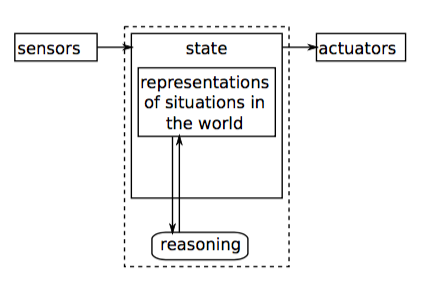
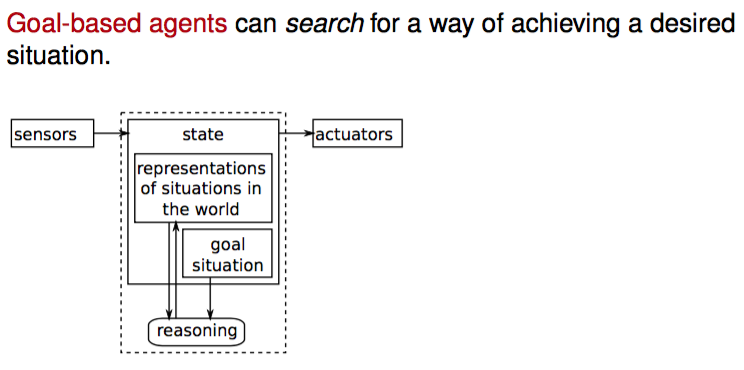
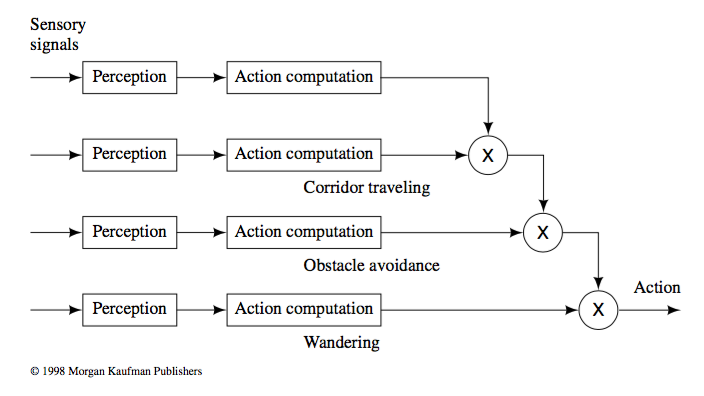
Reflex agent

The importance of environments as a complex equation of real world

#### What is AI?

Artificial Intelligence is the the study and creation of machines that perform tasks normally associated with intelligence.



Cross validation for over fitting

Decision tree. The algorithm to build decision tree. Entropy

The neural network and real neural network is different? Why?

How to encode Boolean values using 1 and 0

Turn un-number input or output into numbers to train

Algorithms for gradient descent

Need error terms for middle hidden layer, you only have error for last output, so we need back-prop

k-means clustering algorithms

Hierarchical algorithms?

All algorithm of state search, their comparison.

Admissible heuristics, never underestimate what? Important

# Note from Statistics

## Conditional probability

A certain disease affects 1% of the population, a certain test for this disease, the test is 100% show positive on an ill person, and 1% positive on a healthy person.

1. Randomly choose a person from population, give Jim a test, what is the probability of the test show positive?

1. A person’s test result shows positive, What is the probability that person has disease?

Suppose Event B has 4 cases:

## Independency

A family has 3 children, assume the 8 possible outcomes (male or female) are equally likely.

Let B be the event that both sexes occur, G is the event that female are majority. Are these two event independent?

# Quantifying Uncertainty

What probabilities are about?

Probabilistic assertions are about possible world. Probabilistic assertions and queries are not usually about particular possible worlds, but about sets of them. In probability theory, these sets are called events. In AI, the sets are always described by propositions in a formal language (see Chatper2.2).

For each proposition, the corresponding set contains just those possible worlds in which the proposition holds.

## Prior probability

Conditional or posterior probability given some event

## Probability distribution for Random Variable

Also, P() gives the values of P() for each possiblepair.

## Joint Distribution:

, if X is a vector of 4 numbers, and Y is a numbers of 2, then P() is a 2\*4 matrix. A probability model is completely determined by the joint distribution of all the random variables – the so called full joint probability distribution.

Probabilistic inference, that is the computation of posterior probabilities of query propositions given observed evidence.

Marginalization or Summing out:

P() =

A full distribution of the Toothache, Cavity and Catch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | toothache | | toothache | |
|  | catch | catch | catch | catch |
| cavity | 0.108 | 0.012 | 0.072 | 0.008 |
| cavity | 0.016 | 0.064 | 0.144 | 0.576 |

A general inference procedure, we begin with the case in which the query involves a single variable X(Cavity). Let be the list of evidence variables (Toothache), let be the list of observed values for them, and let Y be the remaining unobserved variables (Catch), the query is , can be evaluated as:

## Independence

Absolute independence or marginal independence:

## Bayes’ Rule for multivalued variables

We will also have occasion to use more general version conditionalized on some background evidence :

The general form of Bayes’ rule with normalization is:

, where is the normalization constant.

## Using Bayes’ rule combine with evidence:

This has the problem of **not scaling up**! However, we can use conditional independence which:

The general definition of conditional independence of two variable X, Y given a third variable Z, is:

Decomposition of joint distribution:

A commonly occurring pattern in which a single cause directly influences a number of effects, all of which are conditional independent, given the cause, the full joint distribution can be written as:

## 2015.5

1. Given,   
   So, They are conditional independent and not fully independent, because the fact of being a zombie changes the probability of A positive and changes the probability of B positive, but the two test are independent.

Make assumption that TestA=position and TestB=position for given zombie is conditional independent.  
  
What is Bayes’ network?

## 2014.2

1. The joint distribution is:  
    has 8 possible states, any questions has been asked must fall into it, as long as the questions we asked covers the 8 possible state, we can create this joint distribution.
2. So, I am more likely to be happy given I am a student rather than given I am a rich.
3. Rich and Happy are conditional independent given Student, if  
   Do I need to calculate the vector for all situation? Or just show an state/event?

# Machine Learning

# Supervised Learning

#### Supervised leaning definition:

Given training set of N example input-output pairs

Discover a function h that approximates the true function f.

Sometimes, f is not strictly a function of x, and what we have to learn is a conditional probability distribution, . When the output y is one of finite set of values, the learning problem is classification. Otherwise, when y is a number, the learning problem is called regression.

Notation:

, means the th training example

#### Hypothesis space

Hypothesis space is the set of all hypothesis that might be entertained from data set.

#### Consistent

A hypothesis is consistent if it can fit all the training data.

#### Ockham’s razor

When we consider multiple consistent hypothesizes, we use Ockham’s razor.

Tradeoff

Between complex hypothesis that fit the training data well and simpler hypothesis that may generalize better. However, a too simple hypothesis will make a learning problem unrealizable. It is realizable if the hypothesis space contains the true function.

#### Validation

Hold-out cross-validation

k-fold cross-validation

Peeking is the sequence of using test-set performance to both choose and evaluate a hypothesis.

#### 2015.1 Machine learning

1. Hypothesis space is the set of all possible hypothesis that might be entertained from data set.
2. The two dimension of this hypothesis is
3. What does each point in the hypothesis space represent?

Each point in hypothesis space represent a possible hypothesis.

1. This hypothesis function is a weighted sum of a set of base function, all we need to do is the find the which minimize the

This function defines the error surface in the weight space which is the space defined by all the possible settings of the weights.

1. As long as we can define the loss function in the form of weights, we can use gradient descent method to calculate the partial derivative associated with each , then we update each with a ratio. We choose any starting point in weight space, and then move to a neighboring point that is downhill, repeat until we converge on the minimum possible loss.
3. the hypothesis with higher order will minimize the error better, because, the order 3 hypothesis is greater and it has more chance to contain the true function, to make the learning problem more realizable.
4. We can use cross validation. We split the data into training data and testing data. We then use testing data to test if the hypothesis generalizes the training data well. If It has very low loss on training data and high loss on testing data, we know it is overfitting.

# Decision Tree

#### Definition

A decision tree represents a function that takes a vector of attributes as input, then return a single output value which is the decision.

A decision tree reaches its decision by performing a sequence of tests, and each node of the tree is a test on one of the input attributes. Each leaf node in the tree specifies a value to be returned by the function.

Decision-Tree-Learning algorithm adopts a greedy divide and conquer strategy: always test the most important attribute first. Each test divides the problem into smaller sub-problems that can be solved recursively. “Most important attribute” means the one that makes the most differences on the classification of an example.

**Function** ***Decision-Tree-Learning*** (examples, attributes, parent\_examples) **return** a tree

**If** example is empty then return PLURALITY-VALUE(parent\_examples)

**else if** all examples have the same classification then return the classification

**else if** attributes is empty then return PLAURALITY\_VALUE(examples)

**else**

IMPORTANCE ()

a new decision tree with root test

for each value of do:

add a branch to with label () and

**return** tree

**Entropy**:

If we define B(q) as the entropy of a Boolean random variable that is true with probability q:

# Regression and Classification

Regression is the problem of modelling a relationship between input variable of M dimensions, and **continuous output** variable  ,such that .

#### Linear Regression

A regression problem modelled with a hypothesis function that is **a weighted sum of a set of base functions** is called **linear regression**.

* The weights coefficients are the parameters of the model
* The model is linear in parameters
* The model can be non-linear in input

To fit a line to the data, all we have to do is find the values of the weight that minimize the empirical loss. It is traditional to use the square loss function, , summed over all the training examples:

**Weight space: the space defined by all possible setting of the weights.** The best vectors of weights , minimized squared-error loss **over the examples**:

The weight space is k dimensional in which each point represents a possible setting of weights, and the loss function maps each points in weight space to a real number, so the **error surface** is the visualization of Loss function in dimensional space.

A general problem in finding the minimum value in Weight space, using gradient descent. We choose any starting point in the weight space, and then move to a neighboring point that is downhill, repeating until we converge on the minimum possible loss:

<- any point in weight space

Loop until convergence do:

For each

The difference of batch gradient descent and stochastic gradient descent.

#### Linear classification

Linear function can also do classification; the task of classification is to learn a hypothesis that will take and return .

**We can think of as the result of passing the linear function through a threshold function**.

During regression, we solve the loss minimization by using closed form (setting the gradient to zero and solving for weights) or gradient descent in weight space. Here we can’t do that, because the Hard threshold. However, a simple weight update rule that converges to a solution.

We have said that the perceptron learning rule converges to a perfect linear separator when the data are linearly separable, **but what if they are not?** In general, the perceptron rule may not converge to a stable solution for a fixed learning rate , buf if decays as where is the literation number, then the rule can be shown to converge to a minimum-error solution when examples are presented in a random sequence.

**Perceptron Learning Rule, For a single example**:

We have seen that passing the output of a linear function through a threshold function creates a linear classifier; yet the hard nature of the threshold causes problem. We can resolve those problems by approximating the hard threshold function with a continuous, differentiable function.

With the logistic function replacing the threshold function, we get:

Noticing that the output, being a number between 0 and 1, can be interpreted as a probability of belonging to the class labeled 1. The hypothesis forms a soft boundary in the input space and gives a probability of 0.5 for any input at the the center of the boundary region, and approaches 0 and 1 as we move away from the boundary.

The process of fitting the weights of this model to minimize loss on a data set is called **logistic regression**. There is no easy closed-form solution, but we can use gradient descent computation on loss function to solve it.

So the weight update for minimizing the loss is

# Artificial Neural Network

Each unit first computes a weighted sum of its inputs:

Then it applies an activation function to this sum to derive the output:

The differences between feed-forward neural network and recurrent neural network:

A feed-forward network has connections only in one direction—that is, it forms a directed acyclic graph. Every node receives input from “upstream” nodes and delivers output to “downstream” nodes; there are no loops. A feed-forward network represents a function of its current input; thus, it has no internal state other than the weights themselves.

A recurrent network, on the other hand, feeds its outputs back into its own inputs. This means that the activation levels of the network form a dynamical system that may reach a stable state or exhibit oscillations or even chaotic behavior. Moreover, the response of the network to a given input depends on its initial state, which may depend on previous inputs. Hence, recurrent networks (unlike feed-forward networks) can support short-term memory.

#### Perceptron network (Single-layer neural network)

A network with all the inputs connected directly to the outputs is called **a single-layer neural network, or a perceptron network**.

Notice: **A perceptron network with m output is really m separate networks**, because each weight affects only one of the outputs. Thus, there will be m separate training process. Furthermore, depending on the type of activation function used, the training process will be either the perceptron learning rule (Equation 07) or gradient descent rule for logistic regression (Equation 08).

#### Multilayer feed-forward neural network

A single threshold unit can represent basic Boolean function AND, OR and NOT. So connecting large number of units into network will obtain any desired functionally.

A network is a function parameterized by the weights . Each output unit expressed as a function of the inputs and the weights. As long as we can calculate the derivatives of such expressions with respect to the weights, we can use gradient-descent loss-minimization method to train the network. Because the function represented by a network can be highly non-linear-composed, as it is, of nested nonlinear soft threshold functions, we can see neural network as a tool for doing **nonlinear regression**. In fact, with a single, sufficiently large hidden layer, it is possible to represent any continuous function of the inputs with arbitrary accuracy. With two layers, even discontinuous functions can be represented.

#### Learning in multilayer networks

A perceptron network with m output is really m separate networks, this decomposition is **not** working in multilayer network. Because each output unit depend on all of the input-layer weights, so updates to those weights will depend on errors on each all of the output units. However, this dependency is very simple in the case of any loss function that is additive across the components of the error vector . For the loss, we have, for any weight :

Each item in the final summation is just the gradient of the loss of the th output, computed as if the other output did not exist. Hence, we can decompose an -output learning problem into learning problem.

**Back-propagate**: The idea is that hidden node is “responsible” for some fraction of the error

in each of the output nodes to which it connects. Thus, values are divided according to the strength of the connection between the hidden node and the output node and are propagated back to provide the values for the hidden layer. The propagation rule for the values is:

Now the weight-update rule for the weights between the inputs and the hidden layer is essentially identical to the update rule for the output layer:

The back-propagation rule:

* Compute the values for the output units, using the observed error.
* Starting with output layer, repeat the following for each layer until the earliest hidden layer is reached:
  1. Propagate the values back to the previous layer.
  2. Update the weights between the two layers.

#### 2015.2

2. Filled circles, Class A is 1; empty circles, Class B is 0.
3. Activation function is
4. If the perceptron classifies the input example into the right class, then according to Perceptron learning rule, there is no update, otherwise

For example, if and , if the perceptron classifies it to black = 1, then

This make sense, the result make , because we want to make to be smaller, so that can correctly give out 0 instead of 1.

1. The perceptron represents a function of weighted sum of features, we can think about choosing the weights to minimize the loss.
2. This property is called linear separable, which means you can find a linear separator (a boundary line in this case) to separate the data into class perfectly.
3. A single perceptron can only do linear classification, if the data is not linear separable, then single perceptron will have failed on it.

#### 2014.1

2. It is a dummy value as bias
3. could be “Sun shining”
4. could be “Assignment due”
6. It is linear separable
7. Because a single perceptron can only represent a linear classifier. It the data is not linear separable, then it will fail to classify the data.

# Unsupervised Learning

#### Principal Component Analysis (PAC)

We are interesting in finding a mapping from the inputs in the ordinary -dimensional space to a new -dimensional space, with minimum loss of information.

The projection of on the direction of is

The criterion to be maximized is the variance. The principal component is such that the sample, after projection on to , is most spread out so that the differences between the sample points become most apparent.

The original data set is dimensional, we can find a “line” and on its direction the data has the least variance, then we can project the data set onto a dimensional subspace which is orthogonal to that line. Repeat this process, we can compress our -dimensional data onto dimensional space.

#### K-means Cluster

Input: set of points and

Place at random locations

Part1:

for each

calculate the distance , find the nearest centroid

assign

Part2:

For each

Repeat part1 and part 2 until the changes of distance between some threshold.

#### Auto-encoder Network

We can build a multilayer neural network in which its output layer is the same as its input layer with just one hidden layer. The number of units in hidden layer is smaller than input/output layer’s number of units . After training this network. The hidden layer actually learns the compression of input data. It maps data from -dimensional space onto k-dimensional space.

The method to train this network? Gradient descent?

How auto-encoder compare to the representation produced by PCA?

Encoder is compression and PCA is subtraction?

#### 2014.4

e. Suppose you have finished the clustering and a new data point arrives. Which of the existing clusters would you assign this new data point to? Describe the criteria and method you use.

New point p belongs to certain if the distance between p and is minimum among all distance .

# Genetic Algorithm and Optimization

#### Optimization

1. **Describe machine learning as an optimization problem**.
2. For classification: the optimization is to find a classification function which minimum the classification error according to some data set.
3. For linear regression, the optimization is to find the hypothesis function which can minimum the least-square error between hypothesis f and real f.
4. For SVM, the optimization is to find the classification which has maximum margin between support vector.
5. **Random Walk**
6. Write pseudocode for random walk search algorithm.

From initial position:

Randomly choose a neighbor whose state is higher than current state

Set current position to neighbor.

Keep updating until reach the maximum.

1. What is the drawback of this search algorithm?

Compare with steepest gradient method, it is slower than steepest gradient.

1. **Steepest gradient descent**
2. Write pseudocode for the steepest gradient descent search algorithm

From initial position:

choose a neighbor whose state is compare to current one has deepest gradient.

Set current position to neighbor.

Keep updating until reach the maximum.

1. What is the drawback of this search algorithm?

It does not guarantee to find global maximum state, stuck in local maximum state.

1. **Simulated annealing search**
2. Explain how the simulated annealing search algorithm works?

To solve this local maximum (or minimum) state problem, in the process of choose gradient descent:

Instead of choosing the best one, it picks a random one, if the move improves the state, then this move will always be accepted. Otherwise, this move will be accepted with a probability. This probability decreases exponentially with the extent of how badness the move it is and the probability also decrease as the execution time increases.

The algorithm stops eventually when the scheduled time reached.

1. Is it guaranteed to find the global minimum (or maximum)?

It doesn’t guarantee to find the global maximum state, but with enough long time, it can reach the global minimum with a probability which is approaching 1.

#### Genetic algorithms

1. **Explain how the genetic algorithm work**?
2. Randomly generate some ‘individual’ which is your first generation.
3. Using fitness function to evaluate those ‘individual’
4. Then chose pairs from those people. This choosing process can be based on the associated probability which is calculated from proportion on each individual’s fitness.
5. In each of those chosen pairs, randomly do crossover. (exchange a part of each other)
6. After crossover, there is a little chance randomly select one ‘bit’ in the individual and change it randomly (mutation).
7. Keep going with this process when they have offspring which satisfy our goal.
8. **Explain how a child is created from two parent using crossover and mutation**?

Each individual is represented as chromosome.

During the crossover, the selected parents exchange part of it with each other based on crossover point.

Then with some low probability, introduce a mutation in child (one bit in chromosome changed).

1. **What are the benefit and drawback of elitism**?

It can make the converge happened faster than random version. However, it will make the offspring have less diversity.

# Solving Problem by Searching

**Uninformed search algorithms**: algorithms that are given no information about the problem other than its definition. None of them can do so efficiently.

**Informed search algorithms**, on the other hand, can do quite well given some guidance on where to look for solutions.

Goal formulation, based on the current situation and the agent’s performance measure, is the first step in problem solving. We will consider a goal to be a set of world state – exactly whose states in which the goal is satisfied. The agent’s task is to find out how to act, now and in the future, so that it reaches a goal state. Before it can do this, it needs to decide what sorts of actions and states it should consider.

Uniformed search include depth first, breadth first and interactive depth first search.

Informed search algorithm include greedy search and search.

How to prove a heuristic is an admissible heuristic?

A heuristic is admissible when it is exactly the optimal solution cost for the relaxed version of original problem. Because, the optimal solution cost in relaxed version of problem will never be greater than the optimal solution cost of original problem. So, it must be an admissible heuristic.

#### 2015.4

1. In algorithm, each node in the fringe is expanded according to the fitness function

the path cost function is defined as the cost of reaching the node, gives the path cost from the start node to node n.

, the heuristic cost is defined as the cost from the node to the goal, is the estimated cost of the cheapest path from n to the goal.

1. Manhattan distance is an admissible heuristic. Because for moving in grid, if we allow the agent to move to adjacent square which is the relaxed version of problem, then the Manhattan distance is straightforward: it is the optimal solution cost for relaxed version of problem. And because the optimal solution cost for relaxed version is never greater than the optimal solution cost in real problem, then it must be an admissible heuristic.

it is the goal, stop.

BCFI

# Adversarial Search

1. Describe the Minimax algorithm for search in two-player games.

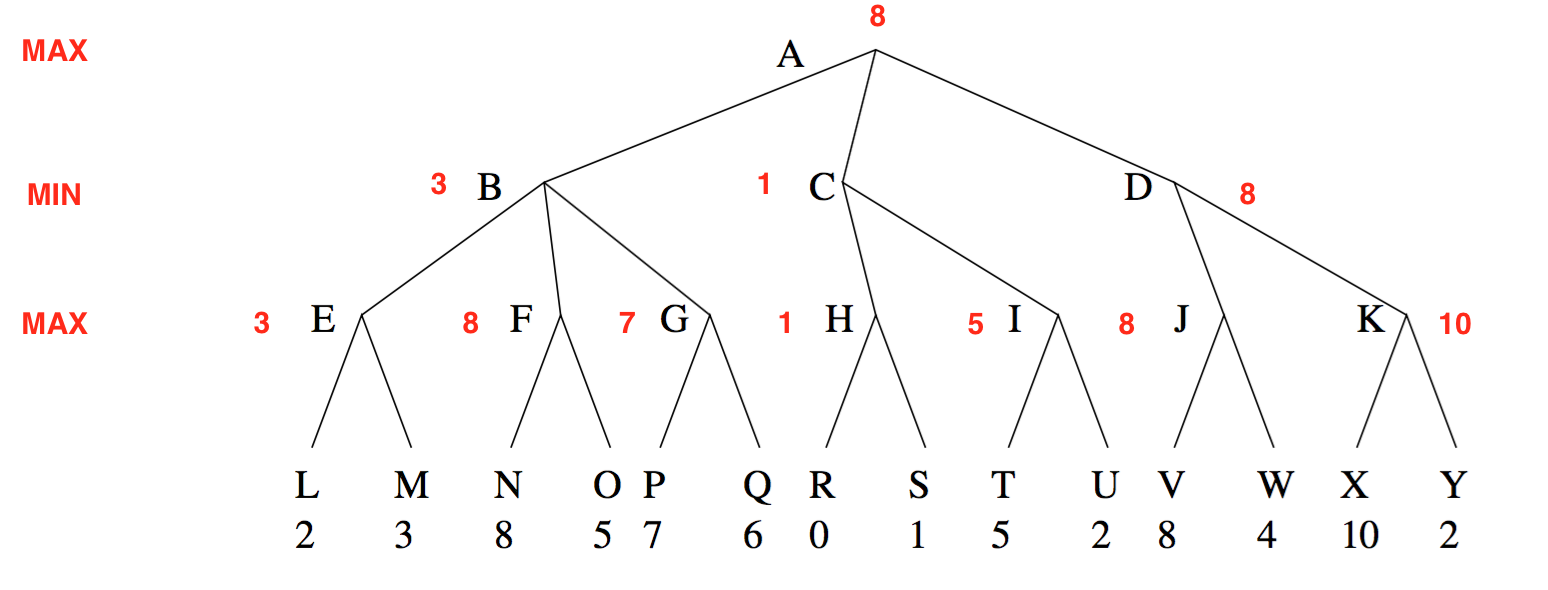
Suppose it is Max’s move at beginning:

1. Max takes the move that have the max value among possible move;
2. Each of these max values comes from the minimum value among each associated Min’s move.

These a), b) happened alternatively down to a state of ending game, then the utility value back up to the top.

It computes the Minimax decision from the current state. It uses recursive computation of Minimax values of each successor state. The recursion proceeds all the way down to the leaves of the tree, and then the Minimax values are backed up through the tree as the recursion unwinds.

1. At the end of the game it is Max’s move, so it chooses the max value, back up then:



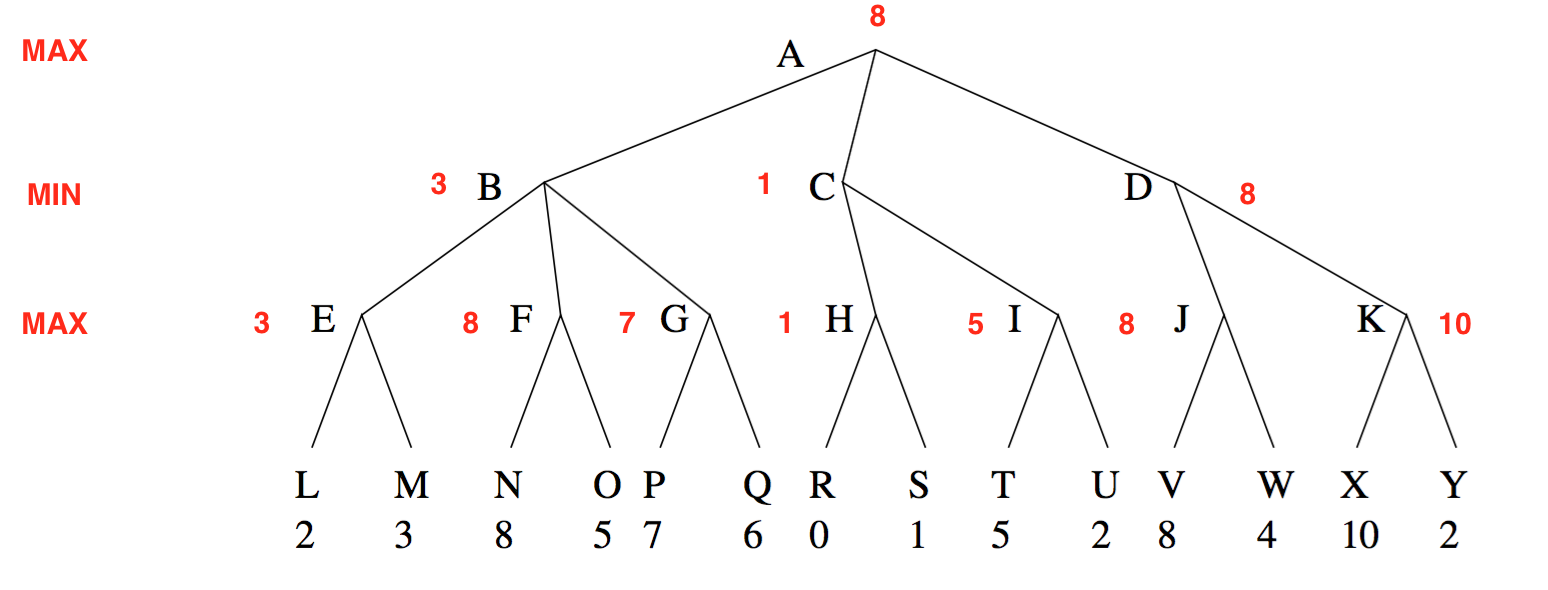
1. Use Minimax to determine which move that the first player should choose:

So, the first player should choose D.

1. What nodes would not need to be examined using the alpha-beta pruning algorithm,

Assuming that nodes are examined in left to right order.

From depth-first search:



{O, Q, U, W, Y} will not be checked.

# Natural Language

## n-gram models

Explain how an n-gram model can be trained to predict the next word in a corpus of text.

For a given corpus, we compute the probability distribution of P

(for n- gram), scanning through the whole article. Then for testing, for predicting some word , we compute the conditional probability of , and we select the one that has max value.

What problems arise if we build an n-gram model with a small n? (E.g. n = 1 or 2?)

With small n, when n = 1, we just compute the prior probability of each individual word. And the testing will appear that we just pick up the highest frequency word.

When n = 2, the connection is just the word and its previous potion word. The whole text generated will not make too much sense.

What problem arises if we build an n-gram model with a large n?

When n is large, we will meet the sparse data problem which means we don’t have enough data to estimate some very rare words. And this problem is particular severe because words are distributed according to Zipf’s law: some few words appears much frequently than other words.

How can an n-gram model be used to resolve word sense ambiguities?

Similar to use n-gram to predict next word, we can build our n-gram to train the ambiguity. During training, we need annotate those ambiguous words with right meaning, and compute the probability , During the testing, we select the max value of .

# Linear models of language with neural network

#### What is an Elman network? How does it differ from a regular back-prop network? Sketch it and explained how it trained?

An Elman network is simple recurrent network in which the internal state of the network is also depended on its previous state.

A regular back-prop network is back-prop the error of output to its previous layer. So the input unit which are connected with strongest weight will take more responsibility for the blame. So we can train the multiple-layer network. But the network is still a feed forward neural network which its internal state is not depended on its previous state.

#### How is the language model learned by Elman network similar to an n-gram model? How does it differ from n-gram model?

When using Elman network to do sentence generation, the last layer before output which is the next word is a probability distribution of all the words in the dictionary (using Soft-max function). Then the output is the word with highest probability. After the training, the hidden layer in Elman network learned to represent the aspects of previous n words that are useful to predict the next word. It is basically learning , similar to n-gram. However, the n is not fix in Elman network training and the training is automatic without teaching.

#### 2015.5