Lecture Notes

Supervised Learning

Lesson 1: Decision Trees

Lesson 2: Regression & Classification

Lesson 3: Neural Networks

Lesson 4: Instance Based Learning

Lesson 5: Ensemble – Bagging & Boosting

Lesson 6: Kernel Methods - SVMs

Lesson 7: Computational Learning Theory

Lesson 8: VC Dimensions

Lesson 9: Bayesian Learning

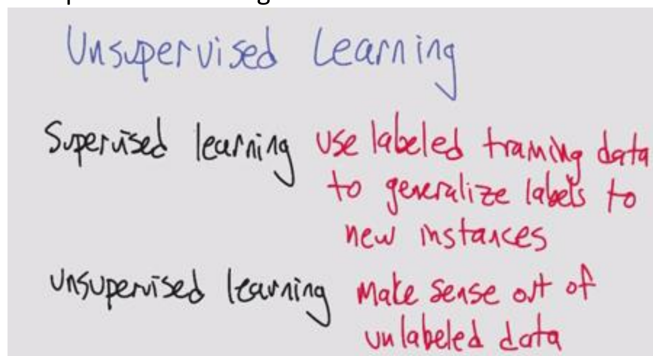
Lesson 10: Bayesian Inference

Unsupervised Learning

Lesson 1: Random Optimization

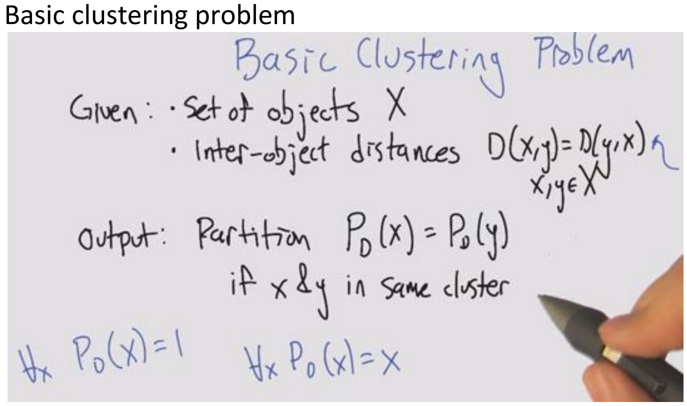
Lesson 2: Clustering

Unsupervised learning



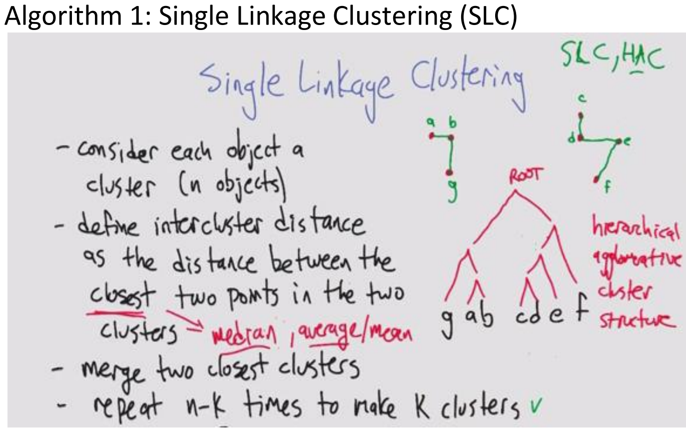
- One of the classic unsupervised learning problems is the Clustering Problem
    - It doesn't have to be an actual distance, just has to be a similarity (like in kNN)

Basic clustering problem



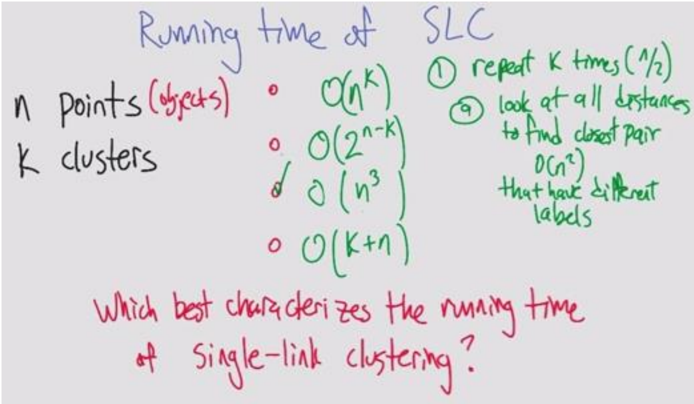
- Trivial clusters: we can all be humans (one big cluster); we can all be ourselves (one cluster per instance)

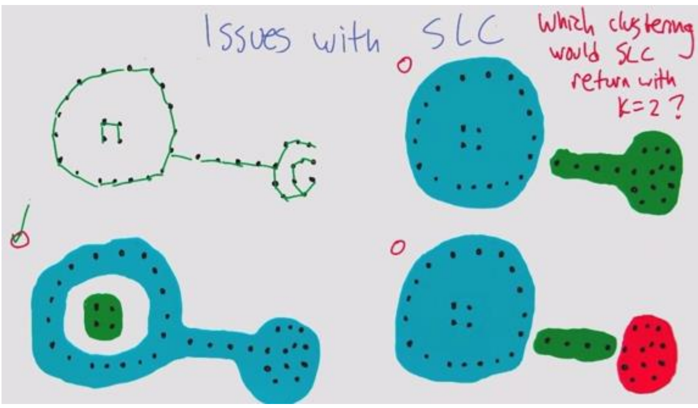
Algorithm 1: Single Linkage Clustering (SLC)



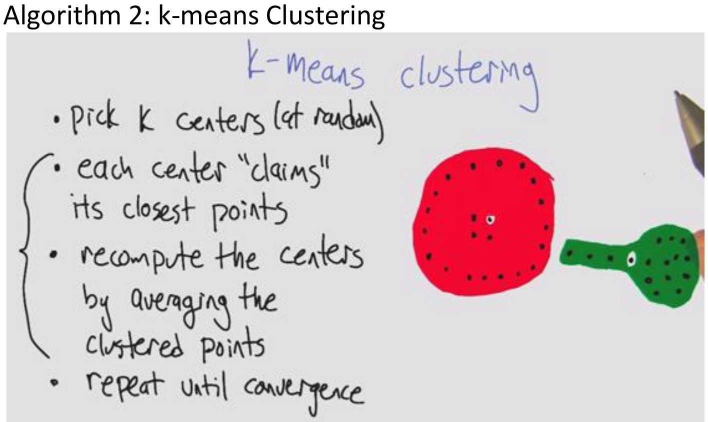
- SLC: single linkage clustering
- HAC: hierarchical agglomerative cluster

Running Time of SLC

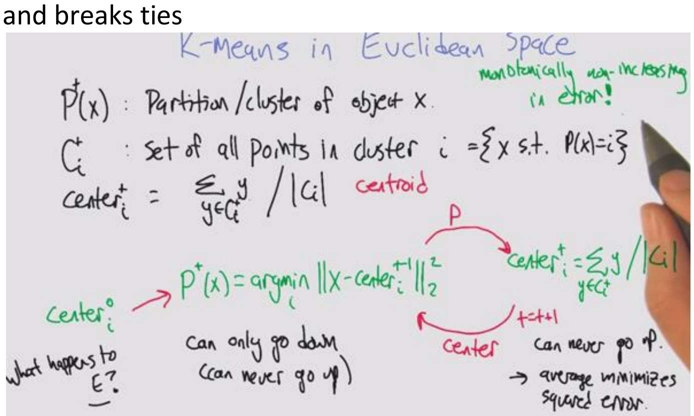




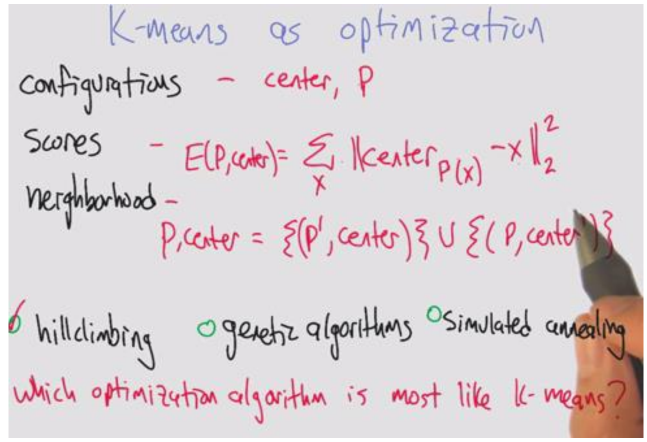
Algorithm 2: k-means Clustering



k-means Euclidean space (Part 1,2). Error does not increase, and breaks ties

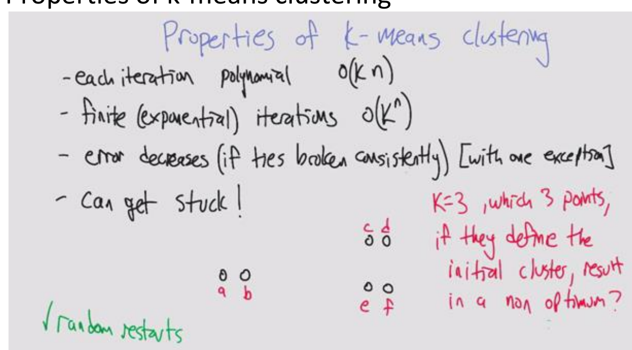


K-means as optimization. Errors as scores. Like hill climbing.



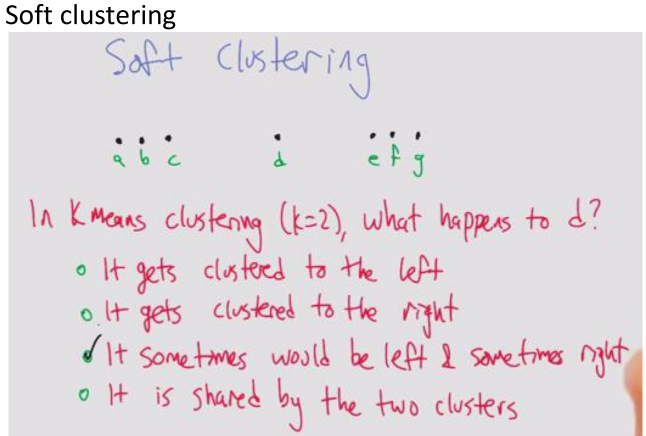
- Monotonically non-increasing in error converges because there are a finite number of objects, even though it's an infinite space
- You may have a point that goes between two partitions equally, so you need a way of breaking ties
  - Tendency to go to smaller cluster

Properties of k-means clustering

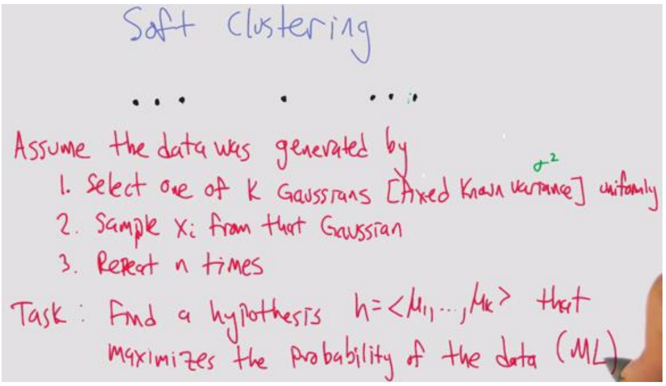


- How do you avoid non-optimum clustering?
  - Random restarts
  - Choosing centers that are furthest apart

Soft clustering

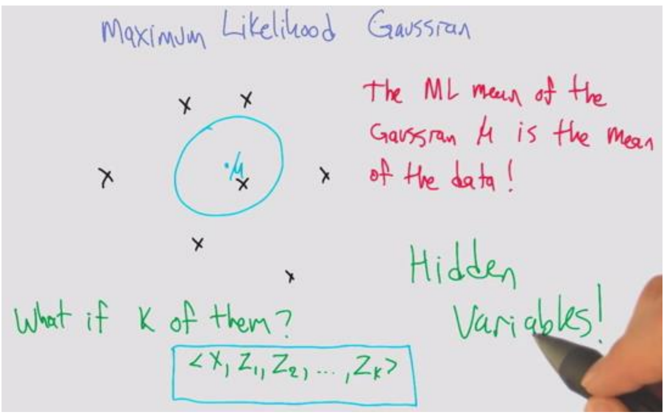


- Example of how the initial centers affects the end convergence
  - D would end up on the right if you start with a/b center, left if you start with f/g and would depend on tie breaking if a/g



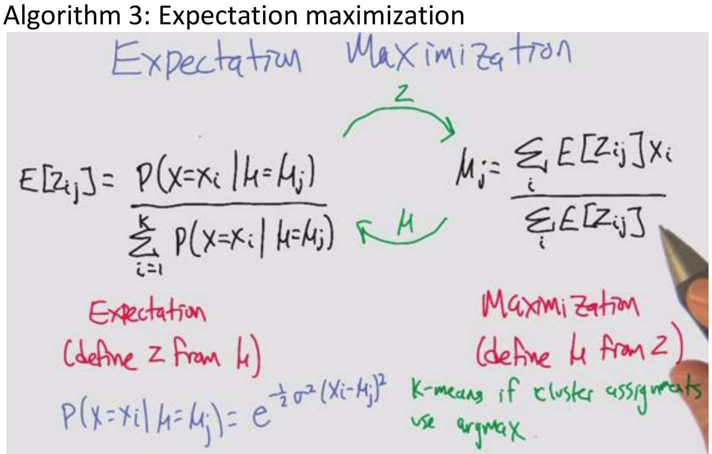
- Thinking “bayesianly”, given the data we’re going find what the clusters would have been to generate the data

Maximum likelihood Gaussian



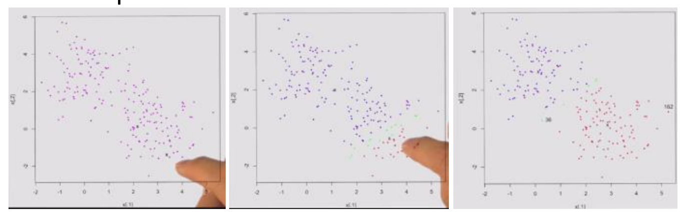
- Using hidden variables to break up the problem in a convenient way
- X lies in one of the clusters Zn (a bunch of 0s and a 1)

Algorithm 3: Expectation maximization



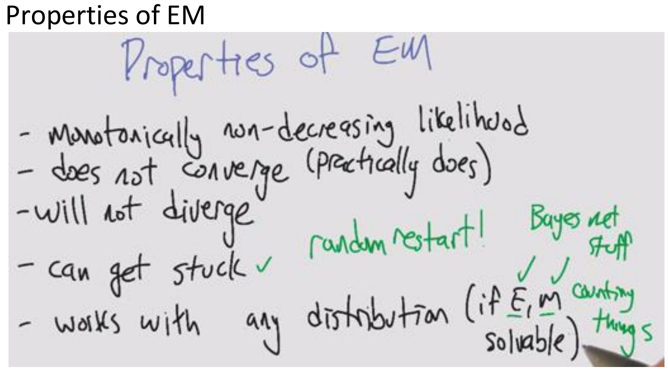
- Similar to k-means at an algorithmic level
- Tick tock between expectation (E)and maximization (miu)
- Maximization allows for partials / weighted average.
- E: Soft clustering, given the miu and x, we can compute how likely it is in z
- M: computing mean, if that is the cluster, we can take the average of the xi within each cluster j, what’s the likelihood it came from cluster j
- This can be kmeans if all the probabilities were 1s and 0s, so the maximization step would just be the means (since it’s just normalizing)
    - Then you have to push the probabilities of being in the clusters 1 and 0s
    - Kmeans if cluster assignments use argmax (1 and 0)

EM Example



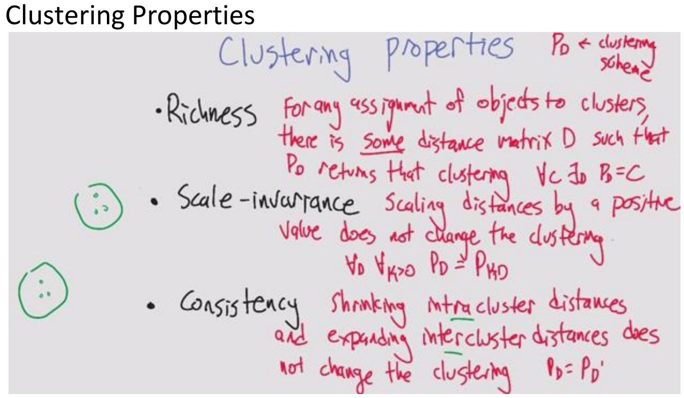
- Pick 2 centers, select 2 points randomly
- Run iteration of EM
    - Expectation
    - Move centers
    - Label the band in the middle that aren’t deeply one or the other
- Not forced to make a decision

Properties of EM

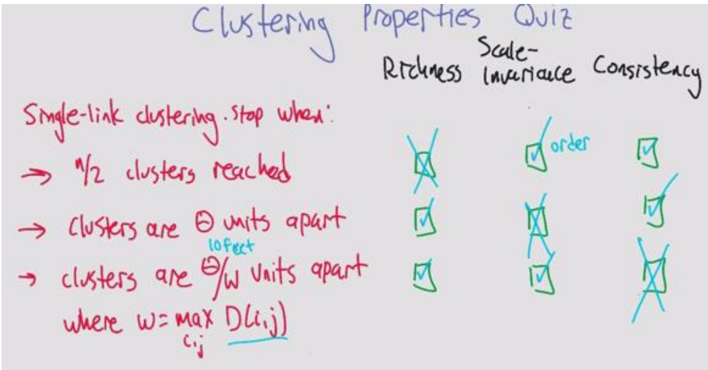


- Sometimes E or M can be hard

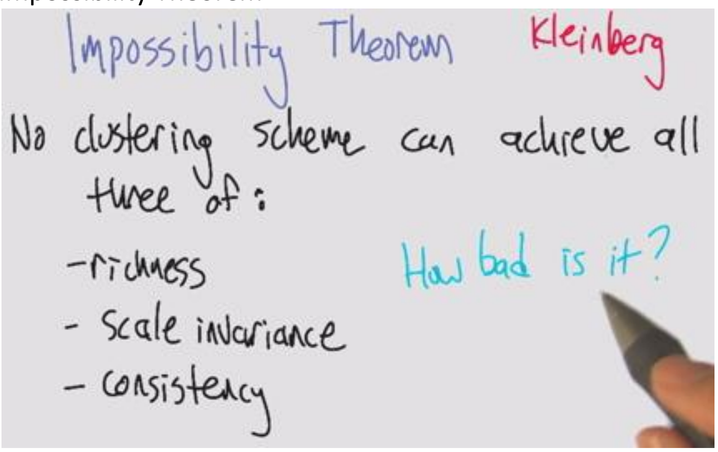
Clustering Properties



- Consistency is harder to visualize
    - If you make the objects in a cluster more similar, and the other ones more dissimilar, the clusters do not change

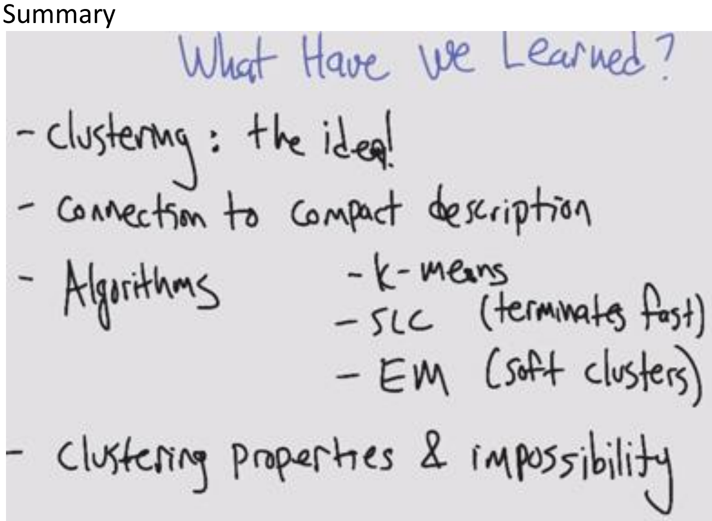


Impossibility Theorem



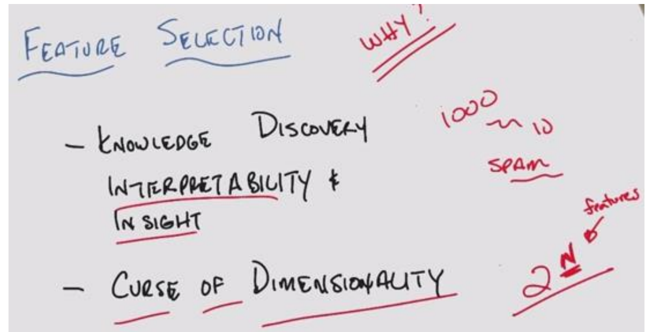
- Defined by Kleinberg
- All 3 are mutually contradictory

Summary



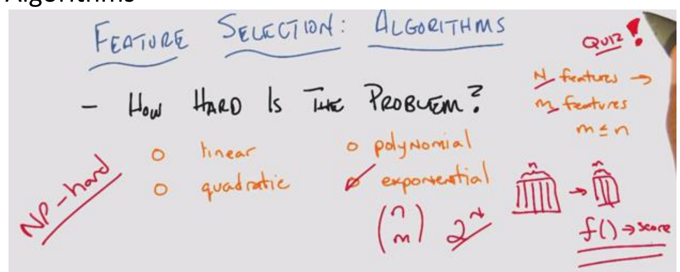
Lesson 3: Feature Selection

Introduction

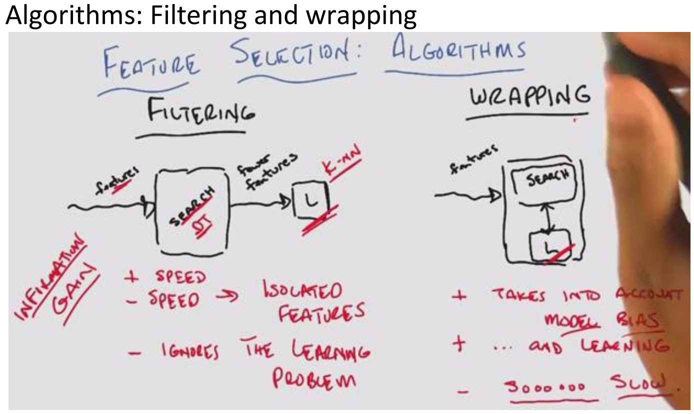


- Feature selection tends to be ignored in the realm of ML

Algorithms

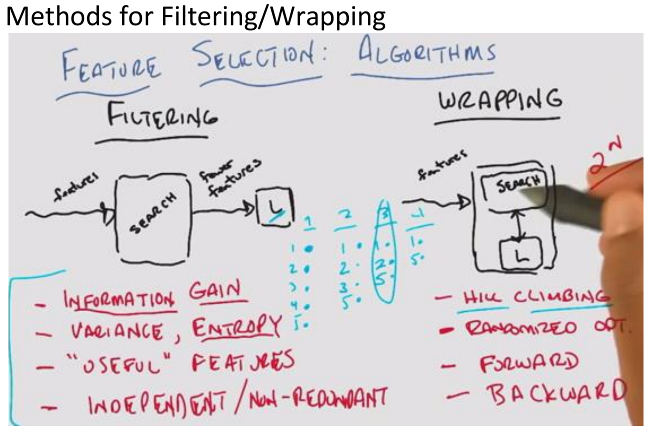


Algorithms: Filtering and wrapping



- Tradeoffs: speed, feedback
- Imagine filtering is like decision trees
- Wrapping requires cross validation, tons of computations

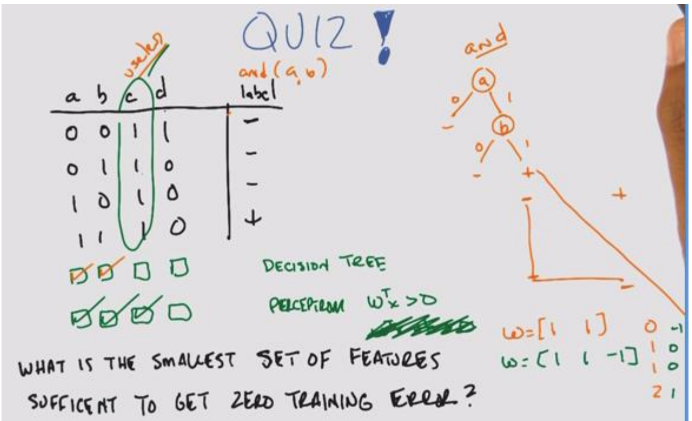
Methods for Filtering/Wrapping



- Forward/backward search. Forward is like hill climbing if you do it one at a time

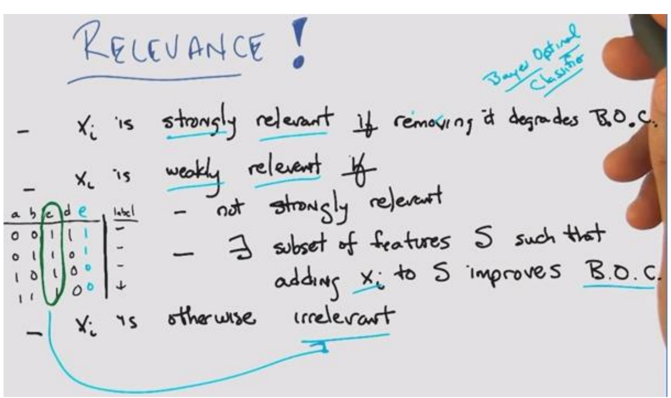
- Forward is like building the team one player at a time, backward is like removing one a time

Zero training error



- Features a, b, c, d. Instances n=4
- DT: Build a decision tree
    - Split on A to see if A gives a positive label
    - Split on B
    - If A AND B are true, output true
    - C is *useless* because it doesn't have info
    - D does not help with +
- Perceptron: Write 2D plane. Without intercept, it's a origin limited perceptron
    - We know A and B are useful from DT
    - Plot A and B on a perceptron graph
        - w [1 1] outputs values of 0, 1, 1, 2
        - But wtx > 0 applies to 3 instances
        - Use a 3rd feature as the intercept
            - B, bias unit
    - Try A and B with C as bias unit
        - w [1 1 -1] outputs -1, 0, 0, 1
        - wtx > 0 applies to just the one +
- C is not useful in DT, but it is for the perceptron.
- You can use information gain, entropy, variance to determine what's relevant

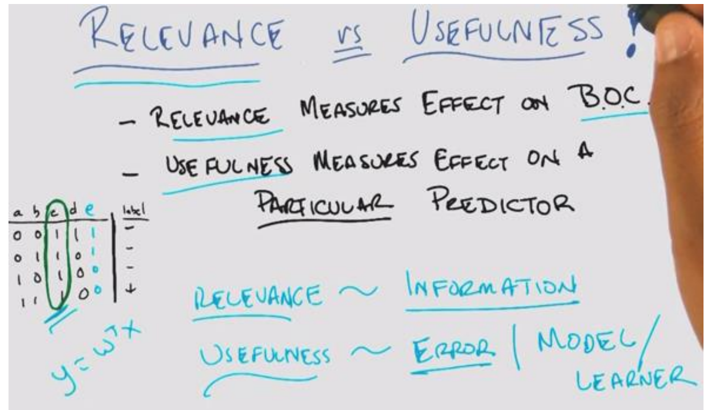
Relevance



- A/B are strongly relevant, C is irrelevant
- With E added
    - A is not E, so they are weakly relevant
    - B is strongly relevant, C is irrelevant

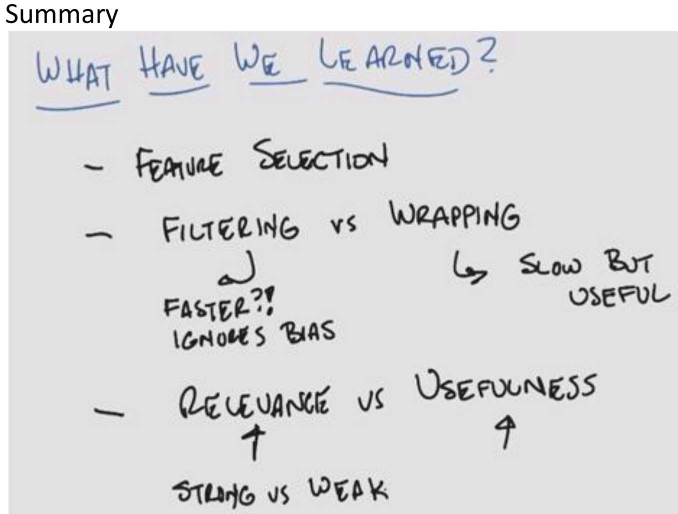
- BOC: Bayes optimal classifier, as discussed earlier in SL (wt avg of all the hypotheses based on the probability they correctly represent the data)
- C is irrelevant, but it was useful in perceptron case

Usefulness



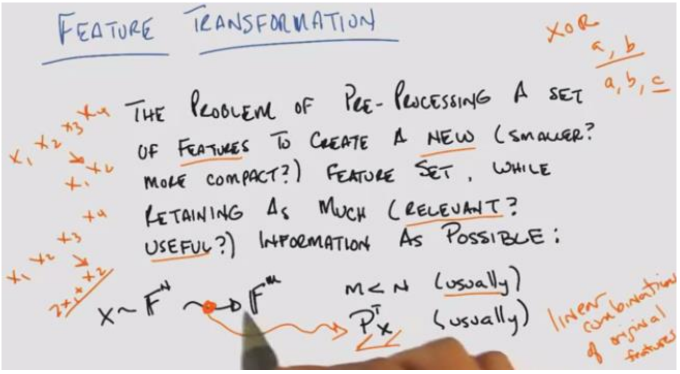
- Look at the cluster changes based on relevance and usefulness

Summary



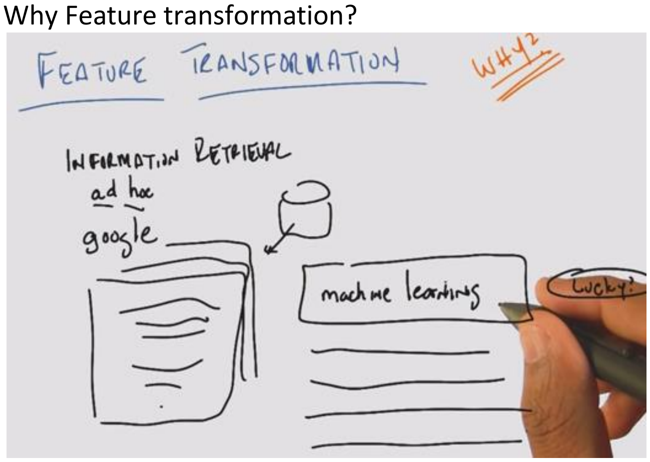
Lesson 4: Feature Transformation

Intro



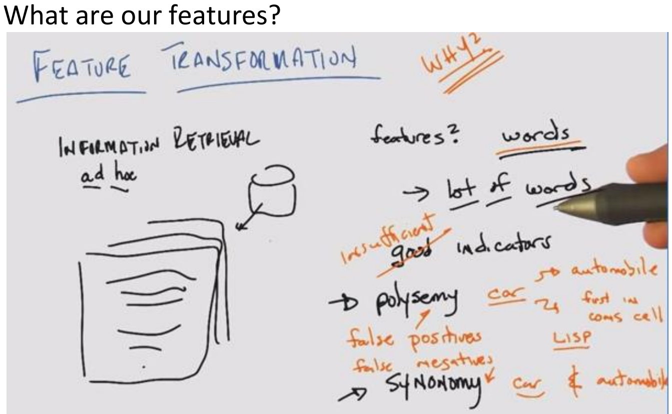
- How is this different from feature selection?
  - Think of this as a subset of feature selection
  - Transformation into a smaller subset, such that you have linear combinations of original features
  - Don't necessarily need to go into fewer dimensions
    - We've done this before in Perceptrons, Kernels, XOR

Why Feature transformation?



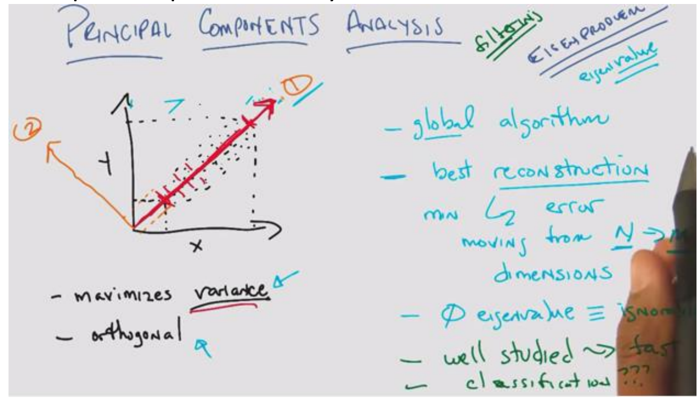
- Example: ad hoc because you don't know what the retrieval is going to be, like Google

What are our features?



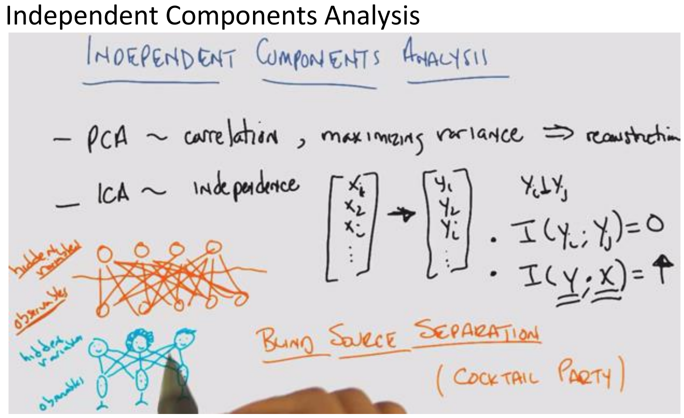
- A lot of words >>> curse of dimensionality
- Words like Tesla
- Polysemy: word w multiple meanings. False pos
- Synonymy: same meaning expressed in different words. False neg

Principal Components Analysis



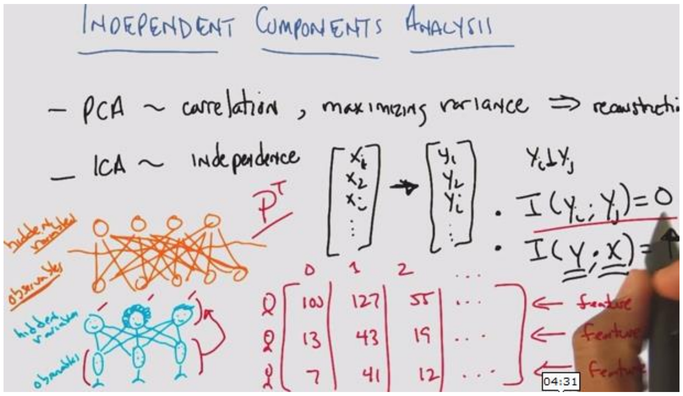
- Projecting 2D data onto x or y axis is an example of feature ***selection***
- Feature ***transformation***: projecting onto a diagonal plane maximizes variance on 1 feature
- PCA
  - 1. Maximizes variance
  - 2. Mutually Orthogonal
- Like filtering
- Reconstruction: You can reconstruct all the original data with the new axes
- Tend to subtract the mean from the data to normalize to the origin, to look for correlation
- Eigenvalue properties

Independent Components Analysis



- PCA ~ correlation, maximizing variance => reconstruction
- ICA ~ maximizing mutual independence
  - Linear transformation such that each new feature is statistically independent from one another yi ⊥ yj
  - Mutual information = I (yi ; yj) = 0
  - Mutual information of the new features and the original features is max (reconstructs well) = I (y . x) high
- Blind source separations (cocktail party)
  - Trying to listen to one person at a party, while cutting out the noise of other conversations
  - Online Demo: http://research.ics.aalto.fi/ica/cocktail/cocktail_en.cgi

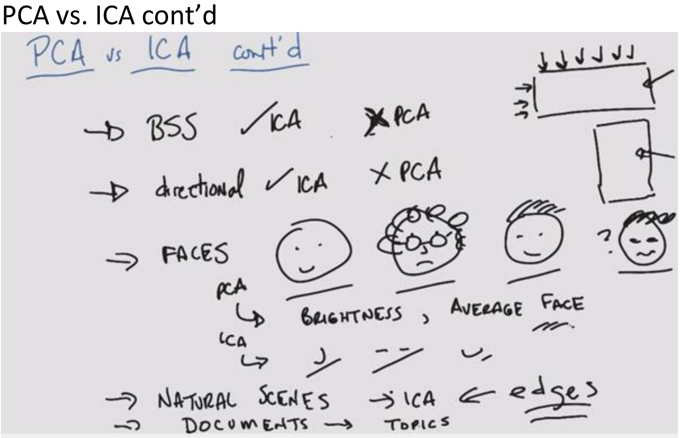
Matrix



PCA vs ICA

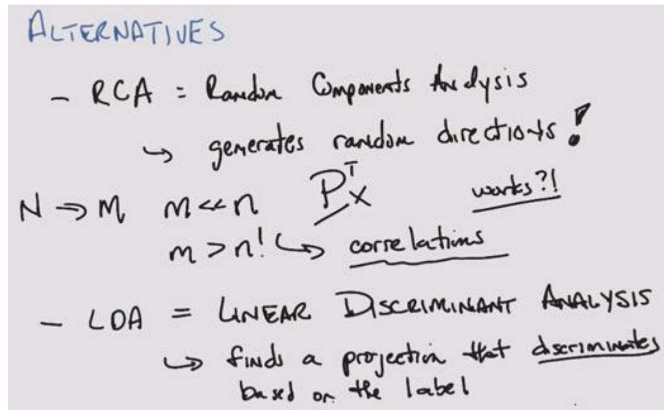
- PCA tends to find things that are uncorrelated, not same as statistically independent
  - Coincidence: When all of the data is Gaussian, it is possible for PCA to both represent data that is mutually orthogonal ***and*** mutually independent
- ICA assuming highly non-normal
  - What if the independent variables are summed together into a linear combination? By central limit theorem, you will get a Gaussian
- ICA: Mutually independent projections, mutual information between old and new
- PCA/ICA fundamental assumptions are different, both trying to represent the data
- Bag of features: Order doesn't matter for ICA

PCA vs. ICA cont'd



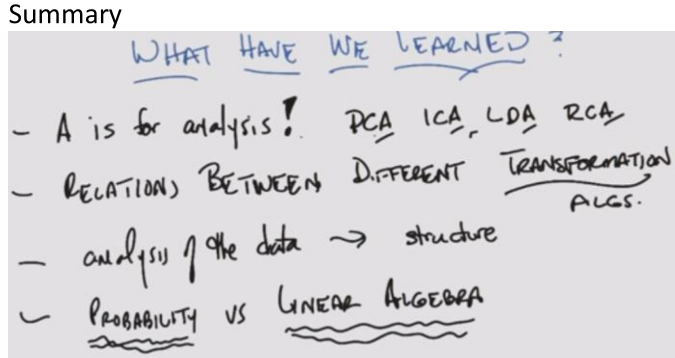
- ICA was designed for BSS (blind source separation problem), whereas PCA does a terrible job
- ICA is directional, whereas PCA finds the same answer whether you give one dimension of a matrix or another
- ICA tends to pull out edges from images, therefore we can use faster algorithms that find edges as a substitute
- Average face = "eigenface"
- PCA tends to find global, ICA tends to find parts of

Alternatives



- RCA: Random components analysis
  - Cheap, easy, fast
  - Simple yet they manage to work
- M tends to be bigger than m in PCA
- Time: ICA>PCA>RCA
  - Feel like filtering, optimize but don't care about the final labels
- LDA is like supervised learning
  - Transformations into clusters based on their label

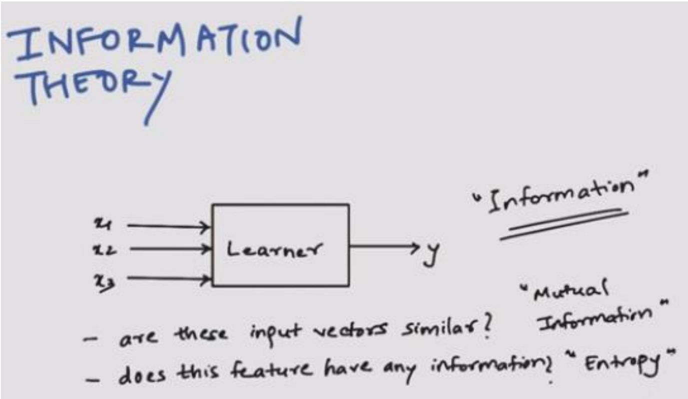
Summary



- ICA: probability. Hard to find, more expensive, but when it does it produces a satisfying answer
- PCA: linear algebra. Well understood, not exactly what you want

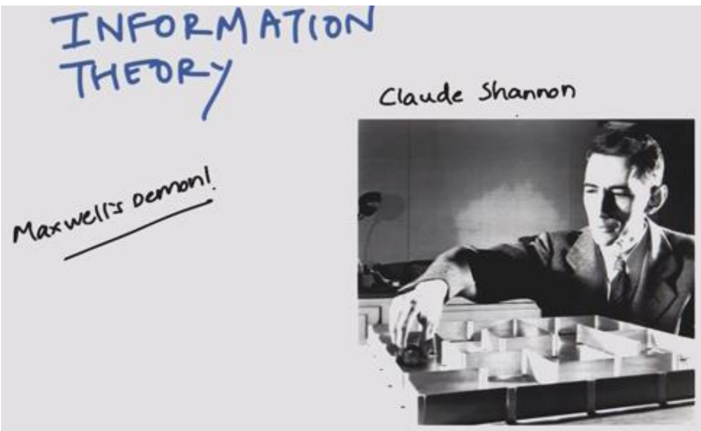
Lesson 5: Information Theory

Intro



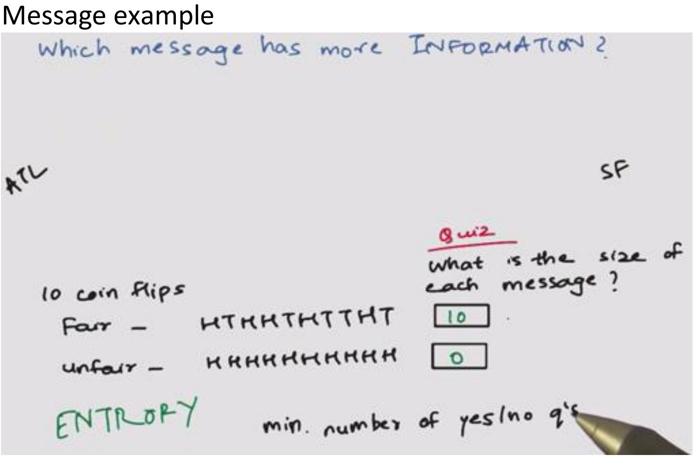
- Mutual information vs. entropy

History

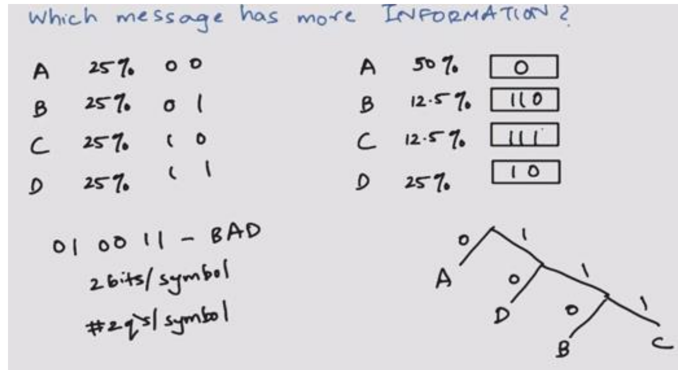


- Claude Shannon and messages
- Maxwell's demon

Message example

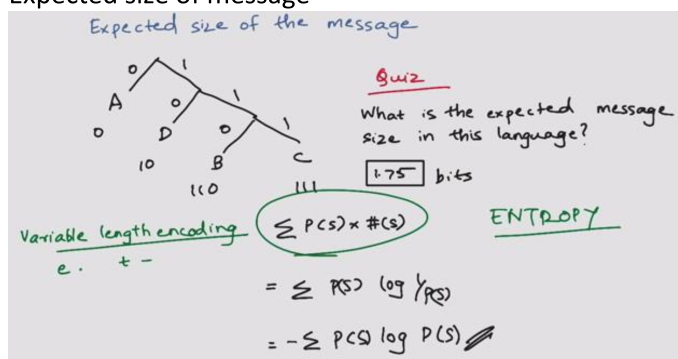


New message example



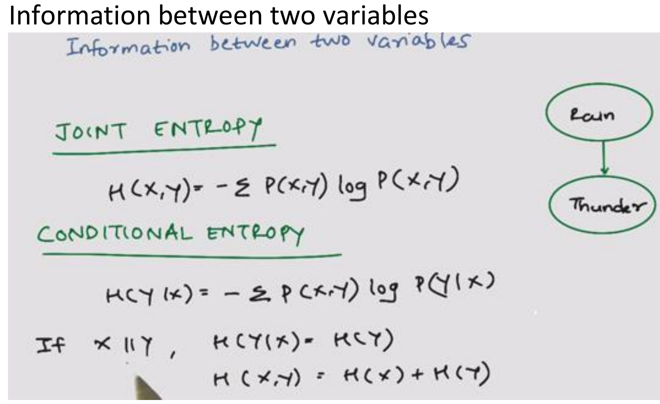
- 25% equal:
  - 2 bits per symbol
  - 2 questions per symbol
- Non-equal

Expected size of message



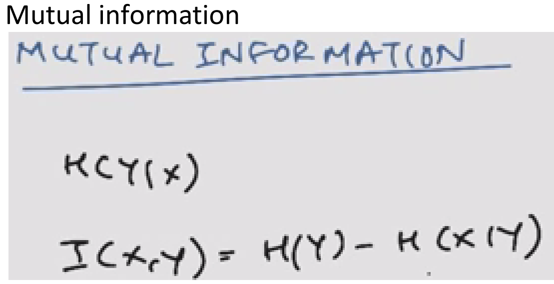
- Variable length encoding, 1.75 bits
- Entropy = #Bits = probability (symbol) * # symbols

Information between two variables

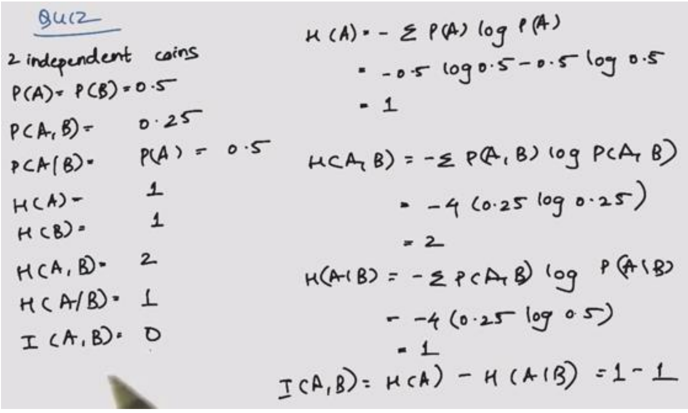


- Joint entropy
- Conditional entropy

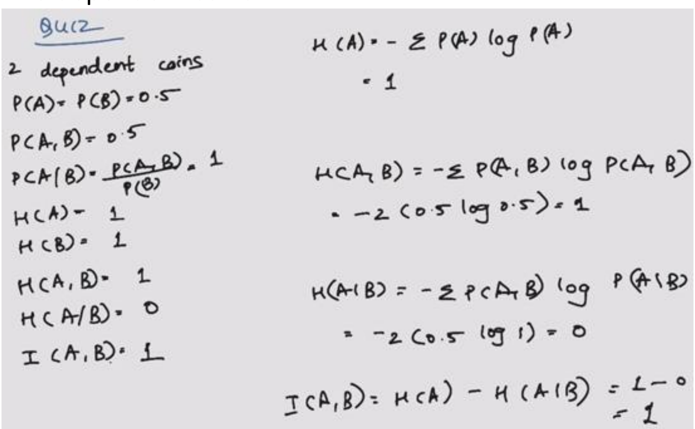
Mutual information



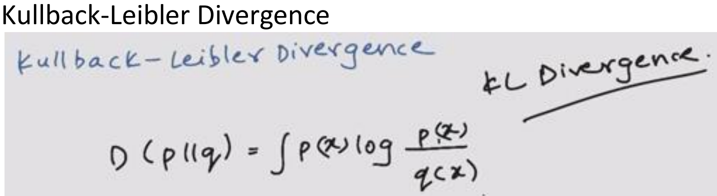
Two independent coins



Two dependent coins

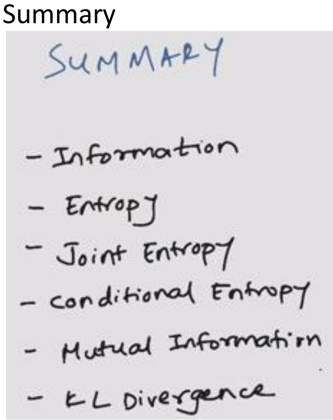


Kullback-Leibler Divergence



- Distance metric that does not follow triangle law
- Substitute to Least Squares

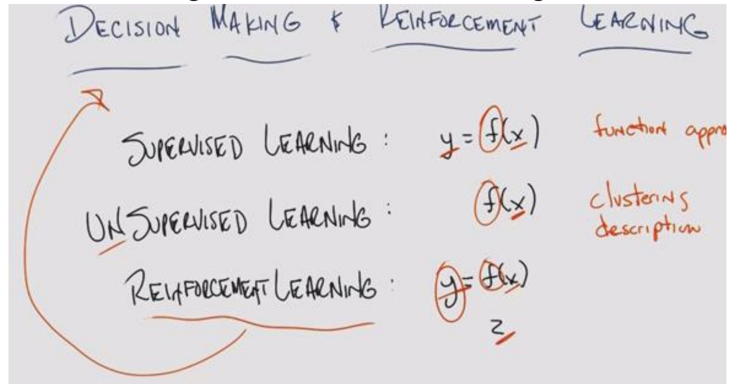
Summary



Reinforcement Learning

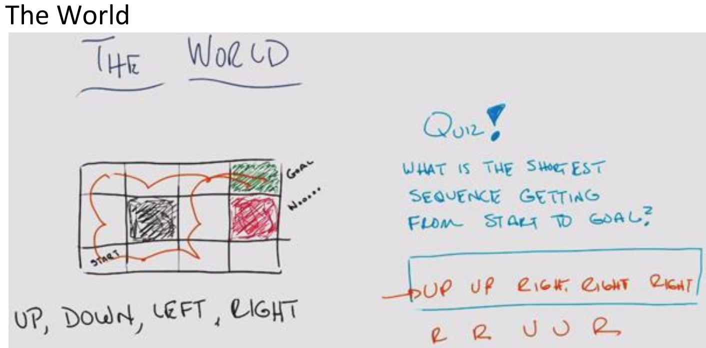
Lesson 1: Markov Decision Processes

Decision Making and Reinforcement Learning

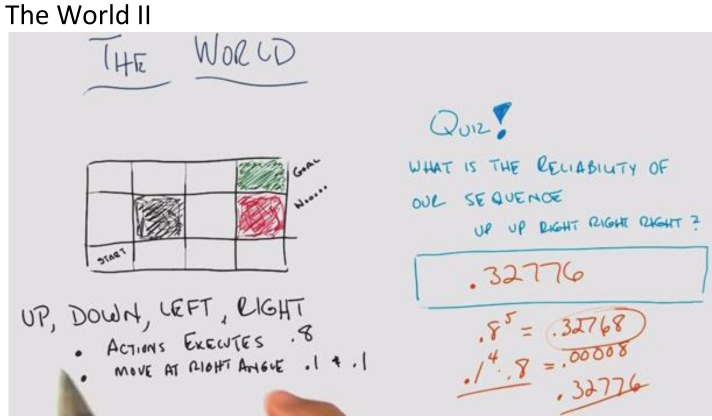


- SL: Given a set of x-y pairs, you are trying to find a function f that will map some new x to the proper y
- UL: Given some x, and trying to find f. Give a compact description of the set of x's
- RL: Looks a lot of SL. Given a string of pairs of data to learn a function. Instead, given x-z pairs to find a function f that generates y
  - X
  - Z
  - One mechanism that is used in decision making

The World

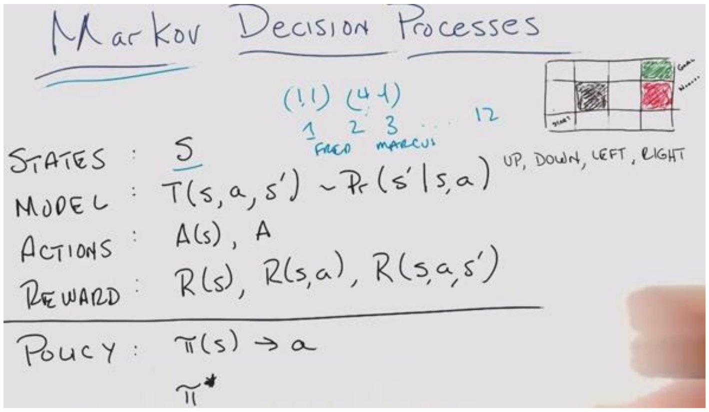


The World II



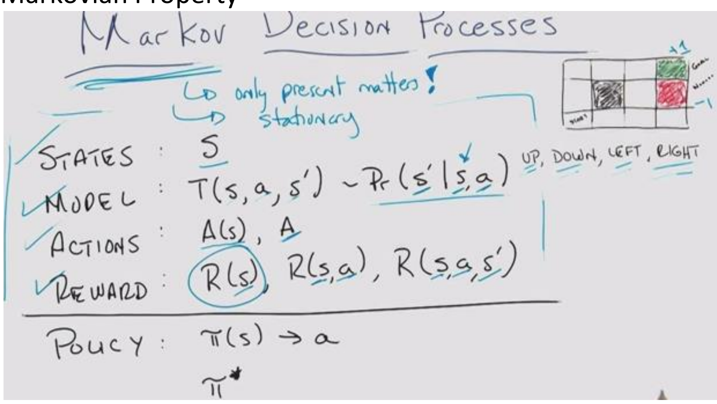
- Plan for when something "goes wrong"

Markov Decision Processes



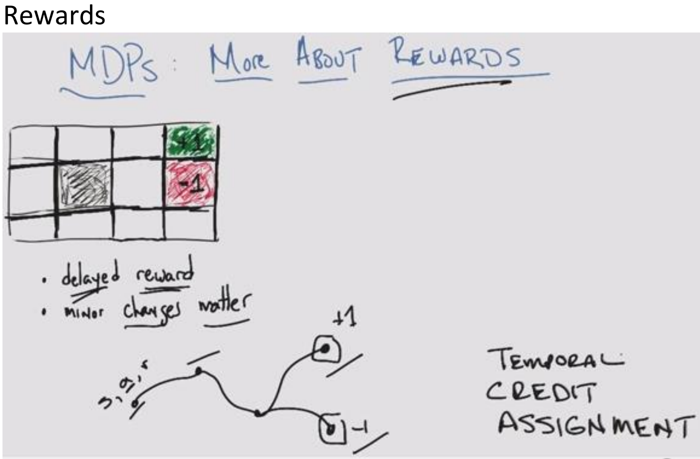
- S: State – grid location
- T: Transition
- Pr: Probability
- A: Action: Up down left right
- R: Reward

Markovian Property



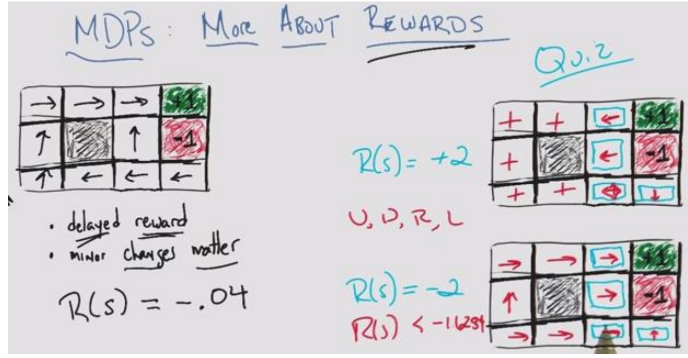
- Markovian: Only the present matters (not the past)
  - You can turn almost anything into a Markovian
- The roles, world, model are stationary
- Rewards represent domain knowledge

Rewards



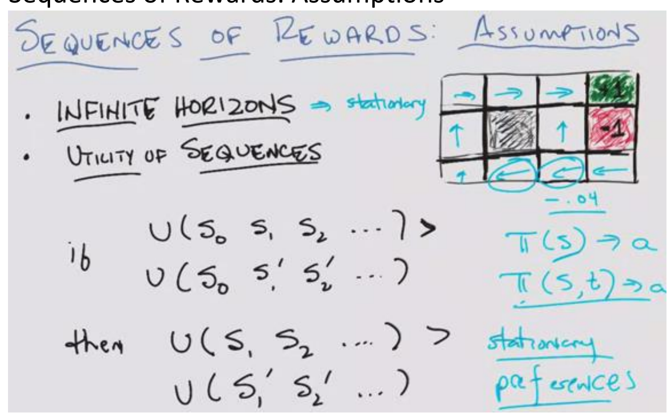
- (Temporal) credit assignment problem
  - Delayed reward, unknown at the time the action was made

Example



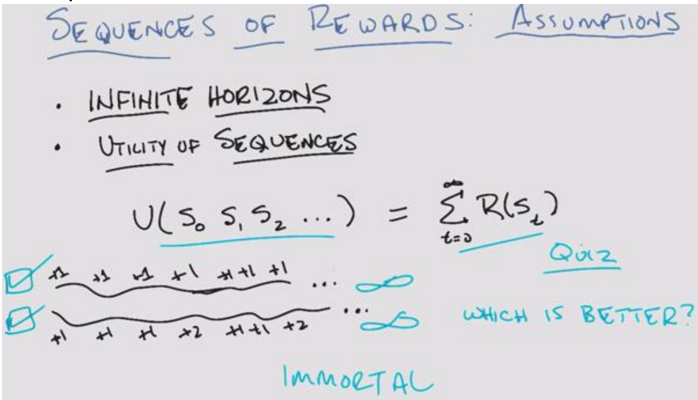
- Reward of -0.04 for all unlabelled states, encourage reaching the goal
- Like walking on hot sand trying to get to the shore
- Minor changes in your rewards matter
- Since this is Markovian, some states don't matter
- Can be solved using expected value

Sequences of Rewards: Assumptions



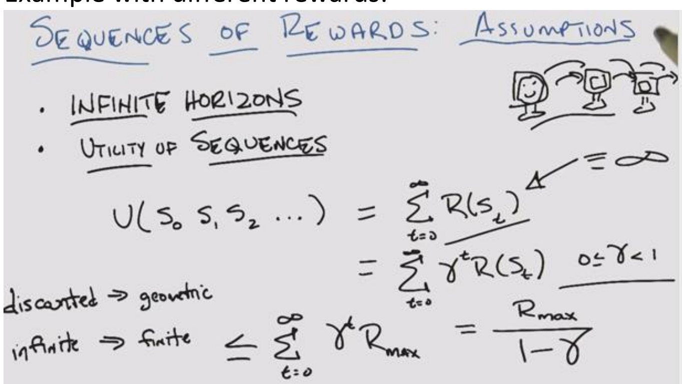
- Infinite horizons: stationary
  - GridWorld example, need to add cost of -0.04 so it ends
  - But also, we assume that the agent is allowed the number of time steps take the long route
  - But if time is limited, then the agent perhaps will take the risky shortcut
  - If you have a finite horizon
    - The policy will change from the converged policy of an infinite horizon
    - But ***also*** the policy will change in time, even for the same state
      - Policy: account for state AND time (not for this course)
- Utility sequences (prior states don't matter)

Example



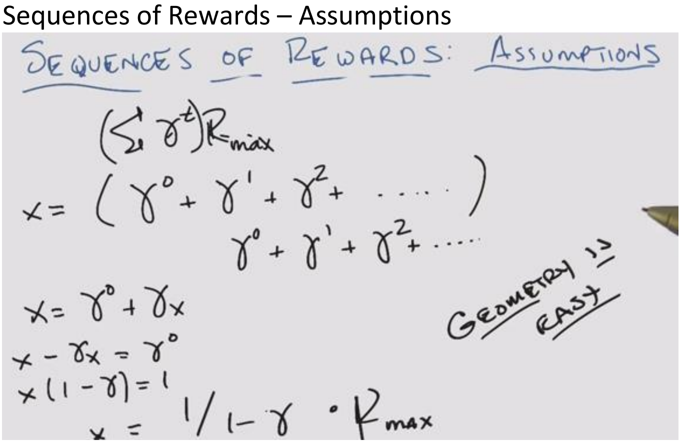
- Given this reward scheme, both approach infinity, so neither is better!

Example with different rewards:

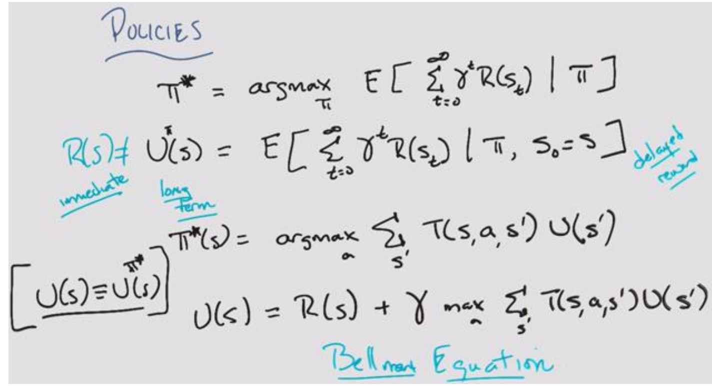


- Discounted rewards brings infinite > finite

Sequences of Rewards – Assumptions

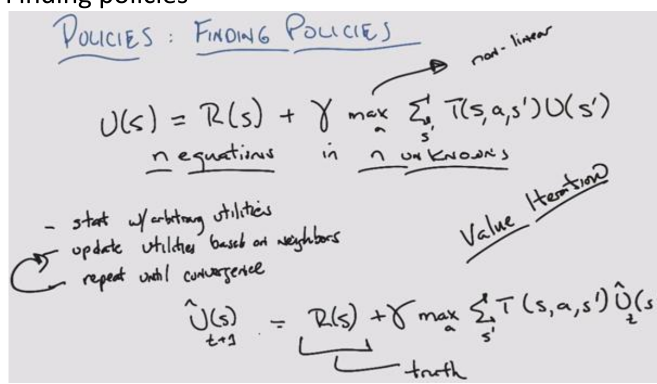


Policy



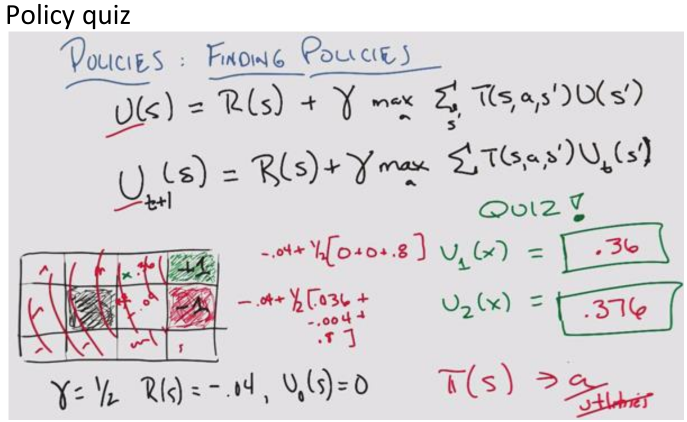
- Reward != utility
- Assume ”true” utility of the state
- Bellman Equation: the key recursive equation in MDP

Finding policies



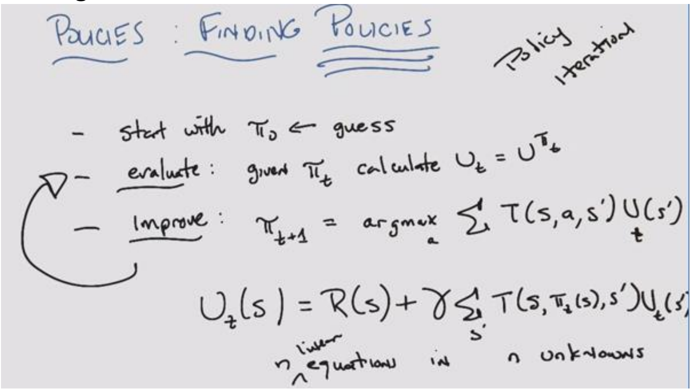
- Adding to arbitrary utility, more and more truth, and estimate is discounted. Will converge
- Value (utility) iteration

Policy quiz



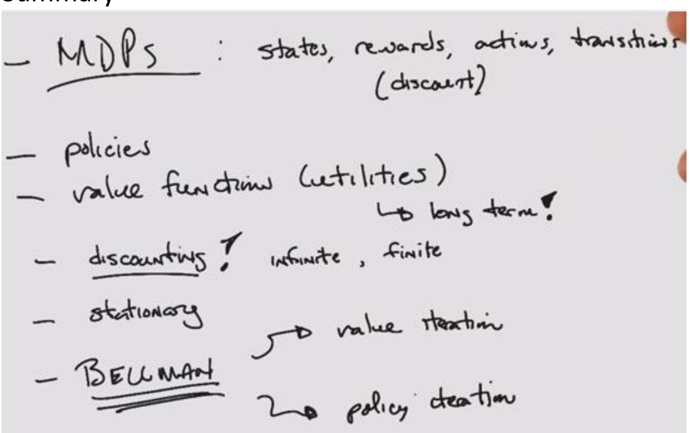
- Policy is like classification, utility like regression

Finding Policies 3



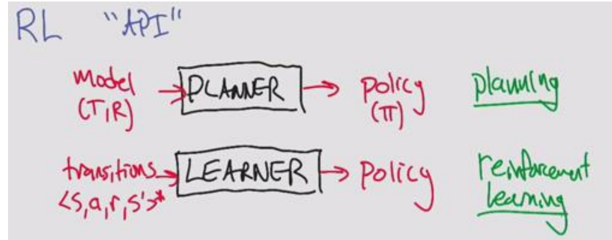
- Max was non-linear, but now this is linear
- Policy iteration (as opposed to utility iteration seen before)
- Guaranteed to converge

Summary



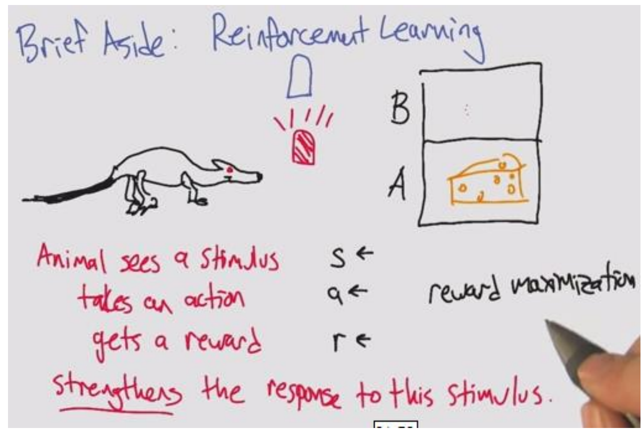
Lesson 2: Reinforcement Learning

Application Program Interface (API)



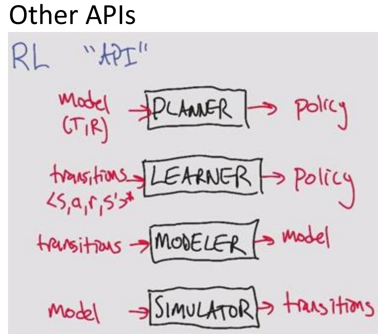
- Compute a policy vs. learn a policy

Rat Dinosaur



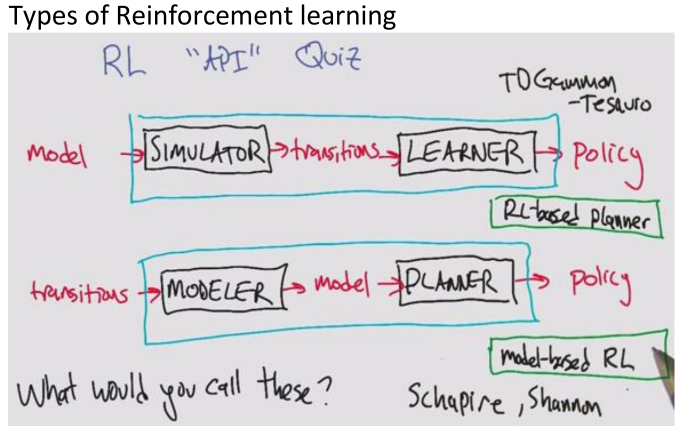
- Reinforcement learning comes from psychology
- Computer science does not focus on strengthening. Focus on reward maximization

Other APIs



- Can be linked together to create different types of policies

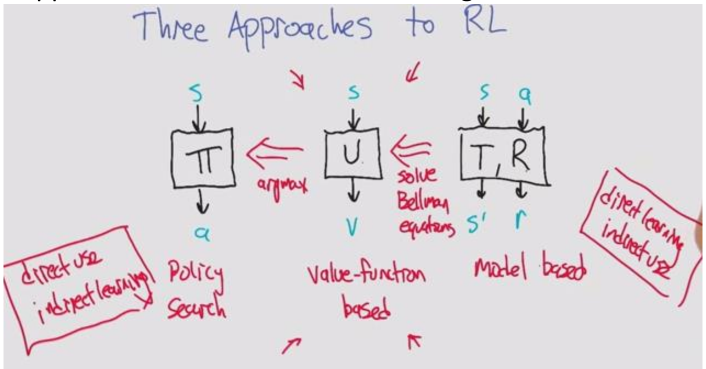
Types of Reinforcement learning



- Schapire: Boosting

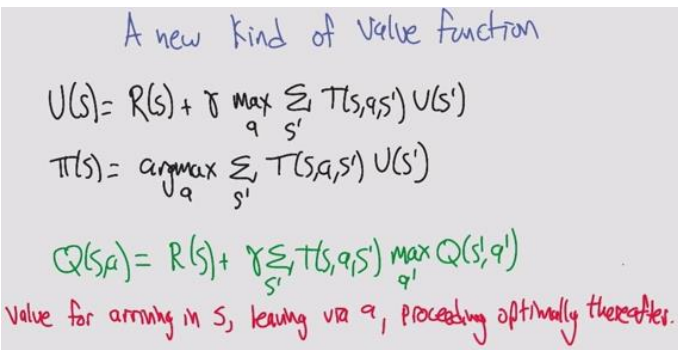
- Shannon: Information Theory

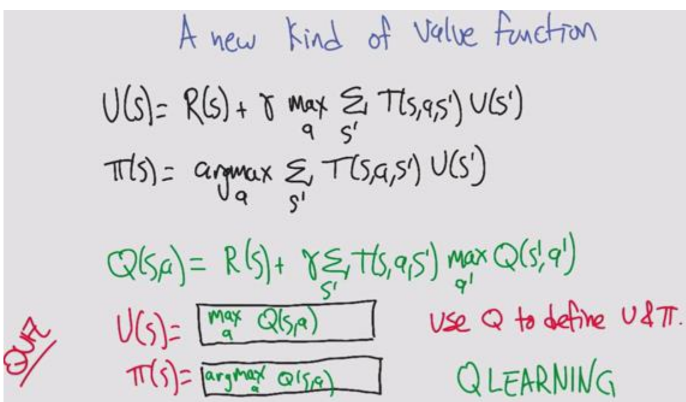
3 approaches to reinforcement learning



1. Policy search: Temporal credit assignment problem. Indirect learning. You learn from delayed reward.
2. Valued function based: You learn from the utility, or expected value of reward. Main focus
3. Model-based: Direct learning. Expensive Computation.

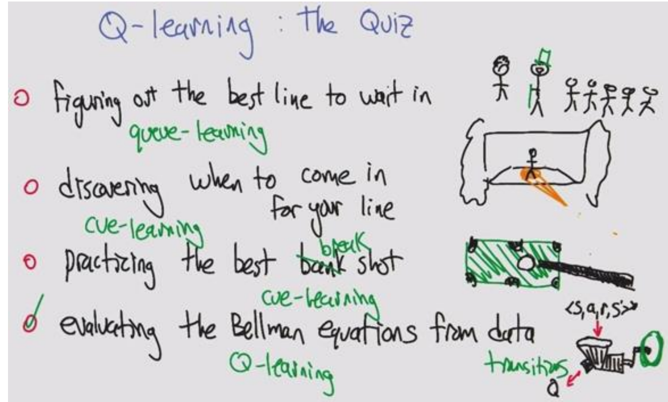
New kind of value function



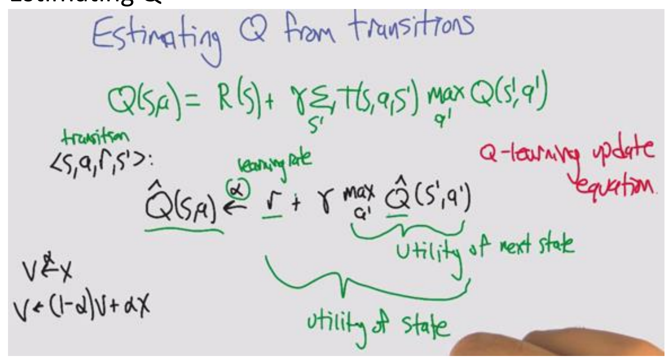


- Q: quality (other letters taken)
- Insert a utility step to compare the actions

Q-learning quiz

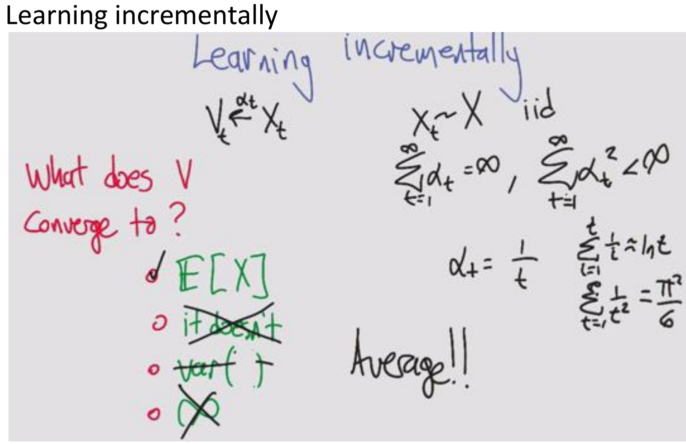


Estimating Q



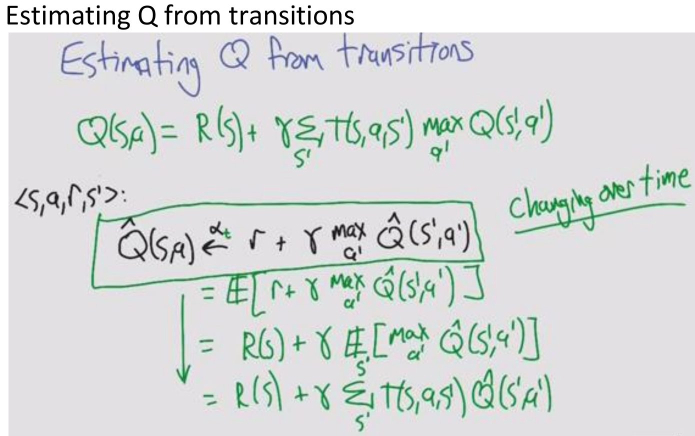
- Can't compute Q because we don't know R or T
- In MDPs, we do have R and T

Learning incrementally



- Basal problem (pi^2/6)

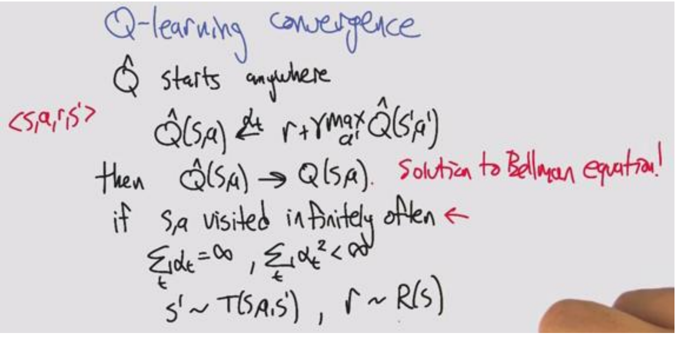
Estimating Q from transitions



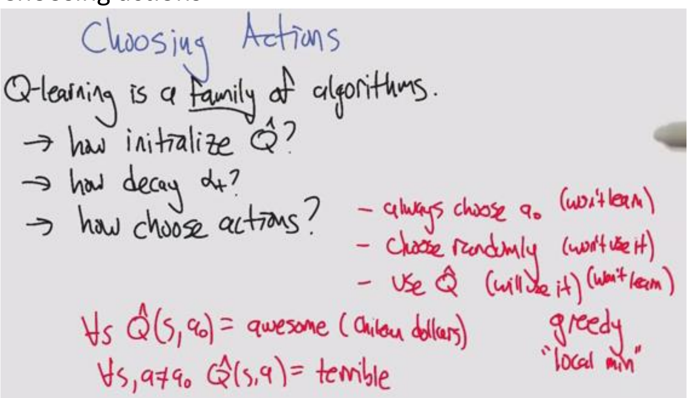
- Math tricks of expectations

- Simple update rule doesn't <u>actually</u> work because Q^ is a changing target

Q learning convergence

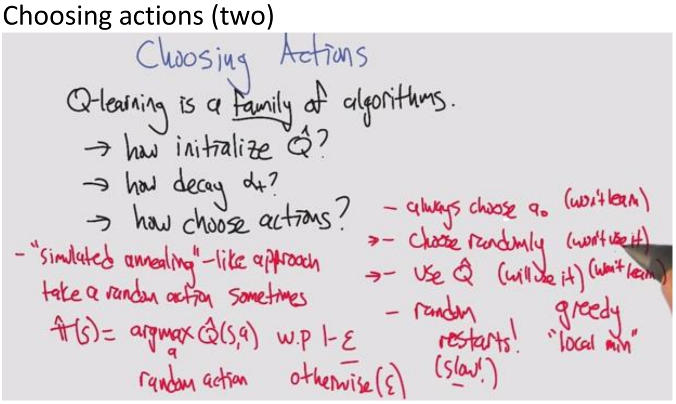


Choosing actions



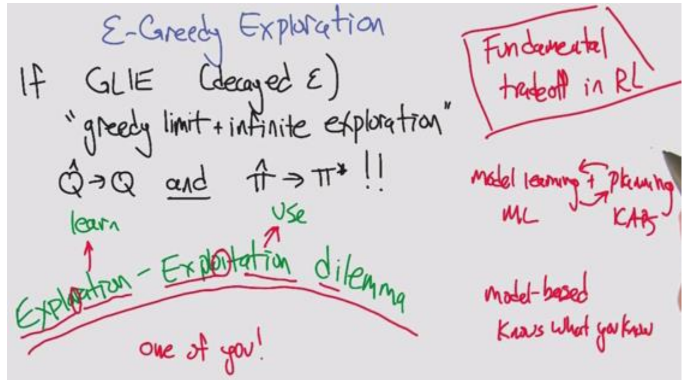
- Choosing bad actions
  - Always choose a0, ignoring q^
  - Choose randomly, wise but stupid. We know a lot, don't do anything about it
- Greedy local min
  - Use random restarts!

Choosing actions (two)



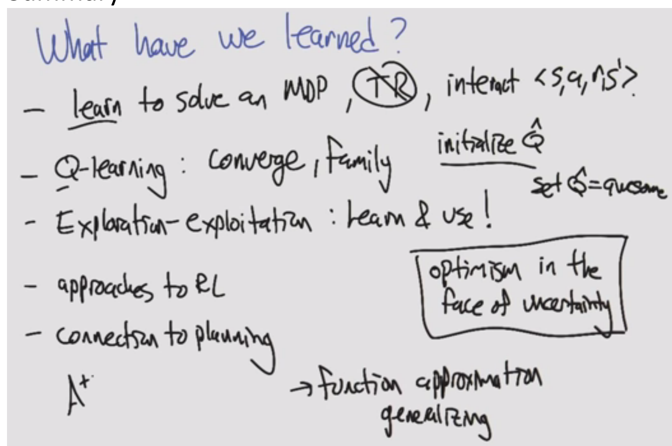
- Random optimization!
  - Simulated anneal: Random action sometimes

Greedy Exploration



- Trade-off: Exploration & Exploitation

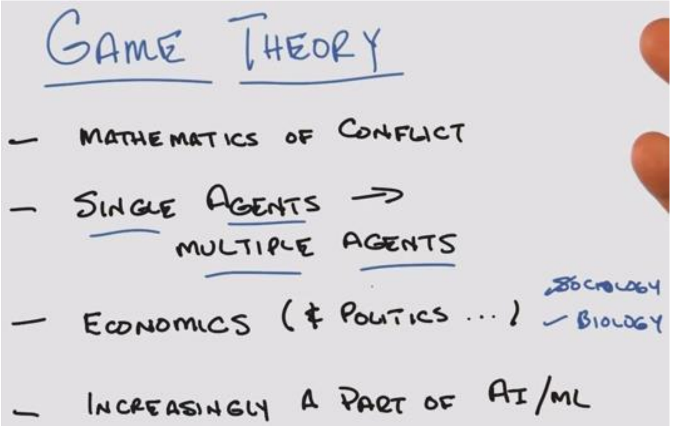
Summary



- Optimism in the face of uncertainty
- Issues of overfitting, importance of generalizing

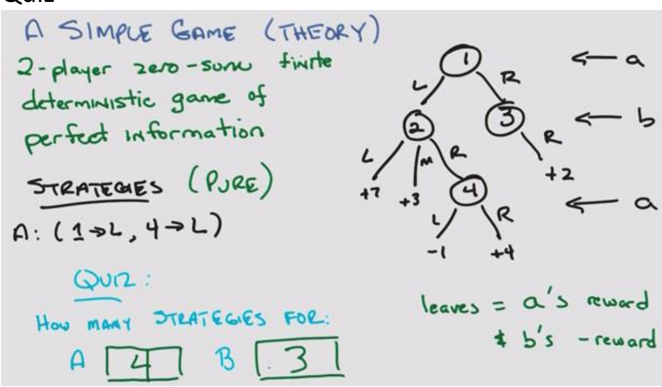
Lesson 3: Game Theory

Game Theory

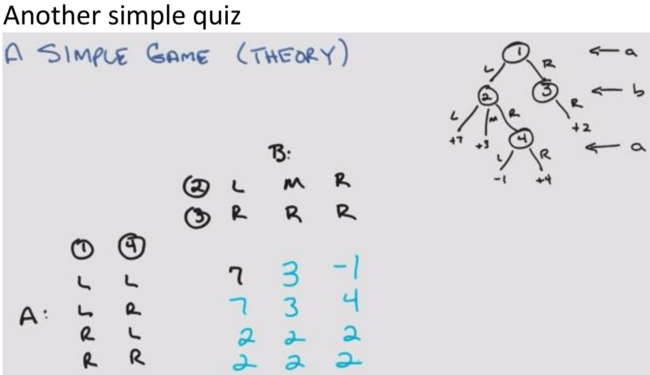


- Math of conflicts of interest
- Each agent has its own goals, intentions, interests

Quiz

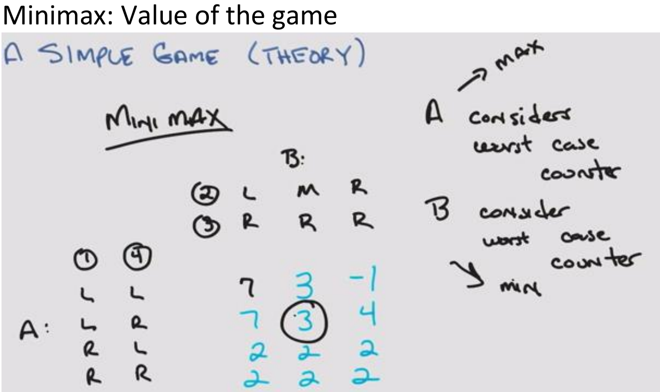


Another simple quiz



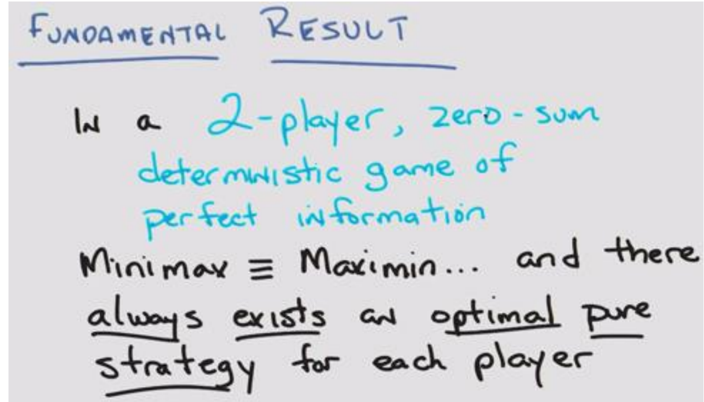
- Strategy and reward, the how is not important
- The only thing that matters is the matrix

Minimax: Value of the game



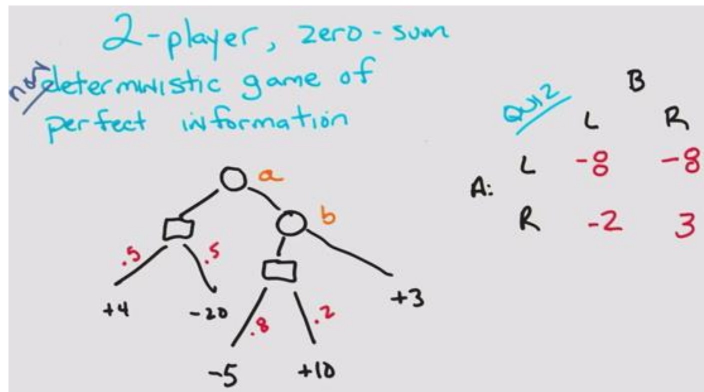
- A and B arrive at the same strategy

Fundamental result



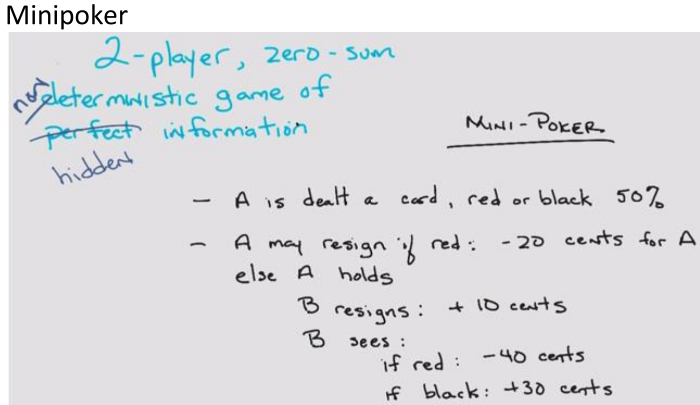
- Assume everyone is rational (trying for optimal, and assumes all other agents are maximizing)

Game tree



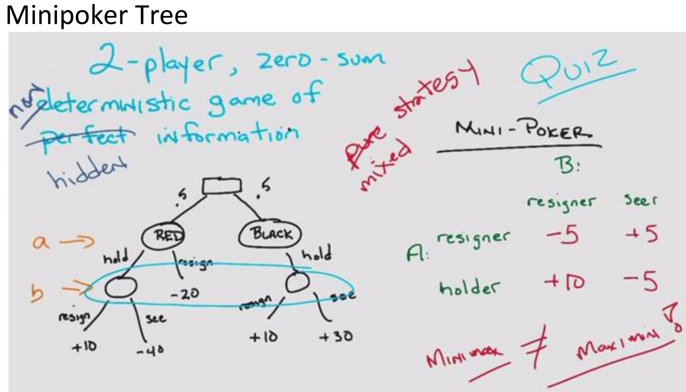
- Use the Expectation of stochastic probabilities
- Other theorem still holds! Von Neumann

Minipoker



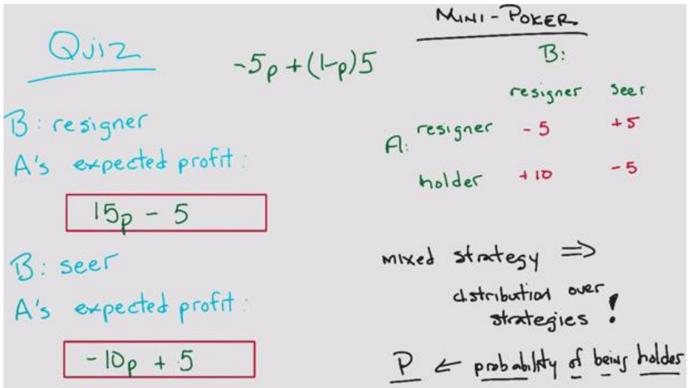
- Non-deterministic & hidden

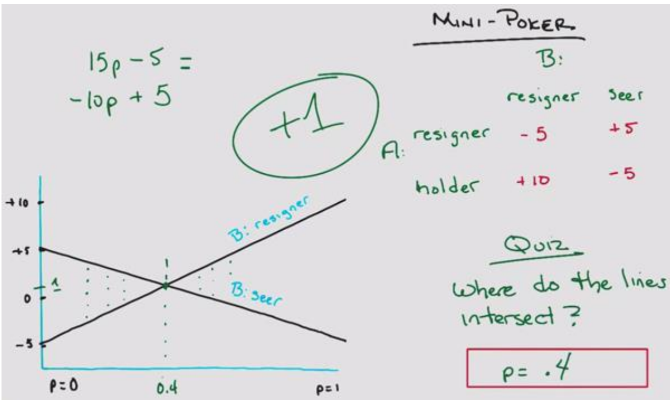
Minipoker Tree



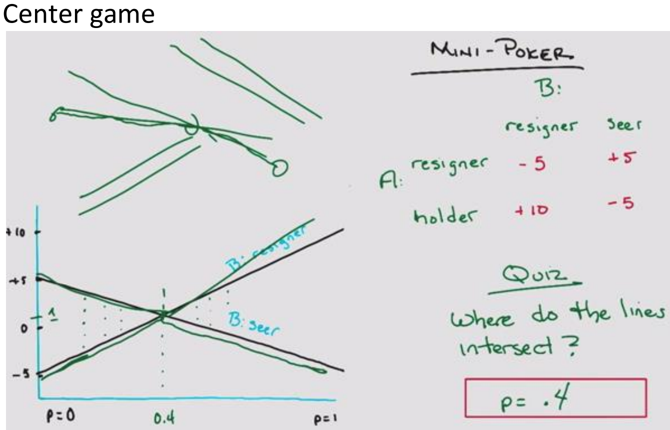
- Minimax != maximin

Quiz



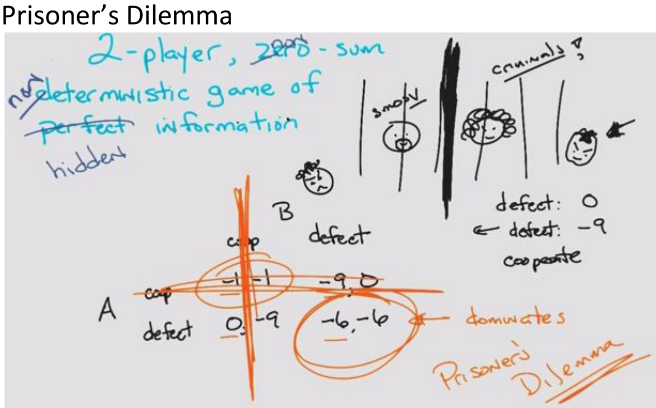


Center game



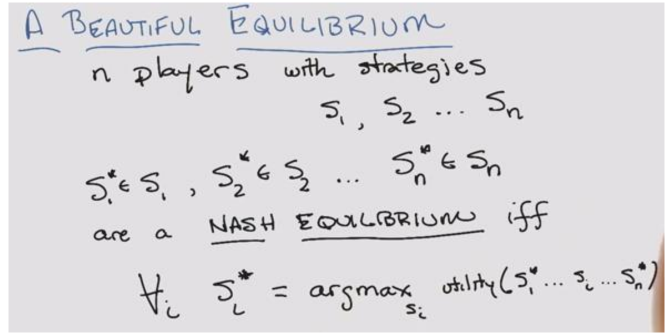
- Maximum of 3 discrete points: 2 extrema and intersection

Prisoner's Dilemma



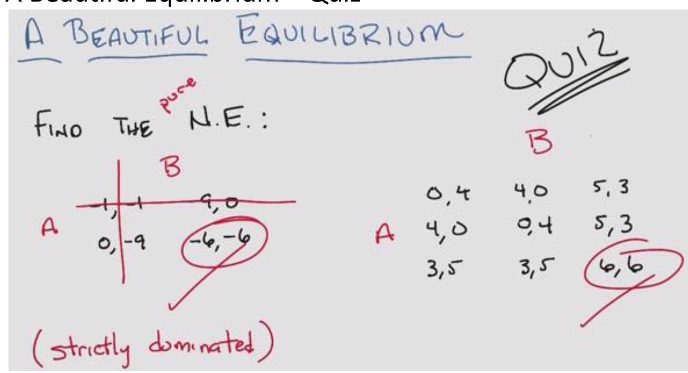
- Non-zero sum

A beautiful equilibrium

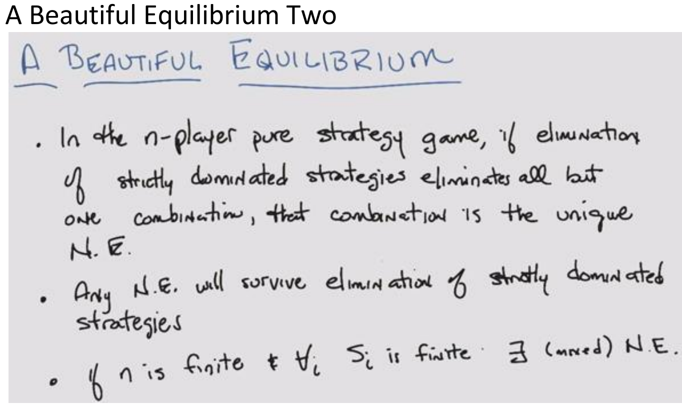


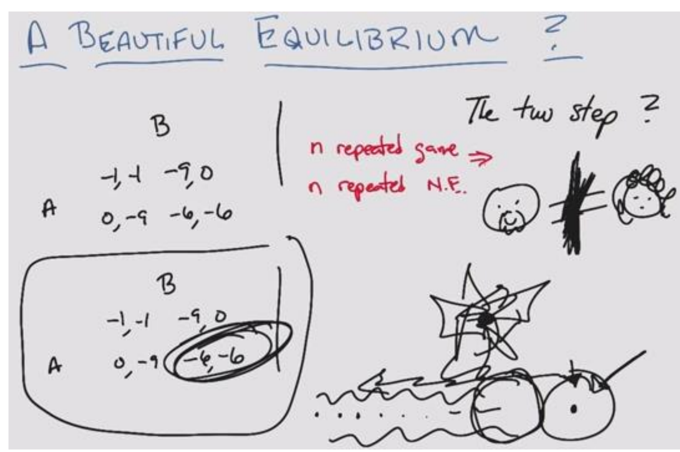
- John Nash: A Beautiful Mind

A Beautiful Equilibrium – Quiz

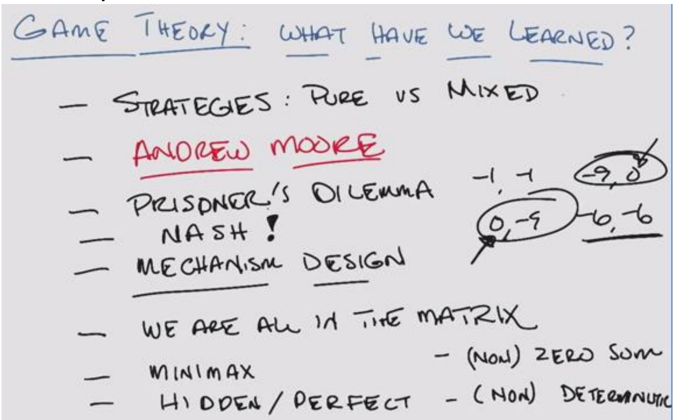


A Beautiful Equilibrium Two





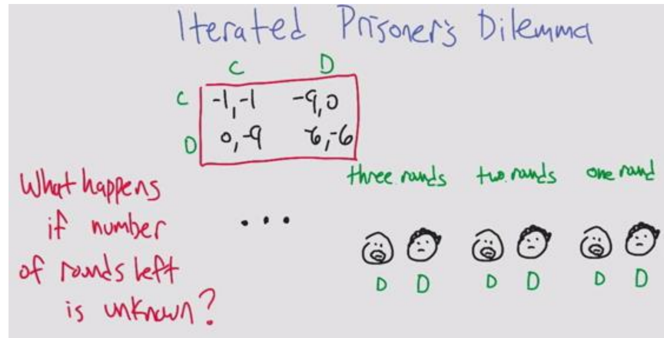
Summary



- Relaxed the constraints

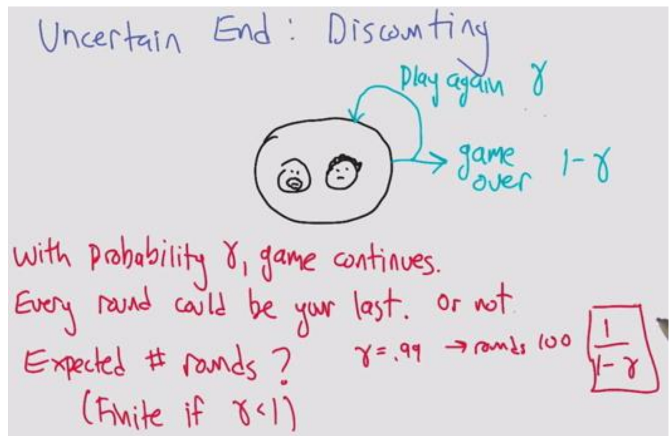
Lesson 4: Game Theory – Continued

Iterated Prisoner's Dilemma

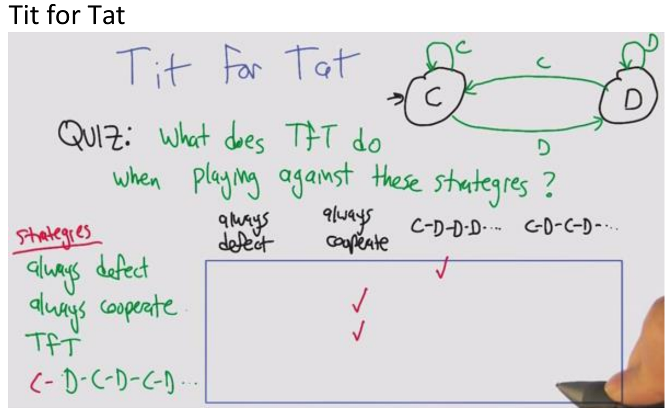


- C: cooperate
- D: defect

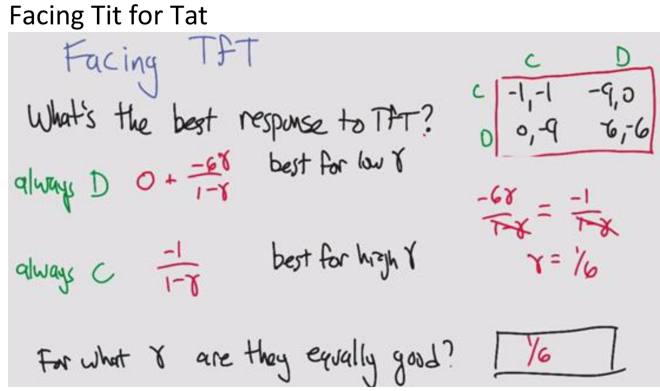
Uncertain End



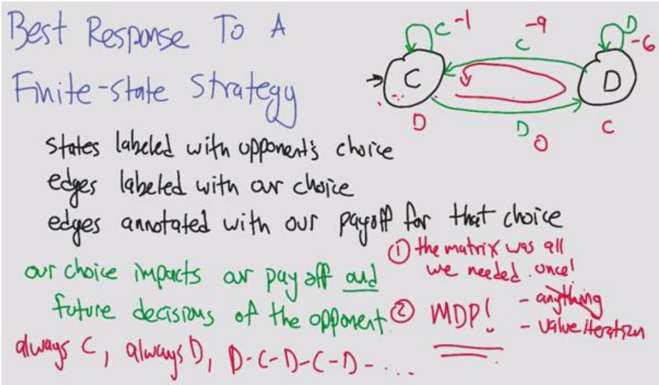
Tit for Tat



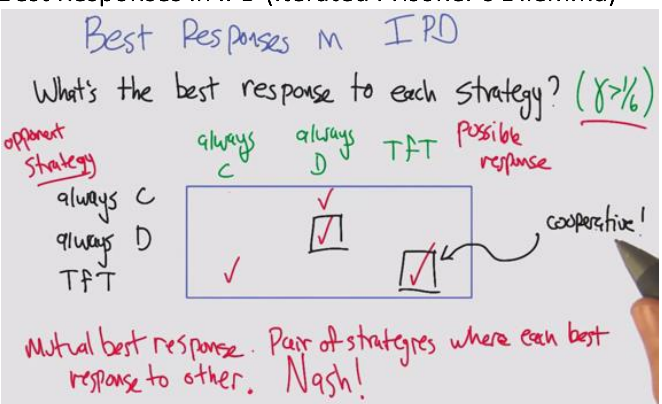
Facing Tit for Tat



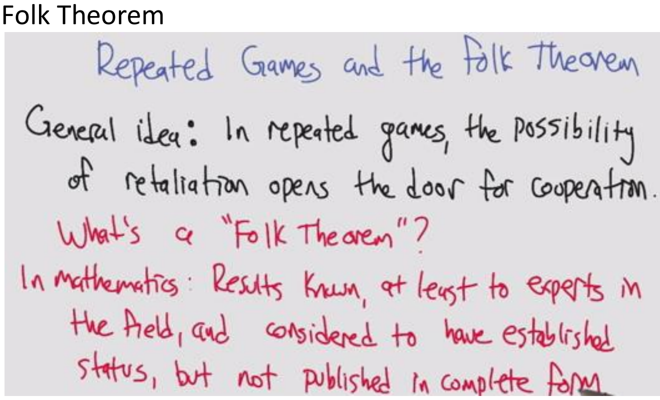
Finite State Strategy



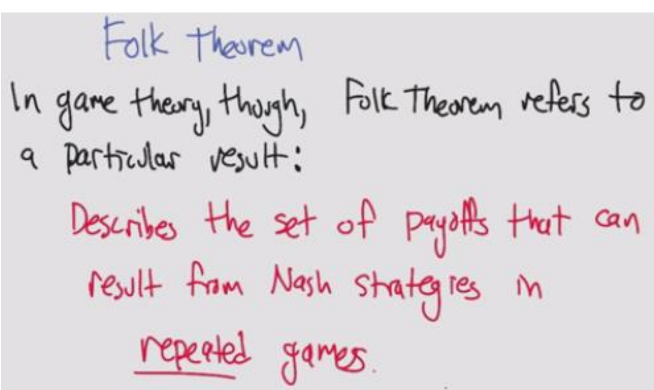
Best Responses in IPD (Iterated Prisoner's Dilemma)



Folk Theorem

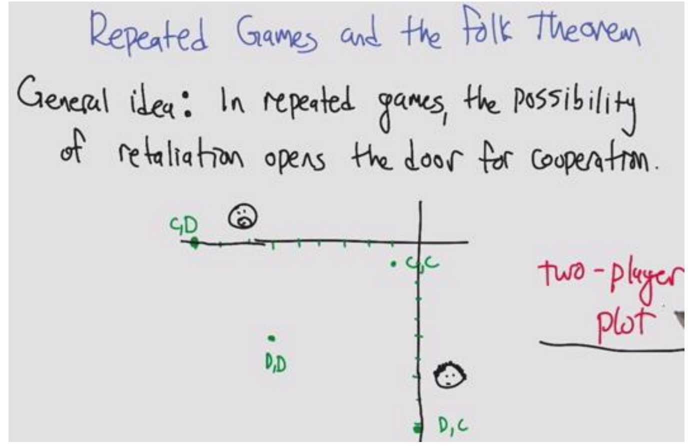


- Folk Theorem is the wrong wording in Game Theory
- In mathematics (ie. NOT in game theory): General understanding, not credited to a specific person

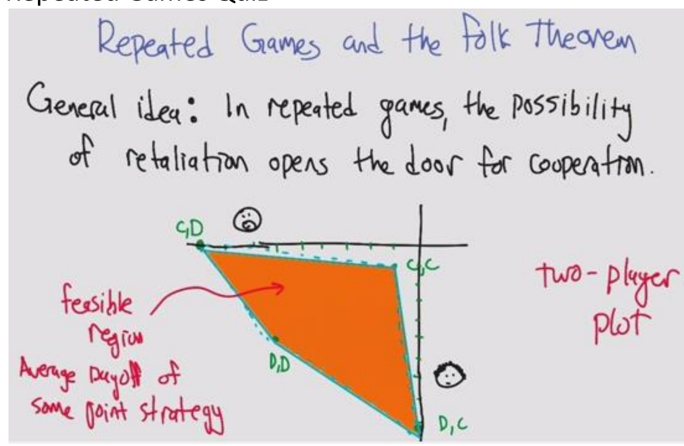


- In game theory: a set of payoffs that results from Nash strategies

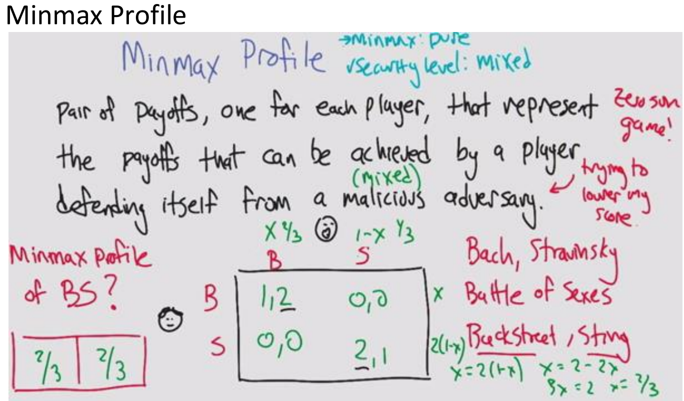
Repeated Games and the Folk Theorem



Repeated Games Quiz

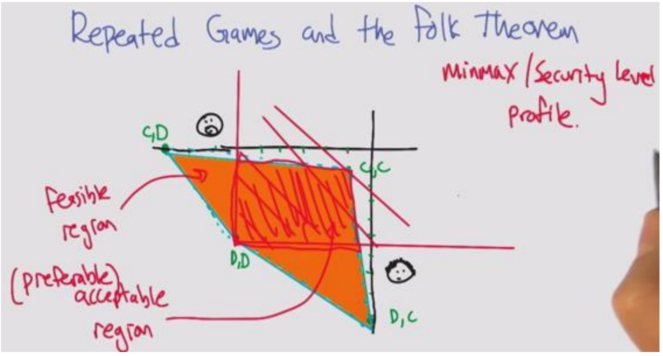


Minmax Profile

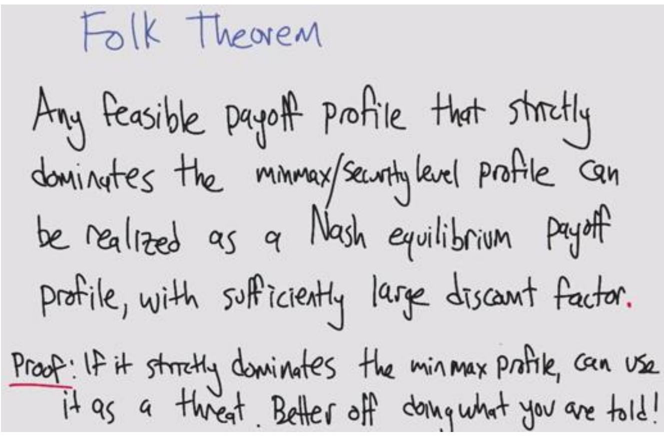


- Agent 1 and 2 deciding to go to Bach/Stravinsky
  - Example also works for Backstreet Boys and Sting
- Minmax Profile
  - Assume malicious AND random adversary: solve it like a zero-sumgame, using x and 1-x
  - MinMax Profile:
    - x = 2*(1-x)
    - 3x = 2
    - x = 2/3
- 1. Minmax strategy, if it is pure, the profile is 1,1
- 2. If mixed, it is 2/3, 2/3

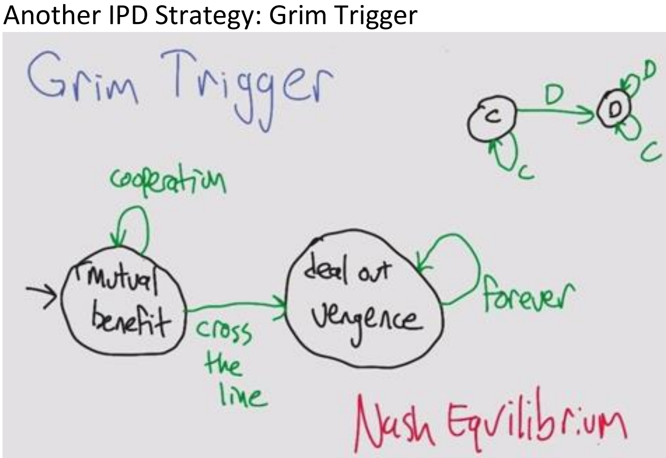
Security level Profile in the IPD (Iterated Prisoner's Dilemma)



Folk Theorem

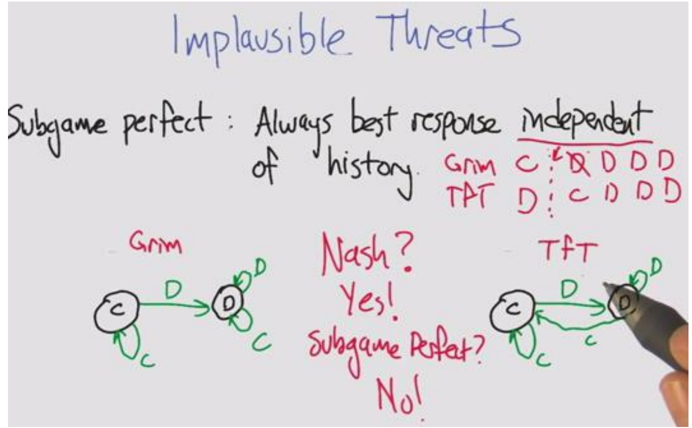


Another IPD Strategy: Grim Trigger



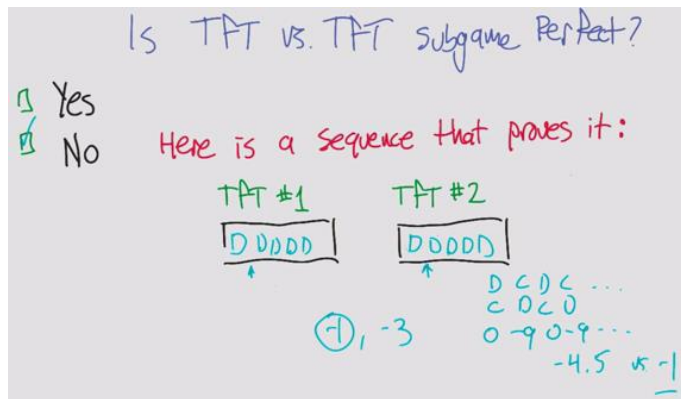
- Added to rational agents, each agent will become vengeful when the other defects
- Can't do anything better against Grim than to use Grim, therefore Nash!

Implausible Threats



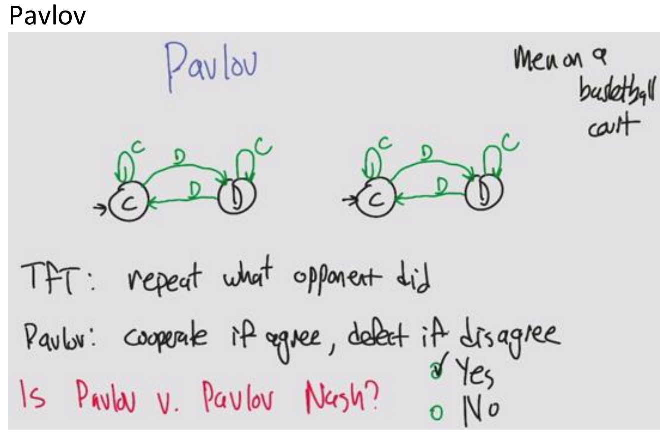
- Assuming that vengeful agents will forgo reward entirely
- Subgame perfect: Always best response independent of history

TfT vs. TfT



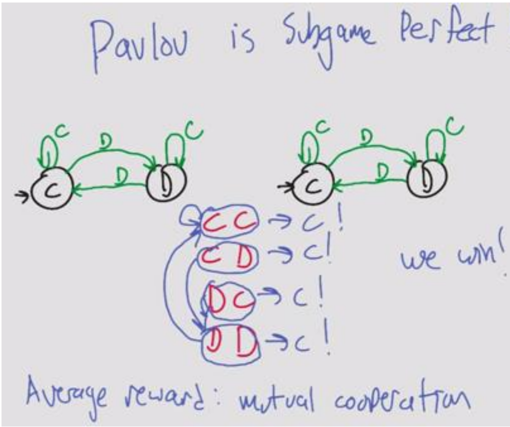
- If you were to change one agent's actions, could you improve the expected value?

Pavlov



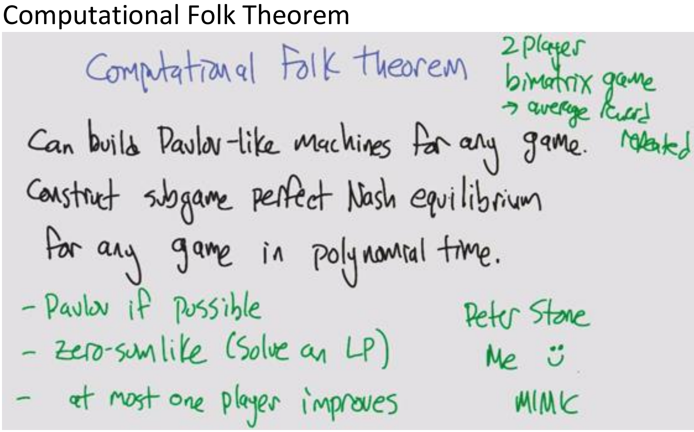
- Pavlov on how animals learn
- Defect against you until you realize it hurts, then cooperate again
- Sort of like men on a basketball court and fouls, aggressive until your opponent is not aggressive
- Pavlov is Nash

Pavlov



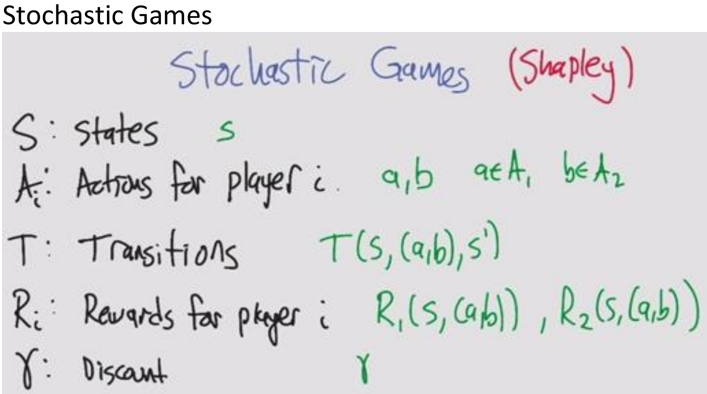
- Pavlov is Subgame Perfect
  - CC gives C
  - DD leads to CC
  - CD and DC lead to DD
  - The result is an average reward of CC
- Why is this interesting?
  - It becomes a plausible threat
  - Even when an agent decides to defect, it turns out the other agent will respond with defect and then the will agree on cooperating again

Computational Folk Theorem



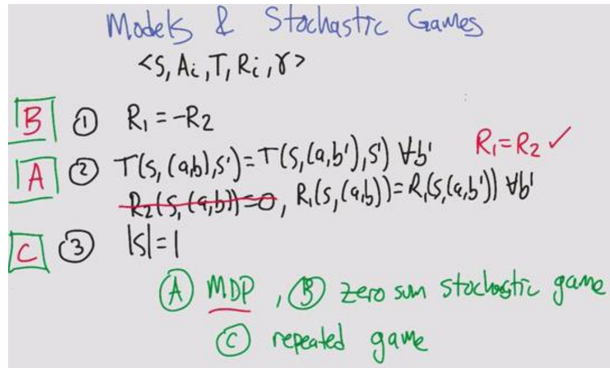
- Assume high discount factor
- Bimatrix if different reward structure per agent

Stochastic Games



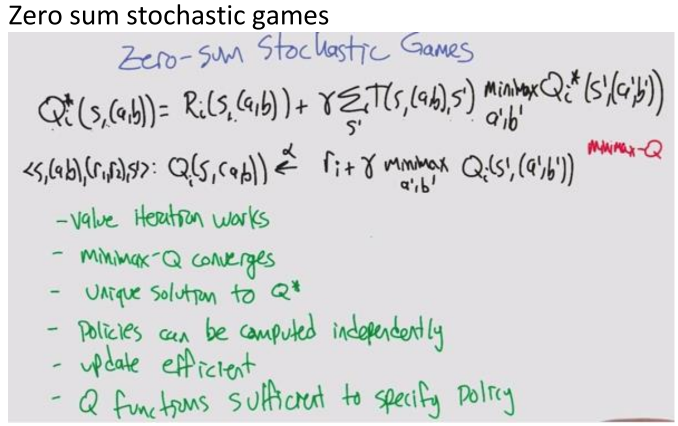
- Sometimes gamma is defined or inherent

Quiz: Match 1/2/3 and A/B/C



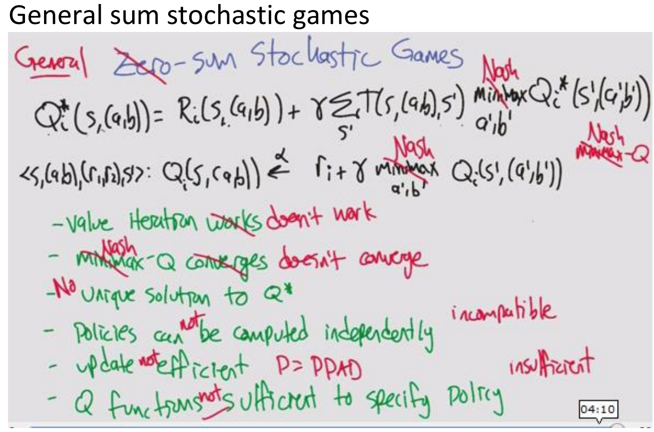
- B: Zero sum: easy
- A: Only one agent: MDP. Rewards only based on agent 1. Rewards of agent 2 don't matter to agent 1
  - R1 = R2 means the actions agent 1 affect both, but R2 isn't relevant to agent 1
- C: Only one state, same game. Actions affect rewards but not transitions

Zero sum stochastic games



- Zero-sum game assumes 2 players. 3 players is general sum with one player to make the sum 0
- Use minimax Q (max Q for a' b' does not make sense, since it is zero sum)
- Minimax can be solved using a linear program
- It's like a 1 agent game, but now with 2

General sum stochastic games

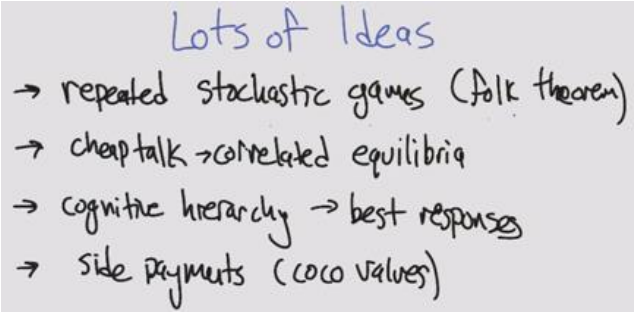


- Minimax doesn't work with 3+ players
  - Replace with Nash

- PPAD, as hard as NP for computation time

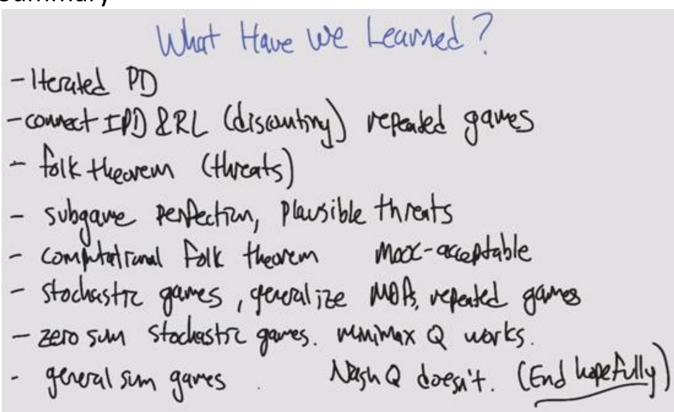
- Many many difficulties

Lots of ideas



- Coco: cooperative competitive values

Summary



ML Terminology

Information Theory

Entropy - does this feature have any information. If the sequence is predictable or has less uncertainty, then it has less information.
Mutual Information - are these input vectors similar?
Entropy of Y - Entropy of X given Y
What is the mutual information?
MI(X,Y)= sumX,Y p(x,y) log p(x,y)/p(x)p(y)
This is the KL divergence between p(x,y) and p(x)p(y).
Independent random variables have zero MI.
What is the relation between mutual information and information gain?
Two names for the same thing: MI(X, Y)=H(Y)-H(Y|X).

MDP

Components of a MDP:
States S - every possible game location
Models T - T(s,a,s'). physics of the world. probability you will transition to state s' when in s and take action
Actions A - thing you can do in a particular state. ie up,down,left.right
Rewards R - reward function. scalar value for being in a state
Policy P - Solution to a MDP. Function that takes in current state and returns action
(Only the present matters)

Reinforcement Learning

Bellman equation - dynamic optimization problem in discrete time can be stated in a recursive, step-by-step form by writing down the relationship between the value function in one period and the value function in the next period. The relationship between these two value functions is called the Bellman equation.
Q-learning - Evaluating the Bellman equations from data. You take in states, actions, reward and next states and try to learn a Q function. Uses transitions (data) to directly product the solution to the Q equations.
Estimating Q from transitions. Don't have R or T so have to come up with some other way to solve these kinds of equations.
In learning scenario we don't have the model but we have the transitions

Game Theory

Strategy profile - A list consisting of one strategy for each player.
Nash equilibrium - A strategy profile for which each player's strategy is a best response to the profile of all other players' strategies.

Additional

- Monotonic: One direction
- Non-Increasing: Equal or smaller
- Clustering: Richness, Scale-Invariance, Consistency
- Mutually Contradictory
- Strongly/weakly relevant & irrelevant have to do with ***information***
    - If removed, how does it impact Bayesian Optimal Classifier?
- Usefulness has to do with ***error***
- Polysemy: word has multiple meanings. False positives
- Synonymy: same meaning expressed in different words. False negatives
- Mutual information: are the inputs similar?
- Entropy: does the feature have any information?
- Markovian: Only present matters
- State
- Action
- Transition
- Reward: Immediate
- Utility: Sum of discounted rewards of a policy (longterm)
- Maximin
- Minimax
- Components of a game: # players, finite, game sum, deterministic/probabilistic, perfect/imperfect/hidden info
- Normal agent: assumes it is trying to maximize, behaves rationally
- Strategy: pure/impure
- Nash's Equilibrium: A game where each agent will not change its strategy as long as all other agents do not either
    - No profitable deviation
- Folk Theorem in Game Theory
- Malicious adversary: All you care about is hurting the opponent
- Minmax profile: Expected pairs of payoffs
- Security profile: Feasible region > Acceptable Region
- Grim Trigger: Strategy of cooperating but becoming vengeful when wronged
- Subgame Perfect: Optimal strategy is independent of history
- Pavlov: Strategy of cooperating when the same, defecting when different

Practice Questions

Supervised Learning

1. Explain cross-validation, and why you may want training/validation/test sets
2. Give examples of restrictive, preference and inductive bias
    a. Restrictive: Perform tests within computation limits, chosen parameters
    b. Preference: Prefer more information gain, correctness, simplicity
    c. Inductive:
3. What is the difference between an eager and lazy learner?
    a. This is a comparison of how much time the learner spends building the model versus evaluating the data
    b. Eager: Lots of learning, Slow model – Lazy: Less learning, fast model
4. Explain decision trees
    a. With the feature with the most information gain as the root node, each successive node is the feature with the next largest info gain
    b. Confidence and minimum nodes
    c. Eager learner
5. Explain pruning
    a. A method of removing branches to reduce the effect of overfitting, but may compromise accuracy
    b. This also has a positive effect on computation, although DT is very fast
6. Explain k-nearest neighbors
    a. Lazy learner
    b. Identifies k points with the most similar features using a distance and predicts the classification
    c. Distance function (Euclidean, Manhattan, KL Divergence)
7. Explain artificial neural networks
    a. Using layers of perceptrons (commonly 3 layers: input, hidden, output), the threshold theta for each perceptron is evaluated using a computation-heavy process of back-propagation
    b. Learning rate (L), momentum (M) and iterations (I)
    c. Eager learner
8. Explain support vector machines
    a. Eager learner
    b. Transforms the data using the Kernel Trick to optimize the distance between points
9. Explain bagging
    a. AKA bootstrap aggregation, Making additional datasets as combinations of your existing data
10. Explain boosting
    a. Make several average-performance models and combine them using a majority vote
    b. Ex. Using a series of parabolas to describe a quartic function

Clustering

11. Explain single link clustering
    a. Consider all objects as clusters
    b. Compute the intercluster distance as the distance between the two closest points of each cluster
    c. Combine the two closest clusters
    d. Repeat for n-k times
12. Explain kmeans (P, x, C, y)
    a. For k clusters, randomly pick centers
    b. Label all points closest to each center as a cluster
    c. Use the average of the intracluster distances to re-compute the cluster centers
    d. Repeat until convergence
13. Explain expectation maximization
    a. For k clusters, randomly pick centers
    b. Expectation: Compute the likelihood that each point is in each cluster
        i. E(z) = fn (P(x), miu)
    c. Maximization: Re-compute the cluster centers based on the weighted average of likelihood and points
        i. Miu = fn (E(z), x)
    d. Repeat (does not converge, but in practice it does)
14. Explain richness, scale, consistency
    a. Richness: There is some distance matrix D that PD returns the clustering
    b. Scale: The clusters remain the same whether the point distances are uniformly increased or decreased
    c. Consistency: The clusters remain the same when intracluster distances decrease and/or when intercluster distances increase
15. In single link clustering, label R, S, C for:
    a. n/2 clusters reached (SC)
    b. clusters are theta units apart (RC)
    c. clusters are theta/w units apart where w =max D(I,j). Normalization (RS)

Feature Selection

16. Explain the difference between filtering and wrapping in feature selection.
    a. Filtering selects features and feeds it into a learner
    b. Wrapping applies the learner on different subsets before selecting the final features to remove

Mock Final

True/False questions

1. K-means is a clustering algorithm that is guaranteed to converge.

Solution: True. There exists a convergence proof.

2. The main difference between immediate and delayed reinforcement learning is in how often the rewards are received.

Solution: False. A delayed reinforcement learning task is one where the optimal solution can only be found by associating incoming rewards with a whole sequence of previous actions, instead of just the latest one. The reward may very well be received in every time step also in delayed reinforcement learning.

3. A tit-for-tat strategy makes it possible for players to cooperate without colluding.

Solution: True

4. When two children fight over a piece of cake, it is an example of a zero-sum game.

Solution: True

5. A Nash equilibrium is always a dominant strategy equilibrium.

Solution: False. A dominant strategy is one which is superior no matter what the opponent does. A Nash Equilibrium is a strategy that, given perfect knowledge of the opponent's actions, would not change.

6. One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

Solution: False. Q-learning does not know about the environment beforehand.

7. Application of Bellman's equations require a complete and accurate model of the environment

Solution: True.

8. Kmeans and EM clustering methods both require providing the value of K in order to form clusters.

Solution: False. Kmeans requires K as an input but EM is able to determine a K value by itself. However, determining the value of K requires more time.

Unsupervised Learning

1. Thinking about unsupervised learning and the k-Means and expectation maximization (EM)
algorithms, which one of the following statements is true:
(a) K-Means algorithm fits clusters only in hyperspheres and EM algorithm fits data only in hyperelipsoids
(b) K-Means using euclidean distance is a particular case of the EM-algorithm when we are fitting K-gaussian distributions with the same variance for each attribute.
(c) EM algorithm has the same computational cost no matter the number of parameters that have to be estimated for the probability distribution that we are fitting to the attributes.
(d) K-Means assigns a probability to the membership of each example to each cluster

Solution:
a- false. K-means create spheres but EM creates clusters based on the gaussian distribution with the data.
b- true
c- false
d- false, this describes EM (originally: KMeans is a 'hard' clustering algorithm)

2. Thinking about unsupervised learning, which ones of the following statements are true (multiple choice):
(a) Differently from partitional graph based algorithms and density estimation algorithms the K-means and the EM algorithm need as a parameter the number of clusters to find
(b) With hierarchical clustering algorithms based on graph theory we obtain a partition of a dataset in K different classes
(c) The K-Means algorithm obtains a global optimal solution for the partition of a dataset by minimizing the square distance between examples and their nearest centroid
(d) The EM algorithm assumes that the model of the data comes from a mixture of K-dimensional probability distributions

a- True (I thought this was false)
b- True: For SLC, if we model the cluster distances as edge weights, then the minimum spanning tree algorithm creates the clusters.
c- False: K-means minimizes ***variance***, but it doesn't have to be square distance as the metric. Also it can fall in local maxima.
d- False: The EM algorithm assumes the model of the data comes from K number of Gaussian mixtures. # of dimensions is inherent in the model.

3. K-means only finds clusters that are a local (but not a global) maximum of the objective function J it minimizes. What does this mean? What are the implications of this fact for using k-means to cluster real world datasets?

Solution: K-means will not necessary find the best clustering with the tightest clusters and the quality of its results strongly depend on initialization. Run K-means multiple times with different ***initializations/random seeds*** and choose the clustering which has the lowest value for J.

4. K-means has difficulties clustering datasets which contain a lot of outliers. Explain, why this is the case! What could be done to alleviate the problem?

Solution:
As K-means requires that all objects need to be assigned to a cluster; therefore, outliers have to be assigned to a particular cluster, leading to non-descriptive centroids that no longer capture the characteristics of most objects, belonging to a particular cluster.
Some techniques that have some merit include:
-***Remove*** outliers, prior to applying k-means

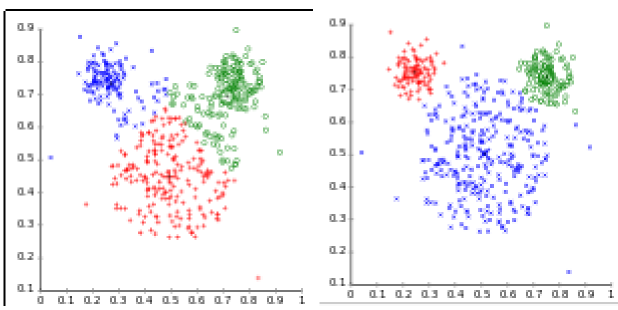
-Use the cluster ***medoid***, instead of the cluster centroid as the cluster summary

5. Contrast between which clustering problems would be relatively/poorly suited to K-means Vs EM.

Solution: More needed!

- ***Cluster Size:*** EM is able to accommodate clusters of variable size much better than k-means.
- ***Time:*** K-Means is more efficient so is useful in situations where the application has computational constraints.
- ***Soft Clustering:*** EM can deal with data between at the same distance between two clusters while K-means will include in one cluster or another.
- When the clusters seem well formed, K-means is faster.

6. Two plots are shown below that have had cluster analysis applied to them. Decide which has been created with K-Means and which has been created by EM. Explain why you have made this choice.



Solution:

The first image was produced with K-Means clustering. You can see that K-means tends to produce ***equi-sized*** clusters. The second image was produced by EM. EM benefits from the Gaussian distribution present in the data set.

7. What is the difference between relevance vs usefulness in the context of feature selection?

Relevance measures the effect on BOC

Usefulness measures the effect on a predictor like (least squared error)

Dimensionality Reduction and attribute selection

1 Compare and contrast the following mechanisms for dimensionality reduction. ICA, PCA, Randomized Projections. How do they work? Strengths, Weaknesses. How does it work?

Mechanisms

PCA finds the basis vectors that best explain the ***variance*** of the data. The magnitude of these vectors is called an eigenvalue and we are able to drop the basis vectors with low value eigenvalues.

ICA works by finding the basis vectors that give a result such that this resulting vector is one of the ***independent*** components of the original data. The relevance of these vectors is judged according to their deviation from the Standard Distribution, often measured by ***Kurtosis***.

Random Projections reduces the dimensionality of the data by ***projecting*** it onto a lower dimensional subspace using a random matrix with columns of unit length.

Strengths

PCA Performs well when finding ***global patterns*** in the data.

ICA performs well to ***decompose mixed signals*** when the original set of signals are mutually independent and the values of each source signal have a non-gaussian distribution.

Random projections tends to be ***faster*** than the other considered algorithms.

Weaknesses

ICA and PCA are based on linear algebra so must convert ***nominal values to binary*** which can cause an increase in dimensionality to a number of dimensions greater than the original ***dimensionality*** even after reduction.

Reconstruction of data transformed by RP is poor.

Here is some more data related to these pulled from Piazza:

PCA	ICA	RCA	LDA
Global	Finds transformed features which themselves are mutually independent	Randomly projects data to M dimension, where M << N (original dimension).	Linear Discriminant use observables label to project data in new space.
	However attempts to find maximum mutual information between transformed feature and observed (original) features.		
Mutually orthogonal	Transformed features are not necessarily orthogonal.	Typically M from RCA tend to be larger than that of M from PCA.?	SVM is an example of LDA.
Transforms features based on largest variance, mutual information	If we are in a world where observables are linear sum of independent causes (variables) then we should not use "maximal variance" to	It appears by randomly reducing features to M dimension we tend to retain enough information	

	transform feature (i.e; do not use PCA).	of original observables.	
	As the process of finding “maximal variance” going to sum together otherwise independent causes (variables) to obtain maximum variance, which will not be useful in our search to find them from observables.		
Finds best correlation	ICA assumes hidden features are “highly non normally distributed”.	Reducing to M lower dimension helps us with curse of dimensionality problem.	
A good PCA transformation will have low reconstruction error.	ICA finds topics as transformed features from set of observable documents.	RCA tend to be FAST.	
Finds a bag of features.	ICA helps us in knowing independent features of domain of our observables (data). Once we know this, then we can write efficient algorithm that targets to find these hidden features directly.	RCA is particularly useful if our goal is to do classification using RCA transformed data (helps beating curse of dimensionality problem),	
Features are ordered,			

1st PCA, 2nd PCA			

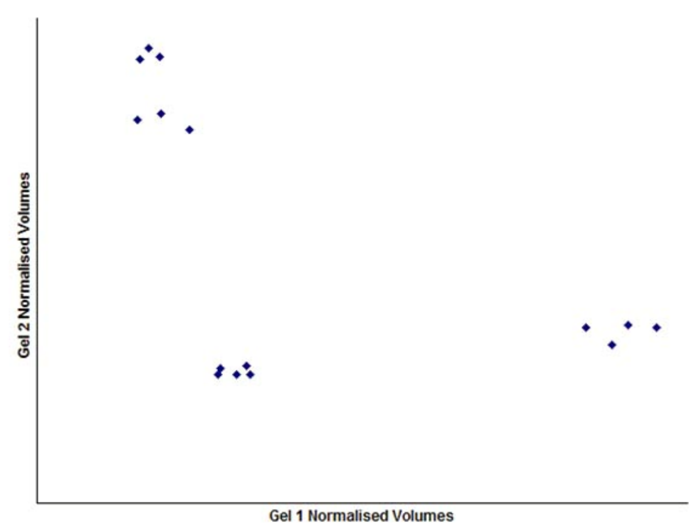
2 Thinking about dimensionality reduction and attribute selection, which ones of the following statements are true (multiple choice):
(a) PCA and ICA transform a dataset to a space with the same dimensionality optimizing a measure that preserves the distances among all the pairs of examples
(b) PCA is an unsupervised method for dimensionality reduction

Solution (my own idea, not taken from some exam)
a - False, PCA and ICA use different mechanism for dimensionality reduction so do not transform a dataset to the same dimensionality
b- True, PCA does not require the use of labeled data

3 The key assumption of a naive Bayes (NB) classifier is that features are ***independent***, which is not always desirable. Suppose that linear principal components analysis (PCA) is first used to transform the features, and NB is then used to classify data in this low-dimensional space. Is the following statement true? Justify your answers.
The independent assumption of NB would now be valid with PCA transformed features because all principal components are orthogonal and hence uncorrelated.

Solution:
This statement is false. First, uncorrelated is not equivalent to independence. Second, transformed features are not necessarily uncorrelated if the original features are correlated in a nonlinear way. It is suggested to use ICA for this application.

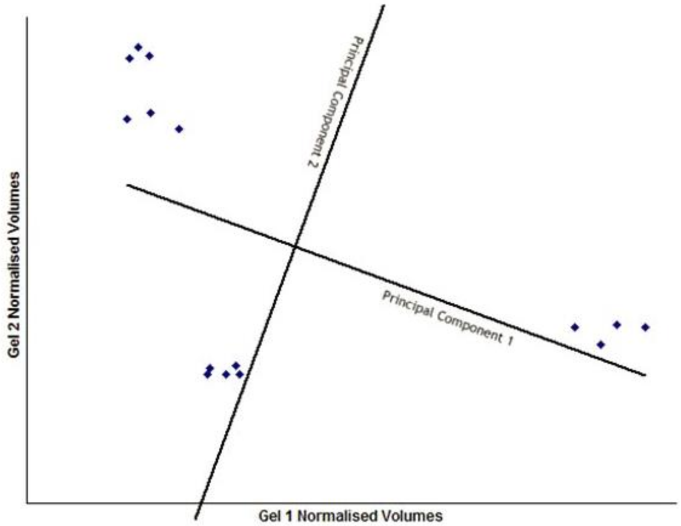
4 Consider a simple biology experiment with 2 gels and 15 spots on each gel. We can plot the normalised volumes of the spots in a 2-dimensional graph.



Find the first and second PCA components

Solution:

The first PCA component is the direction of the highest variance. The 2'nd PCA component is perpendicular to this one.



Information Theory

Assume we have a classification problem involving 3 classes: professors, students, and staff members. There are 800 students, 100 staff members and 100 professors. All professors have blond hair, 50 staff members have blond hair, and 400 students haveblond hair. Compute the information gain of the test "hair_color='blond'" that returns true or false. Just giving the formula that computes the information gain is fine; you do not need to compute the exact value of the formula! Use H as the entropy function in your formula (e.g. H(1/3,1/6,1/2) is the entropy that 1/3 of the examples belong to class1
1/6 of the examples belong to class 2, and half of the examples belong to class 3).

Solution

H(0.8,0.1,0.1)-0.55*H(400/550,50/550,100/550)-0.45*H(400/450,50/450,0)

Assume we have 2 independent fair coins:
P(A) = P(B) = .5
P(A,B) =
P(A|B) =
H(A) =
H(B) =
H(A,B) =
H(A|B) =
I(A,B) =

Solution:
P(A) = P(b) = .5
P(A,B) = Joint probability is given by product of P(A)*P(B) = .25
P(A|B) = P(A) since they are independent of each other = .5
H(A) = - Sum P(A) Log P(A) = ..5 log .5 - .5 log .5 = 1
H(B) = - Sum P(A) Log P(A) = ..5 log .5 - .5 log .5 = 1
H(A,B) = Sum P(A,B) log P(A,B) = -4 (.25 log .25) = 2
H(A|B) = -Sum P (A,B) log P(A|B) = 1
I(A,B) = Mutual Information = H(A) - H(A|B) = 1 -1 = 0. Since the 2 coins are independent they do not have any mutual information on each other.

Assume we have 2 dependent fair coins:
P(A) = P(B) = .5
P(A,B) =
P(A|B) =
H(A) =
H(B) =
H(A,B) =
H(A|B) =
I(A,B) =

Solution:
P(A) = P(B) = .5
P(A,B) = .5 since both can be either heads or tails.
P(A|B) = P (A,B) / P(B) = .5/.5 = 1
H(A) = 1 since still using fair coins
H (B) = 1 since still using fair coins
H(A,B) = 1
H(A|B) = 0
I(A,B) = 1

Dynamic Programming

Markov Decision Process

1. Explain the basic steps in policy-iteration as applied to solving a Markov Decision Process.
Solution:

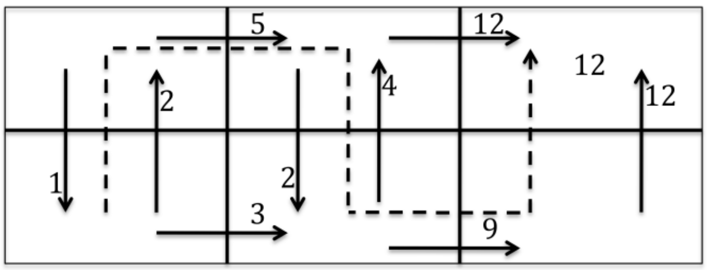
- Start off with an arbitrary policy
- Use the current policy to estimate the value function (utility of each state)
- Can be done in several ways. Solving a system of linear equations, nested value iteration, linear programming.
- Use estimate of the value function to produce a new policy
- Go back to step 2 until convergence or you get tired

Reinforcement Learning

1. Thinking about Reinforcement Learning which ones of the following statements are true (multiple choice):
(a) The maximization of the future cumulative reward allows Reinforcement Learning to perform global decisions with local information
(b) Q-learning is a temporal difference RL method that does not need a model of the task to learn the action value function
(c) Reinforcement Learning only can be applied to problems with a finite number of states
(d) In Markov Decision Problems (MDP) the future actions from a state depend on the previous states

Solution:
a) True?
b) true. Q learning does not need to know about a model of the task

c) false, does not need to know about states
d) false, each state is independent
2. What are the main differences between supervised learning and reinforcement learning?
Solution:
SL: static world[0.5], availability to learn from a ***teacher***/correct answer[1]
RL: dynamic changing world[0.5]; needs to learn from ***indirect, sometimes delayed feedback/rewards***[1]; suitable for exploring of unknown worlds[1]; temporal analysis/worried about the future/interested in an agent's long term wellbeing[0.5], needs to carry out actions to find out if they are good—which actions/states are good is (usually) not known in advance1[0.5]
3. Consider the deterministic world below. Arrows show allowable moves. There is a reward of 12 units for entering the top-right most state. There is no reward for being in any other state. Given the Q(s,a) values in the figure below,



a) show the changes in the Q(s,a) estimates for the first two steps in the path
shown by the dotted line (the agent starts in the lower left cell) when discount factor =0.5.
b) Explain why the new Q value computed in the third step is not affected by the new Q value computed in the second step.
Solution:
First: 0+0.5*5-2 = 0.5
Second: 0+0.5*12-5 =1
Third = r+0.5*max(4, 9) - 2 = 2.5
Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]
According to this formula, a value assigned to a specific element of matrix Q, is equal to the sum of the corresponding value in matrix R and the learning parameter Gamma, multiplied by the maximum value of Q for all possible actions in the next state. So, the new Q value completed in the third step is not affected by the value from the second step.

Game Theory

Find the NE (Nash Equilibrium) for the following 3 games

1. Game 1
Player 1 moves are rows and Player 2 moves are columns

3,2	1,1
1,1	2,3

Solution:
The NE for this game are (A,A) and (B,B).
(A,A) - Neither player can increase the payoff by selecting a different action. This is a NE.
(A,B) - Player 1 can choose B to obtain a higher payoff. Player 2 can choose A to obtain a higher
payoff. This is not a NE.
(B,A) - Player 1 can choose A to obtain a higher payoff. Player 2 can choose B to obtain a higher
payoff. This is not a NE.
(B,B) - Neither player can increase the payoff by selecting a different action. This is a NE.

2. Game 2
Player 1 moves are rows and Player 2 moves are columns

5,4	4,5
4,5	5,4

Solution:
The game has no NE.
(A,A) - Player 2 can choose B to obtain a higher award. This is not a NE.
(A,B) - Player 1 can choose B to obtain a higher payoff. This is not a NE.
(B,A) - Player 1 can choose A to obtain a higher payoff. This is not a NE.
(B,B) - Player 2 can choose A to obtain a higher payoff. This is not a NE.

3. Game 3
Player 1 moves are rows and Player 2 moves are columns

8,8	2,2
2,2	9,9

Solution:
The NE for this game are (A,A) and (B,B)
(A,A) - Neither player can increase the payoff by selecting a different action. This is a NE.
(A,B) - Player 1 can choose B to obtain a higher payoff. Player 2 can choose A to obtain a higher payoff. This is not a NE.
(B,A) - Player 1 can choose A to obtain a higher payoff. Player 2 can choose B to obtain a higher payoff. This is not a NE.
(B,B) - Neither player can increase the payoff by selecting a different action. This is a NE.

Piazza

- Romeo
  - **Let's update this post adding resources to help us prep for the final exam.**
  - There are no transcripts/slides for this section of the course, but I found these screencaps and notes by Qing Yang: Thanks, wherever you are!
  - Clustering: http://www.jianshu.com/p/dd26d36cc465
  - Feature Selection: http://www.jianshu.com/p/87fbe8378873
  - Feature Transformation: http://www.jianshu.com/p/530c4aeac948
  - MDPs: http://www.jianshu.com/p/881ab7e41adb
  - Reinforcement Learning: http://www.jianshu.com/p/5134962f78ee
  - Game Theory: http://www.jianshu.com/p/e294d3f5237c
  - Game Theory II: http://www.jianshu.com/p/ee3f9a553cd2
  -
  - Other notes and study guides:
  -
  - Ty Abonil's notes: https://docs.google.com/document/d/1PFTsqUoP5ZrVadmZDBL3IVPM2yRwwV6LwvkSLLbrU1o/edit
  - From OMSCS's legacy Wiki: http://gtomscs.org/confluence/display/CS7641ML/CS7641.FA14.+Final+exam+prep
  - Michael Simpson's: https://libraries.io/github/mjs2600/ML-Final-Exam-Study-Notes
  - Dudon Wai's: https://github.com/dudonwai/dudonwai/blob/gh-pages/docs/CS7641-lectures-screenshots.docx
  -
  - **Lecture videos to review (all material since midterm):**
  -
  - Information Theory (20m)
  - Clustering (1h 18m)
  - Feature Selection (51m)
  - Feature Transformation (1h 23m)
  - Markov Decision Processes (2 hrs)
  - Reinforcement Learning (57m)
  - Game Theory (1h 51m)
  - Game Theory Continued (1h 40m)
  - Outro (27m)

Extracurricular Learning

- O notation
  - http://cglab.ca/~morin/teaching/2402/notes/bigoh.pdf
  - O(g(n)) is a set of functions that contains f(n)
- Eigenproblems
- Math of policies, converging
- Andrew Moore