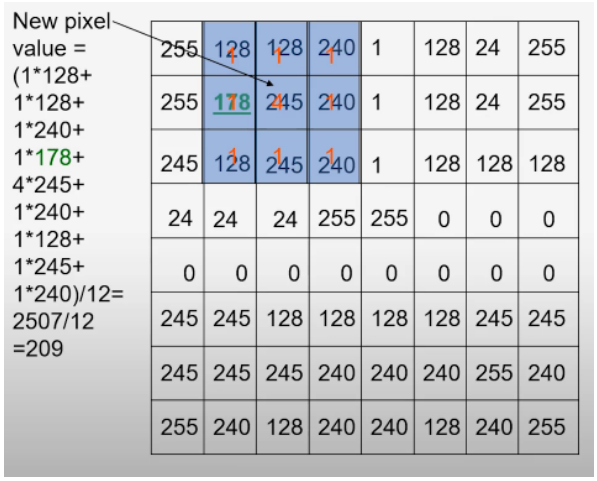
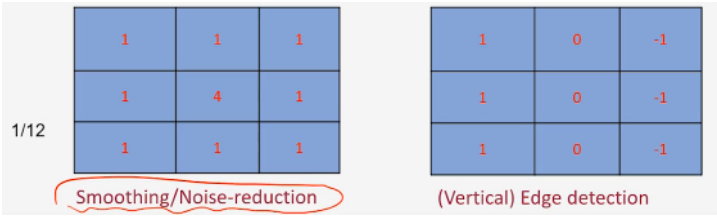


Image Filtering via Convolution - 5 of 5

255	128	128	240	1	128	24	255
255	178	209	240	1	128	24	255
245	128	245	240	1	128	128	128
24	24	24	255	255	0	0	0
0	0	0	0	0	0	0	0
245	245	128	128	128	128	245	245
245	245	245	240	240	240	255	240
255	240	128	240	240	128	240	255

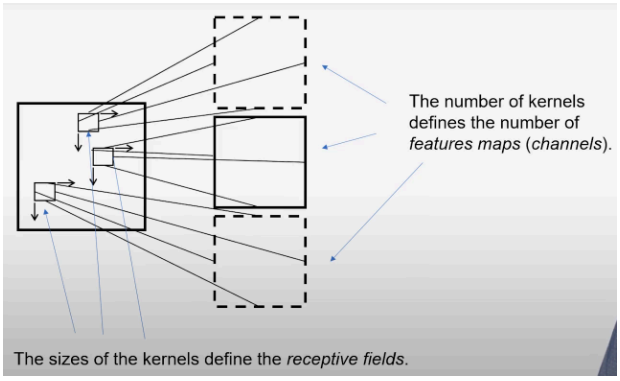
Image Filtering via Convolution: Kernels

- Examples of Kernels:

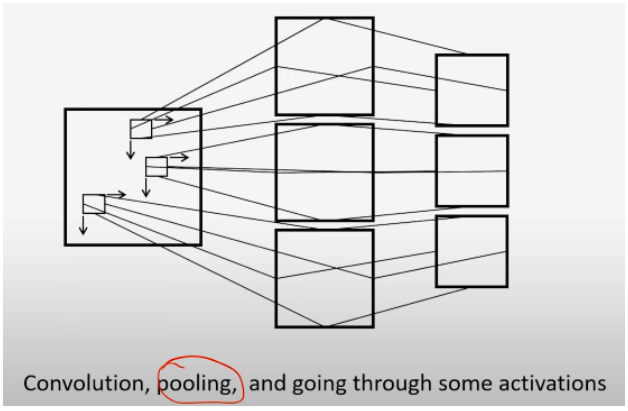


- By varying coefficients of the kernel, we can achieve different goals
    - Smoothing, sharpening, detecting edges, etc.
- Better yet: can we learn proper kernels? → Part of CNN objective

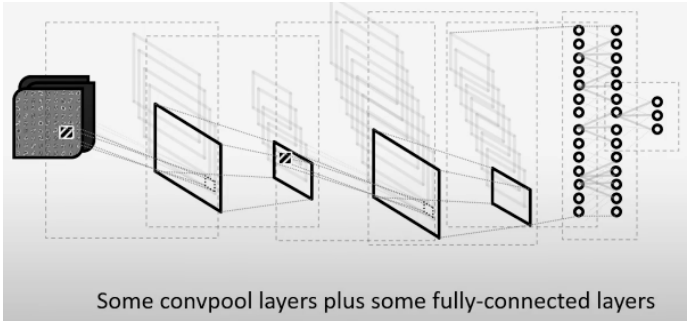
2D Convolutional Neuron



Convpool Layer



Illustrating A Simple CNN



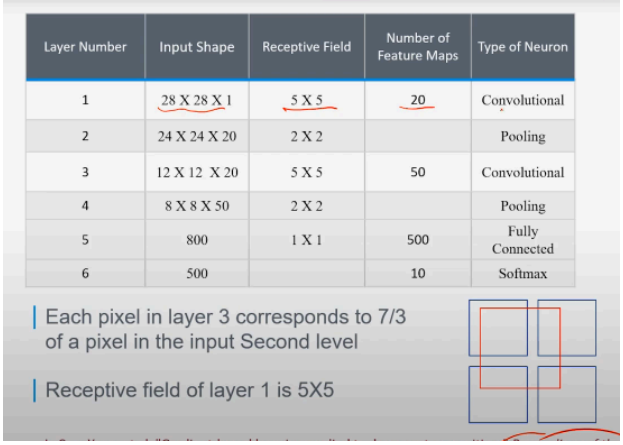
- Some convpool layers plus some fully-connected layers

Some Basic Deep Architectures - Part 2

CNN Examples - Different Complexities

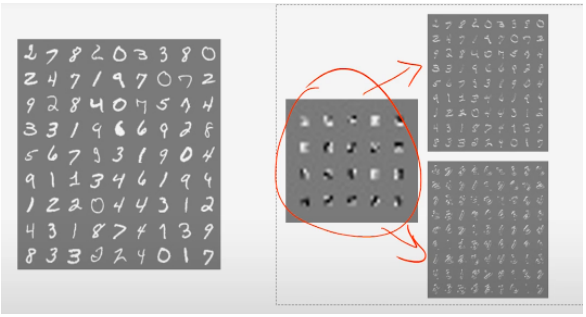
- LeNet
- AlexNet

LeNet



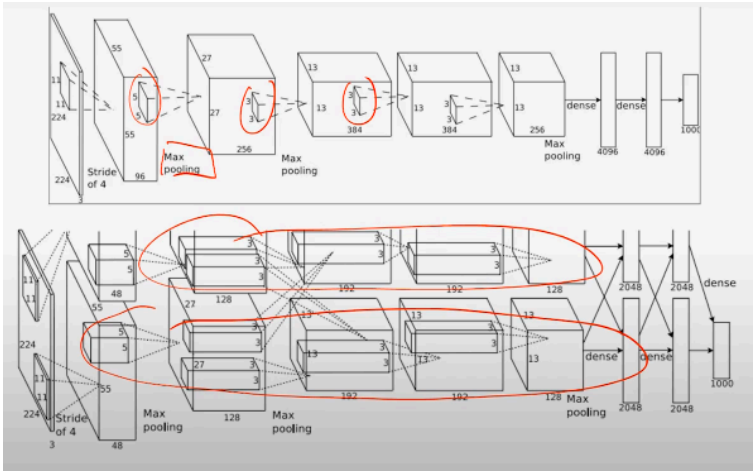
Layer Number	Input Shape	Receptive Field	Number of Feature Maps	Type of Neuron
1	28 X 28 X 1	5 X 5	20	Convolutional
2	24 X 24 X 20	2 X 2		Pooling
3	12 X 12 X 20	5 X 5	50	Convolutional
4	8 X 8 X 50	2 X 2		Pooling
5	800	1 X 1	500	Fully Connected
6	500		10	Softmax

Case Study: LeNet



- This is after training the network for 75 epochs with a learning rate of 0.01
- Produces an accuracy of 99.38% on the MNIST dataset

Case Study: AlexNet



Case Study: AlexNet - 1 of 3

Layer Number	Input Shape	Receptive Field	Number of Kernels	Type of Neuron
1	224 X 224 X 3	11 X 11, stride 4	96	Convolutional
2		3 X 3, stride 2		Pooling
3	55 X 55 X 96	5 X 5	256	Convolutional
4		3 X 3, stride 2		Pooling
5	13 X 13 X 256	3 X 3, padded	384	Convolutional
6	13 X 13 X 384	3 X 3, padded	384	Convolutional
7	13 X 13 X 384	3 X 3	256	Convolutional
8	43264	1 X 1	4096	Fully Connected
9	4096	1 X 1	4096	Fully Connected
10	4096		1000	Softmax

- Receptive field of the layer 7 is
    - 52 pixels !! which is almost as big as an object part (about one - fourth of the input image)

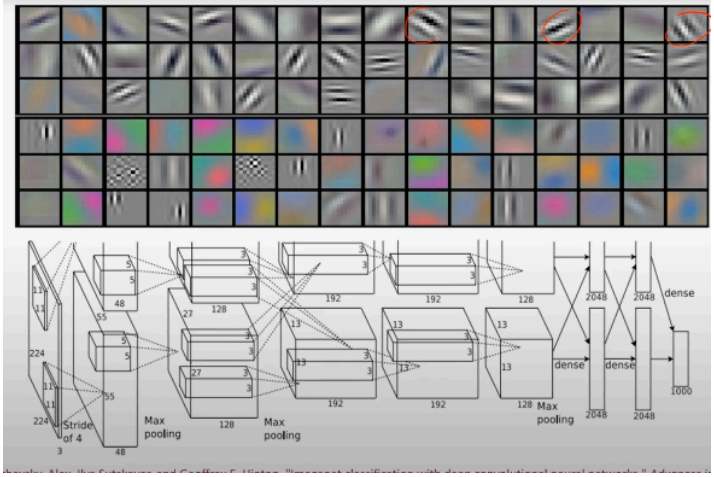
Case Study: AlexNet - 2 of 3

- Imagenet - 15 million images in over 22,000 categories
- Imagenet categories are much more complicated than other datasets
    - Often difficult even for humans to categorize perfectly
    - Average human-level performance is about 96% on this dataset
- (ILSVRC), use about 1000 of these categories
- AlexNet was the earliest systems to break the 80% mark
    - Non-neural conventional techniques were unable to achieve such performance

Case Study: AlexNet - 3 of 3

- AlexNet was huge at the time
    - The size could lead to instability during training or inability to learn, if without proper regularization
- Some techniques were used to make it traininable
    - AlexNet was the first prominent network to feature ReLU
    - Features multi-GPU training (originally trained the networks on two Nvidia GTX 580 GPUs with 3GB)

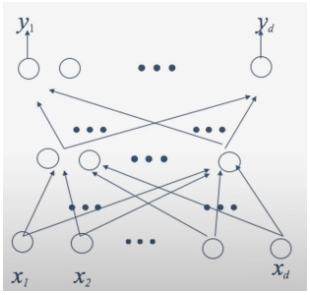
Case Study: AlexNet Filters



CNN Recap

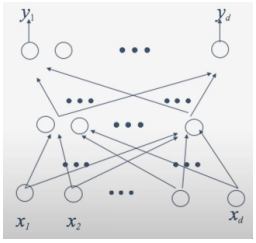
- The CNNs are similar to the basic MLP architecture illustrated earlier, but some key extensions include:
  - The concept of weight-sharing through kernels
  - Weight-sharing enables learnable kernels, which in turn define feature maps
  - The idea of pooling

Auto-encoder - 1 of 4



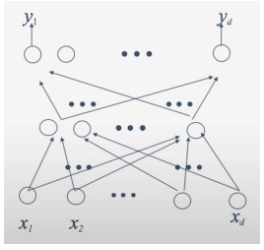
- Networks seen this far are all training via supervised learning
- Sometimes we may need to train a network without supervision
  - Unsupervised learning
- Auto-encoder is such example
  - Consider y_i being an approximation of x_i

Auto-encoder - 2 of 4



- Perfect auto-encoder would map x_i to x_i
- Learn good representations in the hidden layer

Auto-encoder - 3 of 4



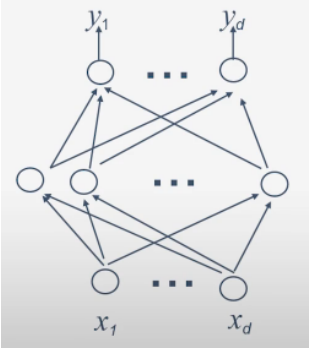
- Consider two cases
  1. Much fewer hidden nodes than input nodes
  2. Many hidden nodes or more hidden nodes than input nodes
- Case 1: Encoder for compressing input and compressed data should still be able to reconstruct the input
  - Similar to, e.g.m PCA

Auto-encoder - 4 of 4

- Consider two cases

1. Fewer hidden nodes than input nodes
2. More hidden nodes than input nodes

- Case 2: Allow more hidden nodes than input
  - Allow more freedom for the input-to-hidden layer mapping in exploring structure of the input
  - Additional "regularization" will be needed in order to find meaningful results

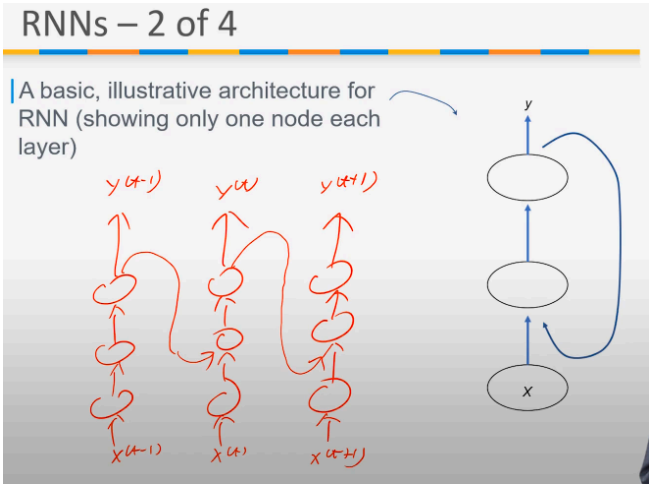


Recurrent Neural Networks (RNNs) - 1/4

- Feedforward networks: Neurons are interconnected without any cycle in the connection
- Recurrent neural networks: Allow directed cycles in connections between neurons
  - Notions of "state" or temporal dynamics
  - Necessity of internal memory
- One clear benefit: Such networks could naturally model variable-length sequential data

RNNs - 2 of 4

- A basic, illustrative architecture for RNN (showing only one node each layer)

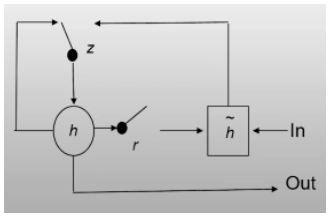


RNNs - 3 of 4

- Training with BP algorithm may suffer from so-called vanishing gradient problem
- Some RNN variants have sophisticated "recurrence" structures, invented in part to address such difficulties faced by basic RNN models

RNNs - 4 of 4

Examples:

- The “Long short-term memory” (LSTM) model
    - used to produce state-of-the-art results in speech and language applications
- The Gated Recurrent Unit model, illustrated here:
    - 

Knowledge Check

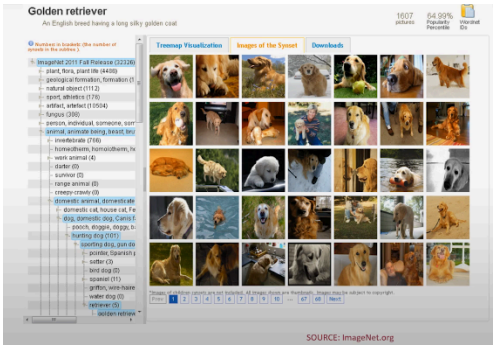
- Which of the following is NOT true for CNNs?
    - The CNNs devise a solution to increase the number of parameters used in a deep neural network
        - The CNNs devise a solution to decrease the number of parameters used in a deep neural network.
- What is the purpose of the pooling layer in CNNs?
    - The purpose of the pooling layer in CNNs is to reduce the dimensionality of feature maps
        - The purpose of the pooling layer in CNNs is to reduce the dimensionality of feature maps.
- Which of the following is NOT true for AlexNet architecture?
    - The performance was below 80%
        - The performance was not below 80%.’

Exemplar Deep Learning Applications

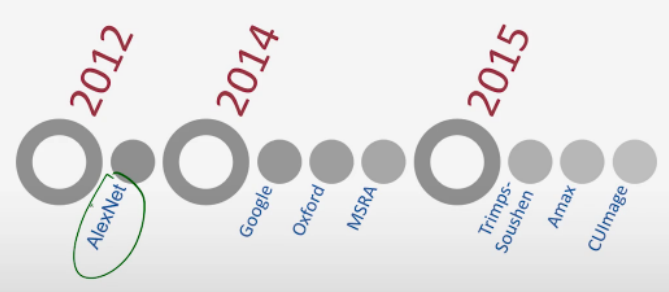
Deep Learning for Image-based Recognition

- Visual recognition is an important part of human intelligence
- ILSVRC (ImageNet Large-scale Visual Recognition Challenge) illustrates such a task
- Many ImageNet images are difficult to classify

ImageNet.org Samples



Success Stories



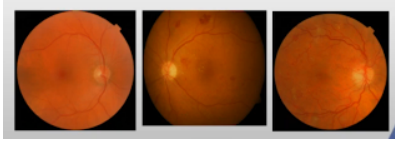
Success Stories: 2014 - Top Three

Rank	Team	Error
1	Google	0.06656
2	Oxford	0.07325
3	MSRA	0.08062

Team Name	Entry Description	Description of Outside Data Used	Localization Error	Classification Error
Trimps-Soushen	Extra annotations collected by ourselves	Extra annotations collected by ourselves	0.122285	0.04581
Amax	Validate the classification model we used in DET Entry1	Share proposal procedure with DET for convenience	0.14574	0.04354
CUImage	Average multiple models – validation accuracy is 79.78%	3000-class classification images from ImageNet are used to pre-train CNN	0.198272	0.05858

Example Application 1: DR Detection

- DR: Diabetic Retinopathy
- A recent work: Gulshan et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs." JAMA 316.22 (2016): 2402-2410
    - Employed large datasets
    - A specific CNN architecture (Inception-v3) taking the entire image as input (as opposed to lesion/structure-specific CNNs)
    - High performance: Comparable to a panel of 7 board-certified ophthalmologists

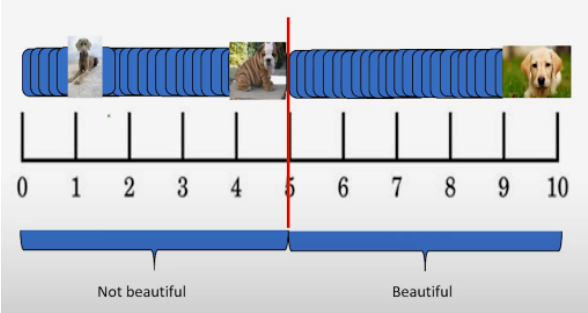


Example Application 2: Visual Aesthetics

- While being subjective, computational modes are possible since there are patterns visually-appealing pictures
    - E.g., photographic rules
- Huge on-line datasets available. If ratings are also available, the problem becomes supervised learning.
    - Conventional approaches still face the bottleneck of feature extraction

Related Approaches

- Solving the task as binary classification



- Image retrieval
- Image enhancement

A Deep Learning Approach

- DUal-channeled CNN trained using relative learning
- Siamese Network characteristics (weight sharing) and hinge-loss function
- A custom data-set with relative labels - pairs formed based on aesthetic rating

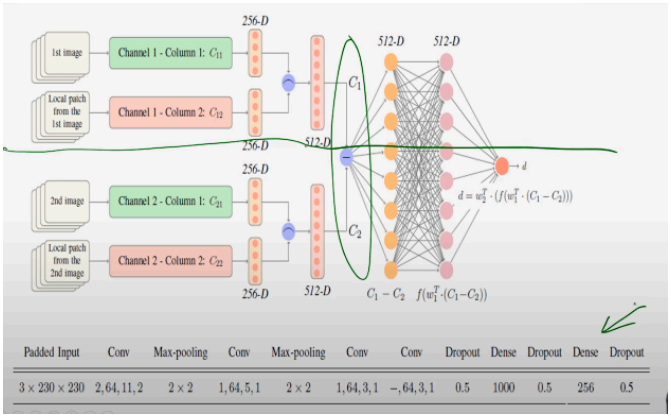
Constructing a Useful Data Set - 1/2

- Total of 250,000 images extracted from dpchallenge.com
- Challenges under which users post their submission
- Peers rate and a final winner is selected based on the average rating
- Belong to a wide variety of semantic categories

Constructing a Useful Dataset - 2/2

- The minimum gap between the average rating of the two images is one
    - e.g., 3.4 and 4.5, 6.3, and 7.8
- The maximum variance allowed between the ratings of different voters is 2.6
- Pick pairs from the same category only
    - e.g., cannot compare an image of a car and a building

The Network Architecture & Other Characteristics



Further Implementation Details

- Each channel contains two streams of processing: column 1 for global, and column 2 for local
- Global Patch
    - e.g., rule of thirds, golden ratio
- Local Patch
    - e.g., smoothness/graininess

The Loss Function

$L = max(o, \delta - y \cdot d(I_1, I_2)) \rightarrow$ Hinge Loss

$d(I_1, I_2) = f(C_1 - C_2)$

where,

$y = True\ label\ of\ the\ image\ pair,$

i.e. $1\ if\ I_1 > I_2$ and $-1\ otherwise$

- $C_1, C_2 = Outputs\ of\ channel\ 1\ and\ channel\ 2\ respectively$

Eight Experiments Total

	Ranking (custom test-set)	Ranking (standard test-set)	Classification (custom test-set)	Classification (standard test-set)
Base-line	62.21	65.87	**59.92**	69.18
Relative aesthetics	**70.51**	**76.77**	59.41	**71.60**

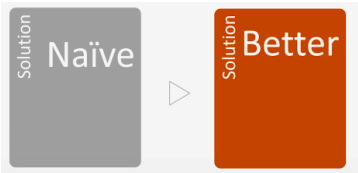
Knowledge Check

- Which of the following is not one of the deep network approaches for solving the image classification problem?
    - K-Means
- Which of the following machine learning techniques is used in the detection of diabetic retinopathy?
    - The CNN architecture, named inception-v3
- Which of the following is not used in the deep network approach to relative aesthetics tasks by Gattupalli et. al?
    - A dataset without labels

Video-Based Inference

Going from Image to Video

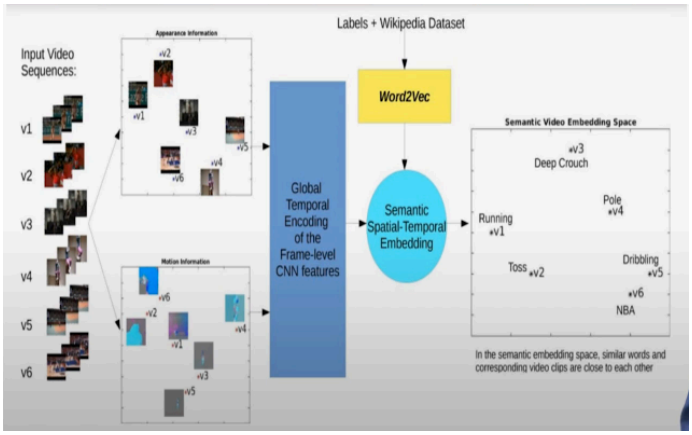
- Processing each form of a video as an independent image and then aggregating the frame-level results
- Extracting spatiotemporal features and an inference task will be based on such features



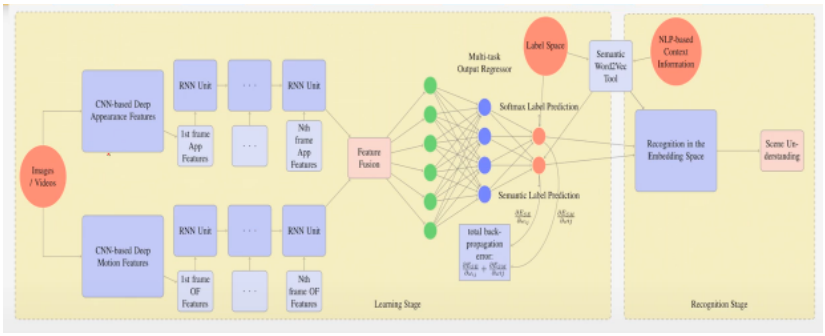
Video2Vec: Sample Applications

- We examining a deep learning approach for finding video representations that naturally encode spatial temporal semantics
    - Mostly based on the following papers:
        - Yikang Li, Sheng-hung Hu, Baoxin Li, “Recognizing Unseen Actions in a Domain-Adapted Embedding Space”, ICIP, Sep 2016
        - Yikang Li, Sheng-hung Hu, Baoxin Li, “Video2Vec: Learning Semantic Spatio-Temporal Embeddings for Video Representations”, ICPR, Dec 2016

Video2Vec Deep Learning Model: Key Idea



Video2Vec Deep Learning Model: Implementation



- A two-stream CNN for extracting appearance and optical flow features
- RNNs for further global spatial-temporal encoding
- A MLP for final semantic embedding space

Applications of the Model

- Visual tasks:
    - Video Action Recognition
    - Zero-Shot Learning
    - Semantic Video Retrieval
- Dataset: UCF101 dataset(13320 video clips from 101 categories; training/testing ratio is 7:3; the split list is provided by its own web)

Additional Implementation Details - 1/4

- Pretraining for the component models:
    - Pretrained Spatial CNN Mode: VGG-f trained on ImageNet
    - Pre-trained OF CNN Mode; Flow-net trained on UCF Sports
    - Pre-trained Word2Vec Model: Wikipedia corpus contained 1 billion words

Additional Implementation Details - 2/4

- Deep model parameter settings:
    - CNNs: Pretrained model + last layer (fc7) features (dimension: 4096x1)
    - RNNs: Hidden layer size is 1024x1
    - MLP: Input layer size (2048x1), hidden layer size (1200x1), output layer size (500x1)
- Loss function:
    - Hinge loss function for semantic embedding
    - Softmax loss function for fine-tuning and classification

Additional Implementation Details - ¾

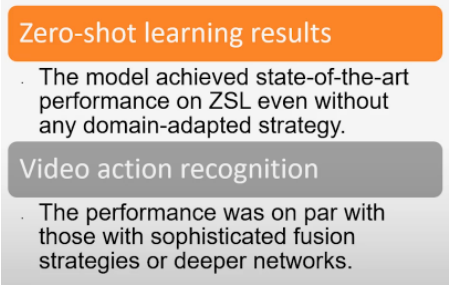
- Video processing settings:
    - Dense Optical Flow and RGB frames are extracted at 10fps
    - Building Video Sequence Mask for each training batch to make each sequence the same length

Additional Implementation Details - 4/4

- Training parameter settings:
    - Learning rate: initialized as 0.0001 and reduced by half each 15 epochs
    - Batch size: 30 video clips
    - Margin value for Hinge Loss function:
        - a. For zero-shot learning, 0.4
        - b. For video retrieval and action recognition, 0.55

Summaries of Key Results

- Dataset: UCF101 dataset



Additional Results

- The task is to retrieve videos from training dataset by using query words that never appear in the training stage but share some information with training labels
- The results show the top 10 retrieval video clips among video dataset

Query Labels	Top10 Retrieve Results	Query Labels	Top10 Retrieve Results
NBA	Basketball Dunk (10)	Extreme	Rock Climbing Indoor (5), Uneven Bars (2), Soccer Juggling (2), Pole Vault (1)
Orchestra	Playing Cello (9), Playing Piano (1)	Tide	Cliff Diving (4), Surfing (2), Throw Discus (2), Sky Diving (1), Rafting (1)
Army	Military Parade (10)	India	Playing Tabla (4), Playing Sitar (2), Head Massage (1), Cricket Shot (1), Mixing (1)
Music	Playing Sitar (9), Playing Piano (1)	Celebrate	Military Parade (6), Long Jump (1), Band Marching (1), Ice Dancing (1), Blowing Candles (1)
Computer	Typing (10)	Home-run	Baseball Pitch (5), Basketball Dunk (3), Field Hockey Penalty (1), Frisbee Catch (1)
Park	Biking (9), Golf Swing (1)	Bout	Kayaking (4), Rafting (2), Rowing (2), Cliff Diving (1), Push Ups (1)
Summit	Cliff Diving (7), Skiing (2), Rope Climbing (1)	Toy	Yo-yo (4), Nun chucks (4), Pull Ups (1), Juggling Ball (1)
School	Skate Boarding (10)	Snow	Skiing (2), Ice Dancing (2), Cricket Bowling (1), Pole Vault (1), Blowing Candles (1), Blow Dry Hair (1), Rafting (1), Sky Diving (1)
Park	Biking (9), Golf Swing (1)	Acrobatics	Juggling Balls (5), Soccer Juggling (5)
Water	kayaking (10)	Ocean	Cliff Diving (4), Sky Diving (3), Kayaking (2), Rafting (1)
FIFA	Soccer Penalty (8), Soccer Juggling (2)	Hurl	Throw Discus (2), Mopping Floor (2), Baby Crawling (1), Javelin Throw (1), Cricket Shot (1), Blowing Candles (1), Pull Ups (1)
Club	Golf Swing (8), Soccer Juggling (2)	Hiking	Biking (5), Kayaking (4), Rafting (1)
Nature	Tai Chi (7), Hammering (2), Walking with Dog (1)	Swim	Diving (5), kayaking (3), Cricket Bowling (1), Sky Diving (1)
Beethoven	Playing Cello (8), Playing Violin (2)	Jogging	Biking (5), Skate Boarding (2), Soccer Juggling (1), Skiing (1), Ice Dancing (1)
Classical	Playing Cello (7), Playing Violin (3)	Foam	Blowing Candles (7), Pull Ups (1), Rope Climbing (1), Juggling Balls (1)
Yankees	Baseball Pitch (10)	Hip-hop	Trampoline Jumping (6), Swing (4)
Duel	Boxing Punching Bag (8), Punch(2)	Scramble	Pull Ups (6), Trampoline Jumping (2), Rope Climbing (1), Cricket Shot (1)
Lifting	Body Weight Squats (4), Rope Climbing (4), Pull Ups (2)	Mat	Rope Climbing (4), Pommel Horse (3), Trampoline Jumping (2), Javelin Throw (1)
Martial	Fencing (3), Archery (3), Boxing Punching Bag (3), Balance Beam (1)	Parachuting	Diving (6), Cricket Bowling (2), Hand Stand Walking (1), Sky Diving (1)
Tumbling	Trampoline Jumping (8), Throw Discus (1), Frisbee Catch (1)	Hunting	Horse Riding (3), Kayaking (3), Nun chucks (3), Frisbee Catch (1)

Knowledge Check

- What makes the classification of images a better solution than the Naive solution?
    - Using spatio-temporal features
- Which of the following information types are used in global temporal representation of the video data? (select all that apply)
    - Appearance information
    - Motion information
- Which of the following is the key idea in the construction of the semantic video embedding space?
    - Associating semantic words with vector representations of the video data

General Adversarial Networks (GANs)

GANs

- Proposed in 2014 by Goodfwllow *et al.*
- An architecture with two neural networks gaming against each other
    - One attempting to learn a *generative model*
- Many variants have been proposed since the initial model

Discriminative vs Generative Models - 1/4

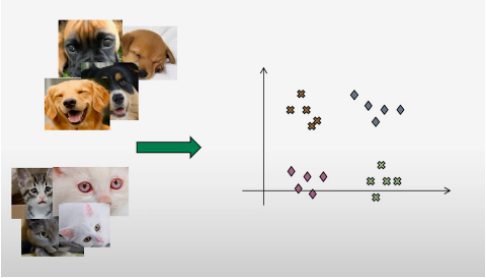
- Discriminative models: E.g., the familiar MLP
  - Given $\{(x_i, y_i)\}$, to learn $P(y_i|x)$
- More generally, we try to learn a posterior distribution of y given x, $p(y|x)$
  - Usually reduced to posterior probabilities for classification problems
- See also earlier discussion on Naive Bayes vs Logistic Regression

Discriminative vs Generative Models - 2/4

- Generative models think the other direction: how to generate x given y
  - E.g., x_i =? if $y_i = 2$?
- More generally, we try to learn a conditional distribution of x given y, $p(x|y)$

Discriminative vs Generative Models - 3/4

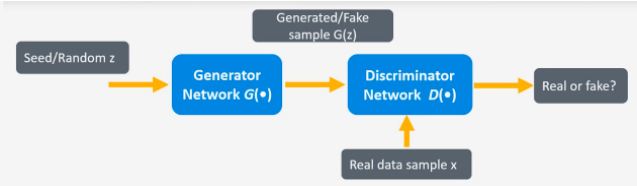
- Illustrating the ideas



Discriminative vs Generative Models - 4/4

- Estimating $p(x|y)$ (or, in general any $p(x)$, if we drop y by assuming it is given)
  - Explicitly density estimation: assuming some parametric or non-parametric models
  - Implicit density estimation: learn (essentially equivalent) models that may create good samples (as if from the "treu" model), without explicitly defining the true model
    - GAN is such an approach

Basic GAN Architecture



- Objective of the Discriminator Network:
  - making D(x) → 1, D(G(z)) → 0
- Objective of the Generator Network:
  - making D(G(z)) → 1

Basic GAN Training Algorithm

for number of training iterations **do**
  for k steps **do**
    Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise distribution $p_g(z)$
    Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data distribution $p_{data}(x)$
    Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$$

  end for
    Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise distribution $p_g(z)$
    Update the generator by descending its stochastic gradient

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)})))$$

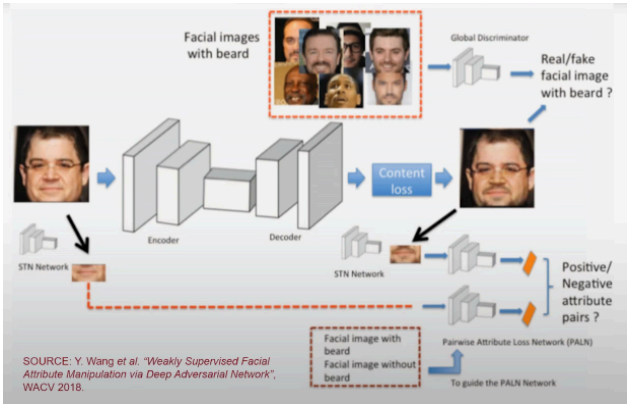
end for

- θ_d and θ_g are the parameters of the discriminator and generator respectively

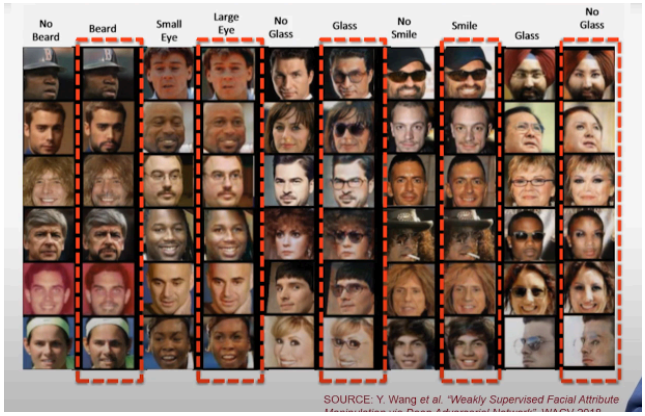
Applications of GAN

- GAN enabled may novel/interesting/fun applications
- Many GAN-based models have been proposed, following the initial paper
- Consider one example: Facial attribute manipulation
  - - Y. Wang *et al.* "Weakly Supervised Facial Attribute Manipulation via Deep Adversarial Network", WACV 2018.

Facial Attribute Manipulation - 1/2



Facial Attribute Manipulation - 2/2



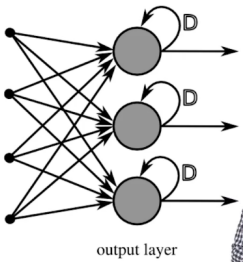
Knowledge Check

- Which of the following is the key architectural difference between GANs and the other NNs?
  - GANs are based on a game-theoretic scenario
- Which of the following is a sample of a discriminative model?
  - Given the dataset representing dog images, predict the probability that the data belong to a certain category of dogs
- How many different networks form the basic GAN architecture?
  - 2

Industry Perspective Wrap-Up

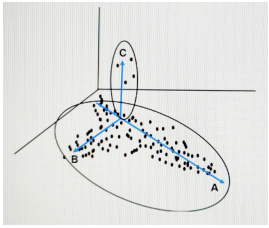
Time Series Forecasting

- DeepAR
  - Cold-start forecasting
  - Probabilistic forecasting



Module 7 Part 1 Quiz Questions

- Which of the following statements is wrong for the PCA?
  - The PCA algorithm is based on gradient search
- Which of the following gives the total variance of a given dataset, after we do PCA on the dataset?
  - the sum of all eigenvalues
- True or False: The principal components that capture the most information of the data are those that correspond to the largest eigenvalues.
  - True
- Which of the following is NOT true for dimensionality reduction?
  - Dimensionality reduction slows down the training of the machine learning systems
- In the attached figure, which vector is the best candidate for the first principal components.

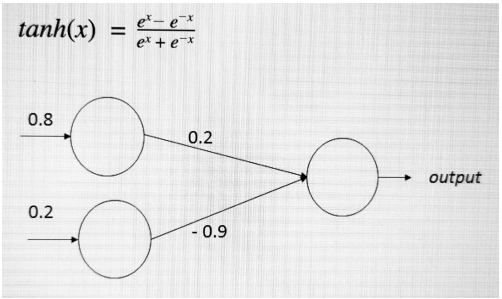


  - Vector A
- The first principal components for a dataset is found to be on the direction given by vector U= (4,2). Which of the following may be the direction of the second principal components? (Select all that apply)
  - V=(-2,4)
- Which of the following is true for the PCA? (Select all that apply)
  - Any two principal components are orthogonal to each other
  - PCA may be used for dimensionality reduction by keeping only a small number of principal components
  - The length of each principal components vector is equal to 1
- Which of the following is one way to find out whether the dimensionality reduction algorithm performs well?
  - Checking if the reconstruction error is small enough
- Which of the following is a motivation for dimensionality reduction? (Select all that apply)
  - Visualizing the data gaining insights on the most important features
  - Reducing computational complex
  - Removing redundant features
  - Saving space

- Which of the following is true about PCA?
    - The first principal components gives the direction along which the data points have the largest variance

Module 7 Part 2 Quiz Questions

- Which of the following created the first image classification application that classified images with at least 80% accuracy?
    - AlexNet
- The idea of Video2Vec is similar to which of the following?
    - Word2Vex
- Which of the following is not a training parameter used in the Video2Vec deep learning model?
    - Optical Flow (OF) feature extraction algorithm
- Which of the following is NOT part of the basic GAN architecture?
    - Autoencoder
- Which of the following is NOT true for the basic GAN algorithm?
    - For each step, update the discriminatory by descending its stochastic gradient.
- The figure I attached shows a simple neural network. An observation with two variables (5,7) is presented to the network as shown. What is the updated weight if linear function y=x is used as the activation function and the Mean Squared Error as the error function?
    - (0.91, 0.83)
- Which of the following is NOT true for pooling in CNNs?
    - It is a method for increasing the number of features for the next layer
- Which of the following is NOT true for autoencoders?
    - The number of neurons in the hidden layers must be less than the number of inputs
- The figure I attached shows a simple neural network. An observation with two variables (0.8, 0.2) is presented to the network as shown. What is the predicted output from the neural network using a tanh activation function?



    - $\frac{e^{-0.02} - e^{0.02}}{e^{-0.02} + e^{0.02}}$

–

-

Knowledge Check

- —

Module X Quiz Questions

- —