1 Intro to Decision Theoretic Agents and Probability Theory

1. **What is the focus of decision theory?**
   a. What are the basic steps of a decision theoretic agent?
      i. Decision Theory = Probability Theory + Utility Theory
      ii. A decision theoretic agent will select the action that yields the highest expected utility, averaged over all possible outcomes of the action (Maximum Expected Utility principle, MEU).
      iii. Steps:
         1. Update a belief-state distribution based on actions/percepts (answering the question: what is the most probable state the world is currently in?)
         2. Calculate outcome of actions given action descriptions + current belief state (compute the utility of each action)
         3. Select action with the highest expected utility given probabilities of outcomes and utility information
         4. Return action
2. **You might be asked to provide any of the rules of probability theory discussed in the class:**
   a. Kolmogorov's axioms:
      i. $0 <= P(a) <= 1, \forall a$
      ii. $P(true) = 1, P(false) = 0$
      iii. $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$
   b. Product rule - with and w/o normalization,
      i. $P(a \mid b) = P(a \wedge b) / P(b)$
      ii. w/ normalization : $P(a \mid b) = \alpha * P(a \wedge b)$
   c. Bayes rule - with and w/o normalization,
      i. $P(b|a) = P(a|b)P(b)/ P(a)$
      ii. $< P(b|a), P(\neg b|a) >=< \alpha P(a|b)P(b), \alpha P(a|\neg b)P(\neg b) >$

d. marginalization and conditioning

i. marginalization: given P(x, y) = P(X ^ y), P(x) = $\sum_{y\ value\ of\ Y} P(x,\ y)$

ii. conditioning (marginalization combined with product rule): P(x) =

$\sum_{y\ value\ of\ Y} P(x|y)P(y)$,

iii. general rule of conditioning: P(x|z) = $\sum_{y\ value\ of\ Y} P(x,\ y|z)P(y|z)$

e. independence and conditional independence properties.

i. Independence: P(a|b) = P(a) or P(b|a) = P(b) or P(a ∧ b) = P(a)P(b)

ii. Conditional Ind.:

1. P(x, y|z) = P(x|z)P(y|z)
2. P(x|y, z) = P(x|z)
3. P(y|x, z) = P(y|z)

3. You might be asked to compare the value of P(a|b) and P(¬a|b) in a way that minimizes the number of probabilities that we have to be aware of.

1. P(a) / P(¬a) P(a|b) = P(b|a) P(a) / P(b)
2. P(b|a) P(¬a|b) = P(¬a|b)P(b) / P(b)
3. P(b|¬a) P(¬a|b) + P(a|b) = 1, use this to solve for
P(b)

4. Assume two random binary variables α, β.

a. *What is the minimum number of distinct probabilities that you need to know to compare P(α|β) and P(¬α|β) using the Bayes rule*? [Clarification of "distinct": if you know P(γ) then you also know P(¬γ) = 1 − P(γ).] **3 (see past Midterm solution on Sakai under Resources)**

2 Bayesian Networks: Properties and Exact Inference

1. What is the definition of a Bayesian network?

   A graphical way to represent dependencies between variables and hopefully allow us to detect interdependencies.

2. Why can we answer any query in a probabilistic domain given the Bayesian network that involves all the variables in this domain and their independence properties?

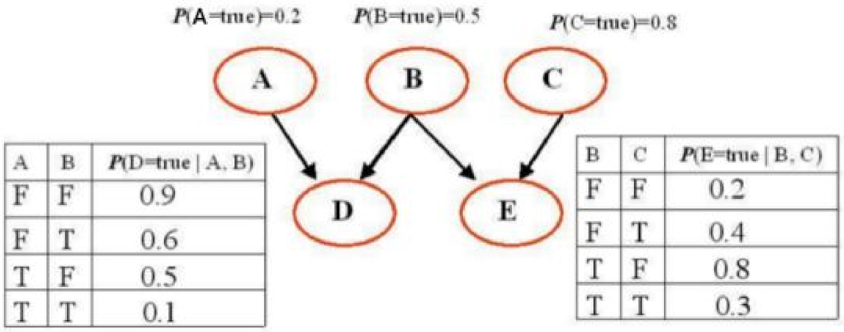
   Because it is a representation of dependencies among variables, thus we can calculate all dependencies and

   probabilities that we need from given variables and properties.

3. How can we compute the joint probability distribution $P(X1, \dots, Xn)$ from the conditional probabilities $P(X|Parents(X))$ stored on the Bayesian network that involves variables $X1, \dots, Xn$?

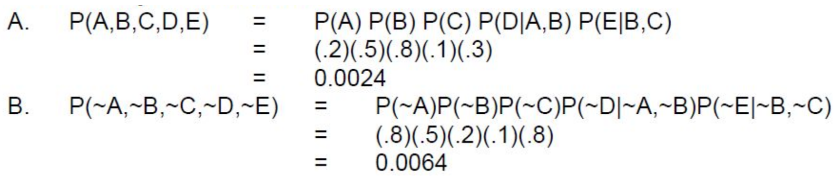
   Multiply all the probability? Using Chain Rule: P(X_1, … , X_n) = P(X_n | X_n-1, …, X_1)*P(X_n-1|X_n-2...X_1)...P(X_2|X_1)*P(X_1) = $\prod_{i=1}$ (P(X_i|X_i-1,...,X_1)

4. You may be provided with a Bayesian network and asked to compute the joint probability distribution.
   a. You can also then be asked to compute conditional probabilities that involve the variables of the Bayesian network. The conditional probability might:OK.
      i. • Involve all the variables in the Bayesian network



A	B	P(D=true | A, B)
F	F	0.9
F	T	0.6
T	F	0.5
T	T	0.1

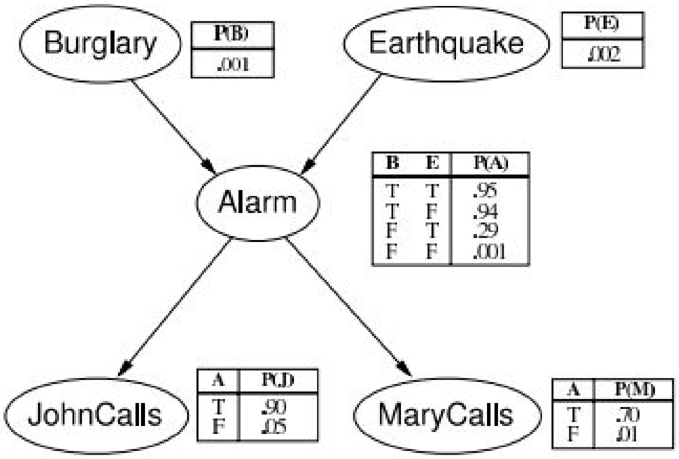
B	C	P(E=true | B, C)
F	F	0.2
F	T	0.4
T	F	0.8
T	T	0.3



A. P(A,B,C,D,E) = P(A) P(B) P(C) P(D|A,B) P(E|B,C)
= (.2)(.5)(.8)(.1)(.3)
= 0.0024

B. P(~A,~B,~C,~D,~E) = P(~A)P(~B)P(~C)P(~D|~A,~B)P(~E|~B,~C)
= (.8)(.5)(.2)(.1)(.8)
= 0.0064

ii. • Involve a subset of the variables in the Bayesian network



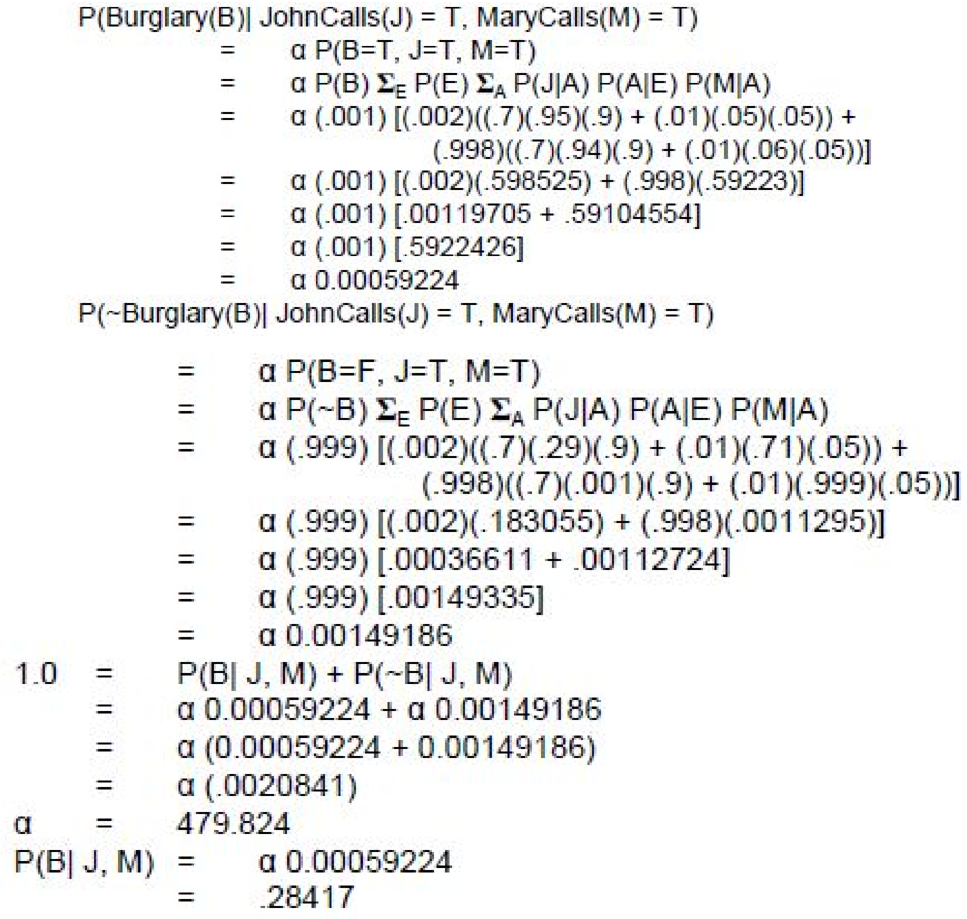
P(B)
.001

P(E)
.002

B	E	P(A)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J)
T	.90
F	.05

A	P(M)
T	.70
F	.01



P(Burglary(B)| JohnCalls(J) = T, MaryCalls(M) = T)

= α P(B=T, J=T, M=T)

= α P(B) Σ_E P(E) Σ_A P(J|A) P(A|E) P(M|A)

= α (.001) [(.002)((.7)(.95)(.9) + (.01)(.05)(.05)) + (.998)((.7)(.94)(.9) + (.01)(.06)(.05))]

= α (.001) [(.002)(.598525) + (.998)(.59223)]

= α (.001) [.00119705 + .59104554]

= α (.001) [.5922426]

= α 0.00059224

P(~Burglary(B)| JohnCalls(J) = T, MaryCalls(M) = T)

= α P(B=F, J=T, M=T)

= α P(~B) Σ_E P(E) Σ_A P(J|A) P(A|E) P(M|A)

= α (.999) [(.002)((.7)(.29)(.9) + (.01)(.71)(.05)) + (.998)((.7)(.001)(.9) + (.01)(.999)(.05))]

= α (.999) [(.002)(.183055) + (.998)(.0011295)]

= α (.999) [.00036611 + .00112724]

= α (.999) [.00149335]

= α 0.00149186

1.0 = P(B| J, M) + P(~B| J, M)

= α 0.00059224 + α 0.00149186

= α (0.00059224 + 0.00149186)

= α (.0020841)

α = 479.824

P(B| J, M) = α 0.00059224

= .28417

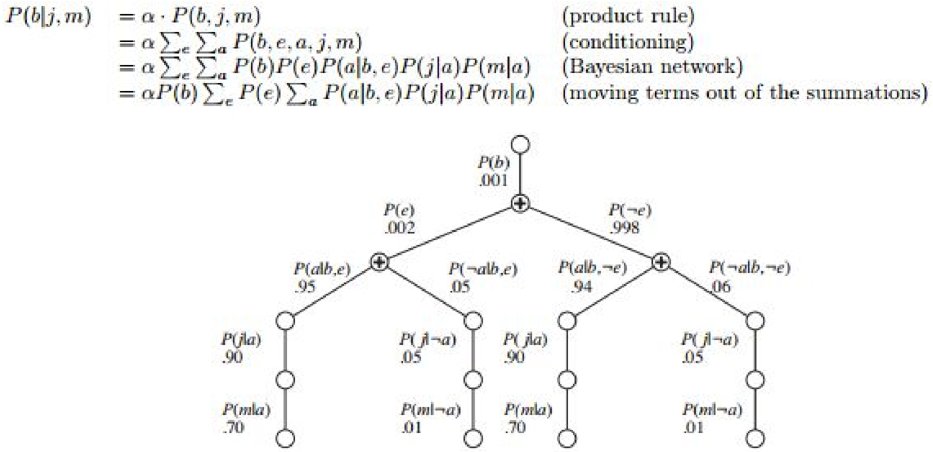
5. How can we compute exactly a conditional NIGGER of the form *P*(*A*|*B*) (without a normalization factor) from full joint probability distributions of the form: *P*(*A, B*) and *P*(¬*A, B*).

P(A|B)=P(A^B)/P(B)

Since P(B)=P(A^B) + P(~A^B)
Thus P(A|B)=P(A^B)/(P(A^B) + P(~A^B))

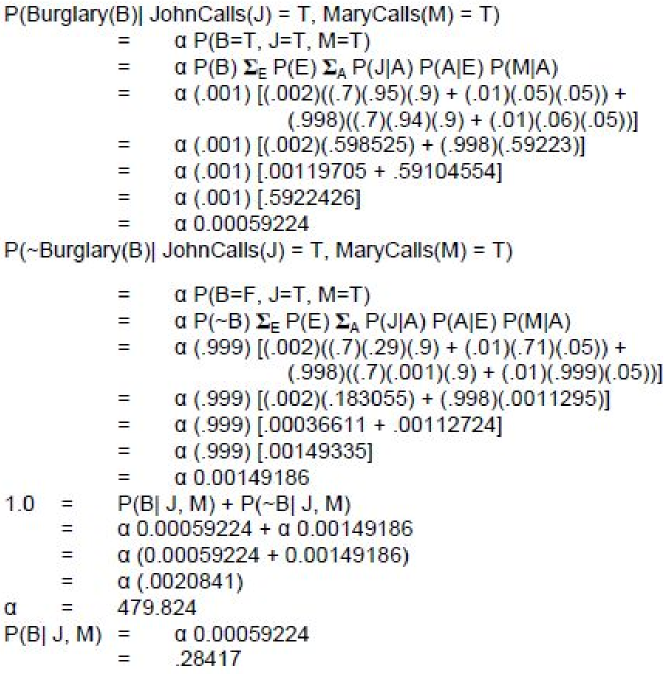
Explanation:
We want *P*(*A*|*B*), but have *P*(*A, B*) and *P*(¬*A, B*).
Conditional probability tells us that *P*(*A*|*B*) = P(A, B)/P(B).
We have P(A, B) but not P(B).
In order to get P(B) we can add up *P*(*A, B*) and *P*(¬*A, B*).
Then substitute P(B) for (P(A^B) + P(~A^B)) and we get
P(*A*|*B*) = P(A, B)/(P(A^B) + P(~A^B))

6. How does exact inference by enumeration work in Bayesian networks? Just applying the marginalization rule
    a. You can be given a Bayesian network (e.g., such as the Burglary-Alarm problem) and asked to compute a conditional probability by applying exact inference.



$$\begin{aligned} P(b|j,m) &= \alpha \cdot P(b,j,m) && \text{(product rule)} \\ &= \alpha \textstyle\sum_e \sum_a P(b,e,a,j,m) && \text{(conditioning)} \\ &= \alpha \textstyle\sum_e \sum_a P(b)P(e)P(a|b,e)P(j|a)P(m|a) && \text{(Bayesian network)} \\ &= \alpha P(b) \textstyle\sum_e P(e) \sum_a P(a|b,e)P(j|a)P(m|a) && \text{(moving terms out of the summations)} \end{aligned}$$

7. What are the basic ideas behind Variable Elimination? To reduce the amount of niggers necessary
    a. How is it advantageous over Inference by enumeration? Eliminates performing same computations in comparison to enumeration inference
    b. You might be provided a Bayesian network and asked to compute a conditional probability by applying Variable Elimination.



P(Burglary(B)| JohnCalls(J) = T, MaryCalls(M) = T)
= α P(B=T, J=T, M=T)
= α P(B) Σ_E P(E) Σ_A P(J|A) P(A|E) P(M|A)
= α (.001) [(.002)((.7)(.95)(.9) + (.01)(.05)(.05)) + (.998)((.7)(.94)(.9) + (.01)(.06)(.05))]
= α (.001) [(.002)(.598525) + (.998)(.59223)]
= α (.001) [.00119705 + .59104554]
= α (.001) [.5922426]
= α 0.00059224

P(~Burglary(B)| JohnCalls(J) = T, MaryCalls(M) = T)
= α P(B=F, J=T, M=T)
= α P(~B) Σ_E P(E) Σ_A P(J|A) P(A|E) P(M|A)
= α (.999) [(.002)((.7)(.29)(.9) + (.01)(.71)(.05)) + (.998)((.7)(.001)(.9) + (.01)(.999)(.05))]
= α (.999) [(.002)(.183055) + (.998)(.0011295)]
= α (.999) [.00036611 + .00112724]
= α (.999) [.00149335]
= α 0.00149186

1.0 = P(B| J, M) + P(~B| J, M)
= α 0.00059224 + α 0.00149186
= α (0.00059224 + 0.00149186)
= α (.0020841)
α = 479.824
P(B| J, M) = α 0.00059224
= .28417

(Besides, below are the answer from Assignment III):

$$P(B|j,m) =$$

$$= \alpha P(B)\sum_{e} P(e)\sum_{a} P(a|b,e)P(j|a)P(m|a)$$

$$= \alpha P(B)\sum_{e} P(e)[.9 \times .7 \times \begin{pmatrix} .95 & .29 \\ .94 & .001 \end{pmatrix} + .05 \times .01 \times \begin{pmatrix} .05 & .71 \\ .06 & .999 \end{pmatrix}]$$

$$= \alpha P(B)\sum_{e} P(e)\begin{pmatrix} .598525 & .183055 \\ .59223 & .0011295 \end{pmatrix}$$

$$= \alpha P(B)[.002 \times \begin{pmatrix} .598525 \\ .183055 \end{pmatrix} + .998 \times \begin{pmatrix} .59223 \\ .0011295 \end{pmatrix}]$$

$$\alpha \begin{pmatrix} .001 \\ .999 \end{pmatrix} \times \begin{pmatrix} .59224259 \\ .0011493351 \end{pmatrix}$$

$$\alpha \begin{pmatrix} .00059224259 \\ .00149118576 \end{pmatrix}$$

$$< .284, .716 >$$

3 Approximate Inference in Bayesian networks

1. Which techniques do you know for approximate inference in Bayesian networks?
    a. What is the basic idea/methodology behind approximate methods for inference in Bayesian networks?

Direct sampling (rejection sampling,likelihood weighting) and Markov Chain Simulation

The basic idea behind these techniques is that if you sample an atomic event multiple times (assignment of variables), then as you continue sampling, the ratio of number of times the sampling matches an atomic event over the number of samples taken converges to the probability of that atomic event.

2. How does direct sampling work?
    a. You might be provided with a Bayesian network and a series of "random" numbers and asked to sample an atomic event by employing direct sampling.

You sample each random variable in topological order according to the conditional probabilities given by the bayesian network. Count the number of times each sampling returns assignments to the variables that are the same as the atomic event being queried and divide that number by the number of samples taken.

3. What direct sampling algorithms do you know to compute conditional probabilities?

Rejection sampling, likelihood weighting

    a. What are the basic ideas behind them?
    b. How do they work?

rejection sampling

Sample like you would with direct sampling.

Then reject the samples where the value of evidence does not correspond to query evidence value

From this subset, then calculate:

<Num samples(X = true, e) / all e samples, Num samples (X = false, e) / all e samples>

This method wastes time sampling the samples that do not correspond with evidence variable value.

Also rejects too many samples.

likelihood weighting

Fix the evidence variables E and only sample the remaining variables X and Y. Each sample stores a weight which is the probability that sample could have been produced if the evidences were not fixed. Weights start at 1.

Sample like in direct sampling, but for each evidence variable, we update the w so that

w = w * P(evidence variable value | parents sample value)

Thus we end up with a sample and a weight.

c. You might also be provided a network and "random" numbers and asked to imitate the operation of these algorithms?

4. What is the Markov Blanket of a random variable in a Bayesian network?

A set of variables that are basically all neighbors (including parents, children, and children's parents) of Xi, that we need to take into account while evaluating Xi.

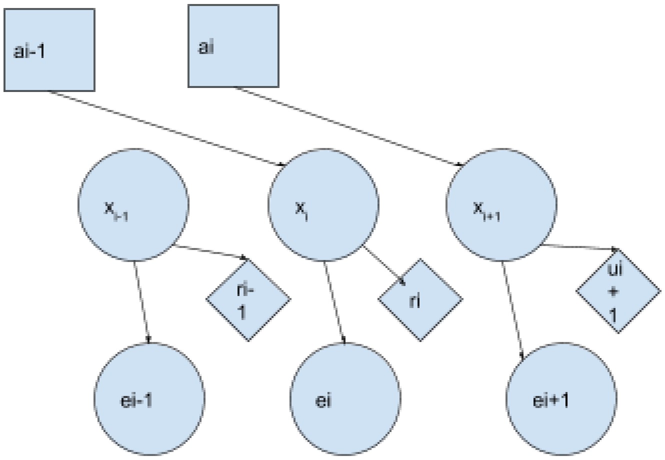
a. Why is it important?

It considers more dependencies and will result a more accurate probability for Xi?

Maybe because a node is conditionally independent of all other nodes in the network, given its Markov blanket.

---

4 Dynamic Bayesian Networks: Temporal State Estimation



Look ahead using the utility - based on this belief and this states you compute the expected utility. Heuristic is the point where you stop and look ahead

Wait, I think this is not a Dynamic Bayesian Network but Decision Network

1. **What assumptions do we typically employ in a Dynamic Bayesian network? Give a graphical representation of a network that involves state and evidence variables.**

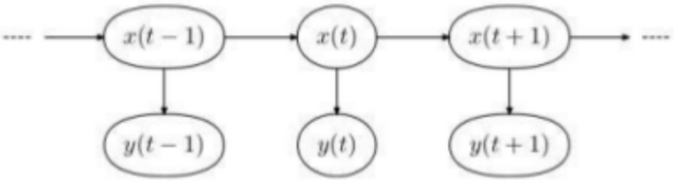


Figure 1: Dynamic Bayesian Network for Temporal State Estimation. In this figure the variable y corresponds to the evidence variables.

   a. Typical assumptions include:
      i. Markov Assumption: the current state depends only on a finite history of previous states.
         1. 1st order Markov depends only on the last state .
      ii. Stationary Process: the transition and the observation model do not change over time.

2. **What problems can we answer by employing temporal models (Dynamic Bayesian Networks)? Provide the mathematical representation of these problems.**

a. Problems where the states of an agent have to be estimated under uncertainty in sensing and action.

b. Math representation: X_t: unobservable state at time t; E_t: observable evidence at time t

$$P(X_0, X_1, \ldots, X_t, E_1, \ldots, E_t) = P(X_0)\prod_{i=1}^{t} P(X_i|X_{i-1})P(E_i|X_i)$$

3. What information/input should we have available in order to solve a problem with a Dynamic Bayesian Network?

a. Initial Probability $P(X_0)$

b. Transition Model $P(X_t \mid X_{t-1})$

c. Observation Model $P(E_t \mid X_t)$

4. Derive the filtering equation in Dynamic Bayesian Networks.

a. Equation:

$$P(X_{t+1}|e_{1:t+1}) = \alpha \underbrace{P(e_{t+1}|X_{t+1})}_{\text{observation model}} \sum_{X_t} \underbrace{P(X_{t+1}|X_t)}_{\text{transition model}} \underbrace{P(X_t|e_{1:t})}_{\text{prior belief}}$$

5. What is the filtering equation? Explain its various elements.You can be provided with a small temporal state estimation problem and asked to apply filtering for one or two steps in order to update the belief distribution. You will be provided a transition, an observation model and an initial probability distribution. For example, consider the rain-umbrella example provided in the notes, or a robot moving in a grid world with obstacles.

a. Filtering computes the current belief state given all evidences so far.

b. Contains a transition model, observational model, and prior belief model.

6. What is the backwards message used for solving the smoothing problem in temporal models?

7. Which state estimate do you expect to be more accurate, a filtering estimate, a smoothing estimate or a predictive estimate? Smoothing because it incorporates more evidence

8. Assume that for a specific system the predictive estimate is more accurate than the filtering estimate? What can you infer about the properties of this system for which we are trying to estimate its state?

9. What is the difference between the “most likely explanation” problem formulation in temporal models compared to filtering, smoothing and prediction?

The “most likely explanation” problem formulation in temporal models compared to filtering, smoothing and prediction is that the query is to return an entire sequence of states.

10.

   a. What is the advantage of the “most likely explanation” formulation over executing smoothing for all the states visited so far?

Filtering = p(xi | E1 to i). If you knew the most likely sequence for all possible then you can compute
Argmax P(x1 to i | e1 to i) = argmax xi p(ei|xi) * argmax P(xi | x i+1) p(x 1 to i-1 | E 1 to i-1)
You have the trajectory then compute each trajectory for xi and then weight then given previous evidence

The advantage of the “most likely explanation” formulation over executing smoothing for all the states visited so far is that it uses a linear-time algorithm while considering joint probabilities over all the time steps. Posterior distributions computed by smoothing are distributions over single time steps, whereas to find the most likely sequence we must consider joint probabilities over all the time steps.

5 Continuous Temporal State Estimation Problems: Kalman and particle filtering

1. **What are the typical approaches for dealing with temporal state estimation problems that involve continuous variables?**
   - i. One approach is to discretize the environment into a grid. If the world can be easily discretized, then we can apply this process.
     1. we get considerable errors in the state estimation process by discretizing.
   - ii. Another approach is to represent the data of the world in such a way that is memory efficient and quick to reference. If we have available a probability density function with such desirable qualities, then we can keep the required number of parameters to represent the probabilities low.
2. **What is the advantage of using a Gaussian distribution to model a continuous temporal state estimation problem? How many numbers do you need to represent an *n*-dimensional state with a Gaussian distribution?**
   - i. The advantage of a Gaussian distribution is that it can be represented (for multi-dimensional problems) just by its mean vector μ and its covariance matrix Σ.
3. **What are the requirements so that the Kalman filter is the optimal solution to the Bayesian filtering problem?**
   - a. The current distribution is Gaussian and the transition model is linear Gaussian.
   - b. If the predicted distribution is Gaussian and the observation model is linear Gaussian.
4. **Under which circumstances does the Kalman filter fail? What approaches can be used to address these challenges?**
   - a. Fails when:
     - i. The underlying processes are nonlinear.

    ii. Multi-modal distributions.

  b. Approaches that can be used include:

    i. Extended Kalman filter

    ii. Gaussian Sum Filter (“switching” Kalman filter)

    iii. Particle Filter

5. **Describe the basic particle filtering algorithm. What is the property provided by a particle filter in terms of computing the correct probability distribution?**

  a. Algorithm goes as follows:

    i. A population of N samples is constructed by sampling from the prior distribution.

    ii. Then the following update cycle is repeated for each time step:

      1. Use transition model to propagate each sample forward by sampling the next state using current.

      2. Use observation model to apply weight to each sample by likelihood it assigns to new evidence.

      3. The population is resampled to generate a new population of N samples. The probability that a sample is selected, is proportional to its weight. The new samples produced are assigned equal weights.

  b. The computation becomes more accurate as the number of particles increases

6 Utility Theory

1. **What is the basic principle of utility theory?**
    a. The basic principle of utility theory is that there is a single # which expresses the desirability of the state. An agent wants to maximize its utility by choosing actions that give it the maximum utility.
2. **How can we compute the expected utility of an action A in a probabilistic setup (probabilistic transition model) given evidence variables?**
    a. Each action has outcome states $\text{Result}_i(A)$.
    b. Each outcome has a probability assigned to it:
        i. $P(\text{Result}_i(A)|\text{Do}(A), E)$
    c. Expected utility is the sum of all probabilities of actions multiplied by their respective utility:
        i. $EU(A|E) = \Sigma_i\ P(\text{Result}_i(A)|\text{Do}(A), E)\ U(\text{Result}_i(A))$
3. **Specify the rules of utility theory regarding preference of lottery outcomes.**
    a. Orderability - given any two lotteries, a rational agent must either prefer one to the other or else rate the two as equally preferable.
    b. Transitivity - given any three lotteries, if an agent prefers A to B and prefers B to C, then the agent must prefer A to C
    c. Continuity - if some lottery B is between A and C in preference, then there is some probability *p* for which the rational agent will be indifferent between getting B for sure and the lottery that yields A with probability *p* and C with probability 1-*p*
    d. Substitutability - if an agent is indifferent between two lotteries A and B, then the agent is indifferent between two more complex lotteries that are the same except that B is substituted for A in one of them. This holds regardless of the probabilities and the other outcome(s) in the lotteries

e. Monotonicity - suppose two lotteries have the same two possible outcomes, A and B. If an agent prefers A to B, then the agent must prefer the lottery that has a higher probability for A (and vice versa)

f. Decomposability - Compound lotteries can be reduced to simpler ones using the laws of probability.

4. **What does the "no fun in gambling" rule of utility theory specify? Why does it have this name, what is its meaning?**

    a. If you are given a choices of playing multiple lotteries it's the same a playing one lottery with the same outcomes .

    b. Compound lotteries can be simplified using the laws of probability. This means that two consecutive lotteries can be combined into one equivalent lottery.

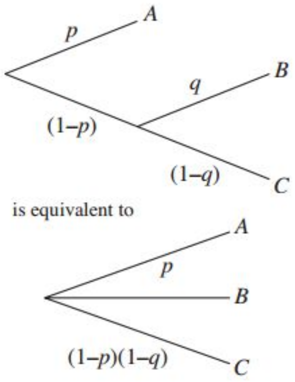


Figure 4: The "no fun in gambling" rule of utility theory.

5. **What are the effects of risk aversion in the behavior of a utility function? Can you provide an example relating to the utility of money?**

    a. Risk-averse agents prefer a sure payoff that is less than the expected monetary value of the gamble.

        i. Consider the case where you can either take $1,000,000 or gamble between winning $0 and $3,000,000 by flipping a coin. So our expected monetary value of the gamble looks like this:

            1. 0.5 · ($0) + 0.5 · ($3, 000, 000) = $1, 500, 000

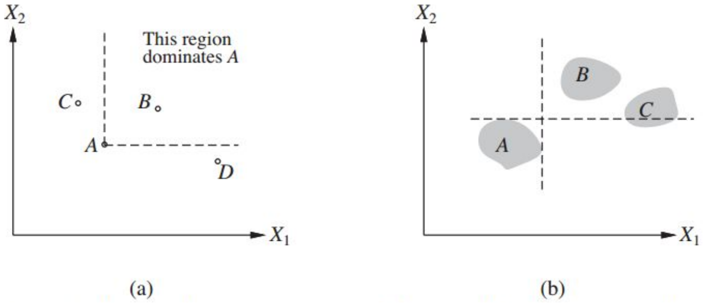
ii. A risk-averse agent would take the 1 mil. instead of flipping the coin even though the expected utility is 1.5 mil.

6. How can we deal with multi-attribute utility functions?

a. We can eliminate options that are strictly dominated by any other option.

i. Suppose that an airport site S1 costs less, generates less noise, and is safer than an airport site S2. Since S2 is worse on all attributes, it does not even need to be considered. S1 has a strict dominance over S2 in this case. Strict dominance can be useful in narrowing choices.

ii. Section (a) shows a deterministic case and section (b) shows an uncertain case. In section (a), Option A is strictly dominated by B, but not by C or D. In section (b) of the figure, A is strictly dominated by B, but not by C. Here Option B would be S2 in our airport example.



7 Decision Networks and Value of Information

1. **How is a decision network different than a Bayesian network? Describe the purpose of the additional nodes.**
    a. Decision networks combine Bayesian networks with additional nodes for types of actions and utilities.
        i. Chance Nodes(ovals): represent the possible attributes that affect the problem's state.
        ii. Decision Nodes(rectangles): correspond to the different actions that the agent can take.
        iii. Utility nodes(diamonds): used to show which attributes affect the utility.
2. **Describe the algorithm for computing the best action given a decision network.**
    a. Set evidence variables for the current state .
    b. For each possible value of the decision node.
        i. Set decision node to that value.
        ii. Calculate posterior probabilities for the state nodes that are parents to the utility node.
        iii. Compute expected utility for this action.
    c. Return action that maximizes the expected utility
3. **How can we compute the value of information that we can acquire in order to make a decision?**
    a. $$VPI_E(E_j) = (\sum_k P(E_j = e_{jk}|E) * E(U)(a_{E_{jk}}|E, E_j = e_{jk})) - EU(a|E)$$
4. **You might be given an example similar to the "oil company - seismologist example" in class and asked to compute the value of a piece of information.**
    a.

Consider the following example: An oil company has n indistinguishable blocks of ocean drilling rights.

- Only one of them will contain oil worth \$C
- The price of each block is \$$\frac{C}{n}$

A seismologist offers to survey block 3 and will definitely indicate whether the block has oil or not. How much should the company pay the seismologist? In other words, what is the "value of information" that the seismologist offers?

There are two outcomes when using the seismologist:

1. $\exists$ oil in block 3, which has a probability of $\frac{1}{n}$
   Then the best action is to buy block 3
   Because we want profit (C) - cost ($\frac{C}{n}$), the total profit is $C - \frac{C}{n} = (n-1)\frac{C}{n}$
2. Does not $\exists$ oil in block 3, which has a probability of $\frac{n-1}{n}$
   Then the best action is to buy a block other than 3
   Because expected profit is $\frac{C}{n-1}$ and since we choose among n-1 blocks, cost is $\frac{C}{n}$ of buying a block.
   The total profit is therefore $\frac{C}{n-1} - \frac{C}{n} = \frac{C}{n(n-1)}$

The expected utility in the case that we use the seismologist is:

$$EU = \frac{1}{n} * \frac{(n-1)C}{n} + \frac{n-1}{n} * \frac{C}{n(n-1)} = \frac{C}{n}$$

If we do not use the seismologist:

$$EU = \text{ expected profit - cost } = \frac{C}{n} - \frac{C}{n} = 0$$

8 (Partially Observable) Markov Decision Processes

1. **What do we need to define in order to formulate a Markov Decision Process?**
    a. Transition model: T(s, α, s')
    b. Rewards: R(s) - We typically assign some positive value to the desired goal state (e.g., +1 in the example), and some negative value to a finish node that is undesirable (e.g., -1 in the example). In addition to this, we assign a small negative value to each cell that the agent visits in hopes to entice the agent to find the goal quickly (assume -0.04 for each cell in the example).
    c. Initial state: s_0
2. **What do we have to compute in order to solve a Markov Decision Process?**
    a. Compute an optimal policy, a policy that yields the highest expected utility.
3. **How do we compute utilities of state sequences (i.e., paths)?**
    a. Simply add up the rewards of the state along the path:
        i. U([s0, s1, ...]) = R(s0) + R(s1) + ...
    b. Use discounting: (for some γ such that 0 < γ < 1)
        i. U([s0, s1, ...]) = R(s0) + γR(s1) + γ 2R(s2) + ...
4. **How does value iteration work? What is the main idea behind the algorithm?**
    a. Calculates error from actual policy on every iteration to eventually converge to optimal solution.
    b. Idea: Calculate the utility of each state then use the state utilities to select an optimal action in each state. The utility of a state is the expected utility of the state sequences that might follow it.
5. **Derive the Bellman equation and describe the value iteration algorithm.**
    a. Bellman Equation:

$$U(s) = R(s) + \gamma * max_{\alpha} \sum_{s'} T(s, \alpha, s') * U(s')$$

b. Value Iteration Algorithm:

i. Make an initial assignment.

ii. Calculate the right hand side of the Bellman equation.

iii. Plug the values into the Left hand side: Ui+1(s) = R(s) + γ ∗ maxP s0 T(s, α, s0) ∗ Ui(s 0)

iv. Repeat until you reach equilibrium

6. What are the properties of value iteration?

a. The algorithm always converges to an optimal solution

b. At each iteration, you always know your error from the actual solution

c. It might take a very long time

7. How does policy iteration work? What is the main idea behind the algorithm? Describe the algorithm. What is the algorithm's advantage over value iteration?

a. The inspiration for policy iteration comes from the fact that when the utilities at the states are slightly inaccurate, then the optimal policy tends to be the same.

b. Given a policy π_i, calculate $U_i = U^{\pi i}$, the utility of each state if π_i were to be executed. Then given U_i, we can calculate a new policy π_{i+1} that maximizes expected utility using one-step look-ahead.

c. The advantage of policy iteration is that once you assume the first policy, computing the utilities for the cells is easier than in value iteration. Instead of the operand max in the equations, we have the operand Σ, which implies that the equations are now linear in nature.

8. You might be provided a small Markov Decision Process and asked to solve it by applying either policy iteration or value iteration.

a.

9. What do we need to define in order to formulate a Partially Observable Markov Decision Process? What makes POMDPs a challenging problem?

a. To define we must have:

i. Initial probability distribution: b_0

ii. Transition model: T(s, a, s_0)

iii. Rewards function: R(s)

iv. Observation model: O(s, o)

b. Challenging because:

i. The uncertainty in observations implies that the agent does not know exactly its state. Similarly, the agent is not able just to execute an action π(s) for state s, because there is no certainty that this is the correct state of the agent.

10. Describe how to turn a POMDP into an MDP over a belief state space.

a. First we must compute a new belief-level transition model.

b. Then we have to define how the reward function is computed over the belief states.

11. Describe a general version of a decision-theoretic agent that solves POMDPs by employing a look-ahead approach. Give the corresponding graphical representation and describe the algorithm for estimating the best action at each time step.

a. Elements of agent include:

i. The transition and observation models are represented by a dynamic Bayesian network.

ii. The dynamic Bayesian network is extended with decision and utility nodes, as used in decision networks. The resulting model is called a dynamic decision network or DDN.

iii. Filtering is used to incorporate each new percept and action and to update the belief state representation.

iv. Decisions are made by projecting forward possible action sequences and choosing the best one.

b.

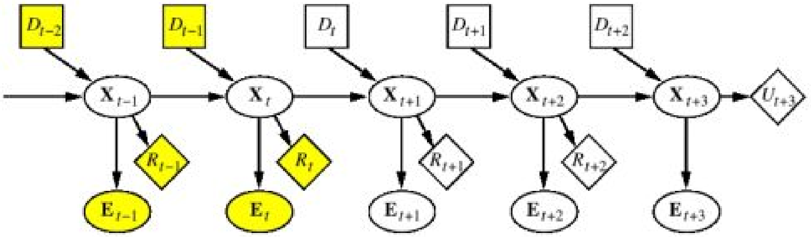


Figure 3: A generic structure of a dynamic decision network.

c. Decisions are made by:

i. Projecting forward possible action sequences

ii. Estimating their effects

iii. Choosing the best one

12. You might be provided a small POMDP like the tiger problem and asked to compute an appropriate policy.

a.

9 Inductive Learning: Decision Trees

Data points are vectors - we want the simplest decision trees by priorities attributes at top

- Will get questions like homework

1. **What is the input and output of inductive learning?**
    a. Input: The correct value of an unknown function for particular inputs
    b. Output: Recover the unknown function or approximate it.
2. **What is a hypothesis in the context of machine learning and when is it a desirable one?**
    a. Hypothesis is a function that approximates the desired unknown function.
        i. A consistent hypothesis is one that agrees with all the available data points/examples.
        ii. A good consistent hypothesis: Generalizes well.
3. **How is a decision tree built from examples? What are the steps of the algorithm?**
    a. The decision tree approach constructs a tree data structure. The internal nodes correspond to attributes. The edges correspond to possible values that the parent attributes can acquire. Once a tree is constructed we can use it to correctly classify a new example according to its attribute values by performing a sequence of tests at each node of the tree. The final decision of the tree is stored at the leaf nodes.
4. **What is the measure of an attribute's utility in the context of inductive learning? Provide the mathematical expression and derivation.**
    a. An attribute's utility is related to the amount of information it provides.
    b. u_i = possible answer, $P(u_i)$ = prob. of possible answer

$$I(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) \cdot log_2 P \cdot (v_i).$$

5. **You might be given a problem like the "wait for a table" example from the lectures and asked to compute the decision tree for the problem. You will have to make use of the information gain approach.**

    a.

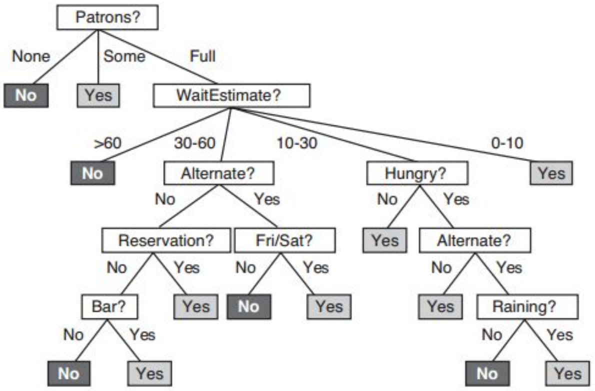


Figure 2: A decision tree for deciding whether to wait for a table.

6. **What is the problem of overfitting?**

    a. The goal is to design a learning procedure that is able to generalize and avoid overfitting which occurs when the learning algorithm is trained to detect meaningless "regularity" in the data.

7. **What is the basic methodology to train a learning algorithm given examples so as to achieve good performance? How should the examples be used and how can you measure the performance of a learning algorithm?**

    a. The steps of the methodology to assess the performance of a learning algorithm are the following:

        i. Collect a large set of examples.

        ii. Divide it into 2 distinct sets: Training set and Test set.

        iii. Apply the learning algorithm to the training set and generate hypothesis h.

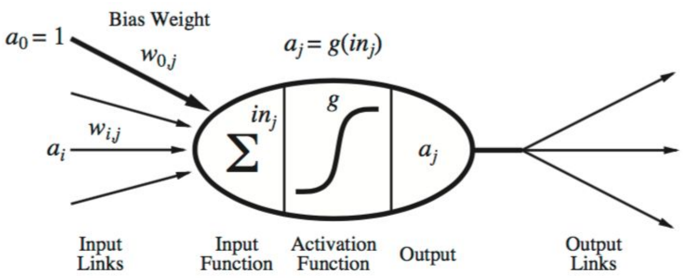
        iv. Measure the percentage of examples in the test set correctly classified by h.

        v. Repeat steps 2→4 for different sizes of training vs. test set, and randomly select examples for training and test sets.

10 Artificial Neural Networks

1. **Give a graphical representation of a single artificial neuron.**

   a.



   b. nothing more than a weighted summation

   c. Write the output of the neuron as a function of the input.

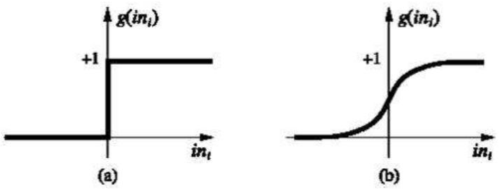
      i.

$$a_i = g(\sum_{j=0}^{n} W_{j,i}a_j)$$

      ii. Output = activation function(sum(wj*aj))

2. **Why is the sigmoid function preferable as an activation function over the step function?**

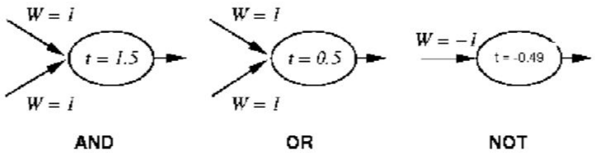
   a. Sigmoid function is differentiable which comes in handy for machine learning. Differentiable means that you can find the slope - so you can see the change.



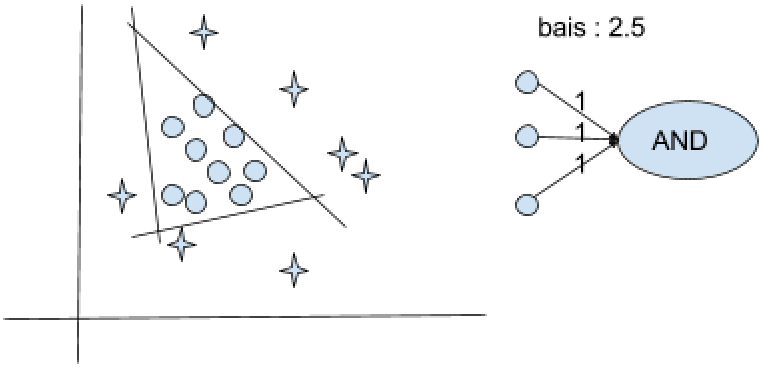
   b. - Figure 3: (a) The threshold activation function, which outputs 1 when the weighted sum of inputs is positive, and 0 otherwise. (b) The sigmoid function.

   c. You can take the derivatives to compare the error

3. **You may be provided simple Boolean functions and asked to define the weights of the neuron that represents these functions.**

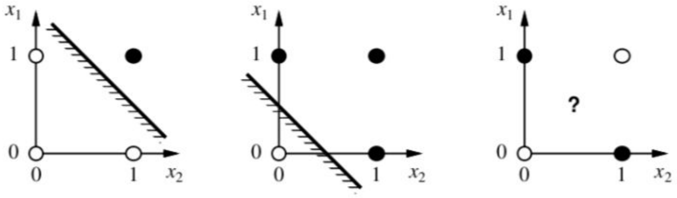


   a. Figure 4: Units with a threshold function acting as logic gates, given appropriate input weights and bias weights



   b.

4. **You may be asked to give examples of linearly separable data points and points that are not linearly separable.**



(a) x_1 **and** x_2 (b) x_1 **or** x_2 (c) x_1 **xor** x_2

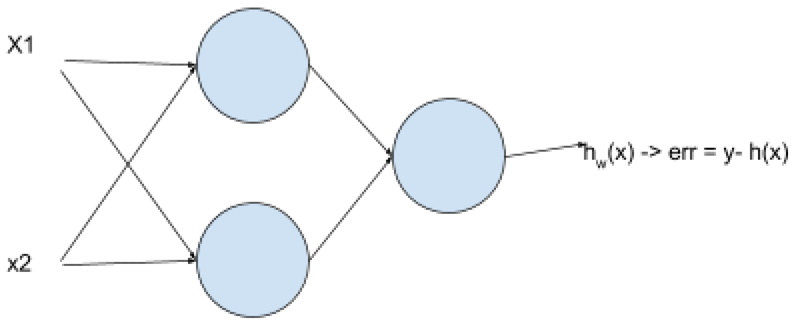
   a. Figure 6: Illustration of linear separability in threshold perceptrons.

   b. For the first case you may be asked to return the weights of a perceptron that separates the data to the correct classes.

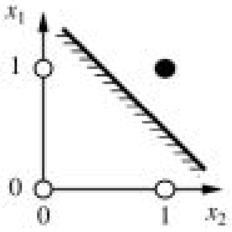
      i. Y -> -> err = y - h_w (x)

      ii. If you have 2 (or more) linear classifiers then err is more difficult to calculate.

      iii. Take the error from parent node and distribute back depending on the weight

iv. 

5. **Give a graphical representation of a perceptron and of a multi-layer feed-forward neural network. What are the properties (advantages/disadvantages) of each scheme?**
    a. Perceptron (single-layer feed-forward)



(a) x_1 and x_2

- i. Advantages
    1. each output unit is independent of the others
    2. inputs connected directly to the outputs
    3. there is a simple learning algorithm that will fit a threshold perceptron to any linearly separable training set
    4. we can derive a similar algorithm for learning in sigmoid perceptrons
- ii. Disadvantage
    1. many Boolean functions that the threshold perceptron can not represent
    2. can only express a limited number of functions

b. Multi Layer Feed-Forward

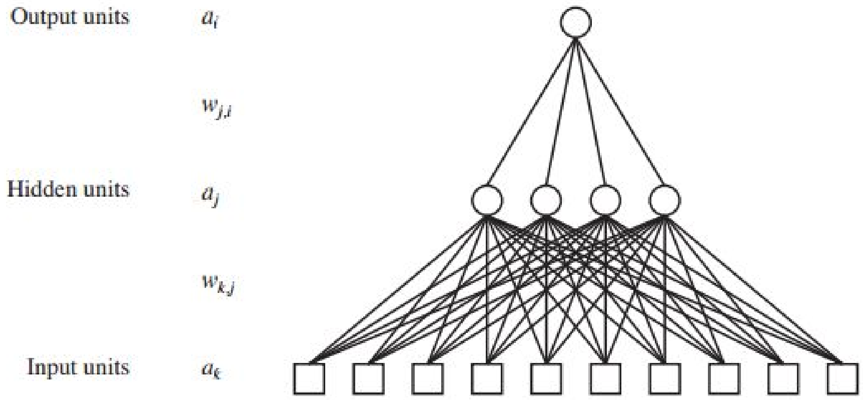


Figure 8: A multi-layer neural network with 10 inputs, 4 hidden units, and 1 output

i. Advantages

   1. hidden layers will increase the size of the hypothesis space that the network can represent.

ii. Disadvantages

   1. the error at the hidden layers is confusing, must use backpropagation.

6. **How can we learn the weights of a perceptron given examples so as to best approximate the underlying process that produced these examples? Describe the basic idea, the algorithm and derive the related mathematical expression.**

   a. The idea behind this algorithm is to adjust the weights of the network to minimize the error on the training set.

   b. Algorithm:

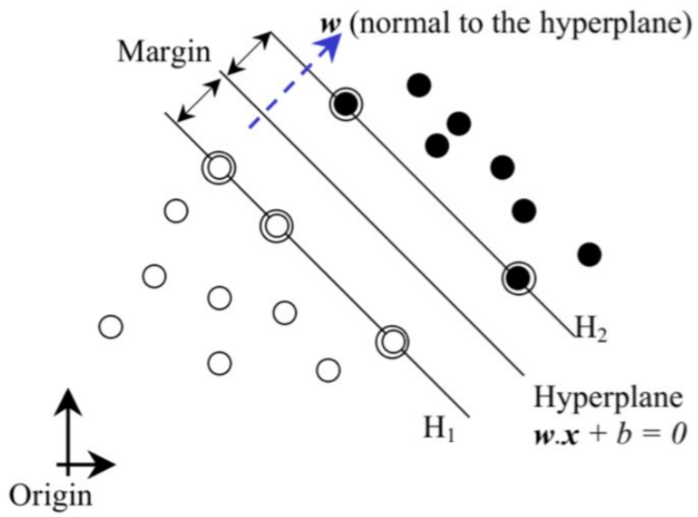
$$W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j$$

   c. Math Expression: y is expected output, $h_w(x)$ is actual

$$E = \frac{1}{2}Err^2 \equiv \frac{1}{2}(y - h_w(x))^2$$

7. **What is the complication that arises in a multi-layer feed-forward network when trying to apply back-propagation so as to learn the weights from examples? How can we overcome these challenges? Describe the underlying idea.**
    a. To update the connections between the inputs and the hidden units, we will need to define a quantity that is analogous to the error term for the output nodes. The idea is that the hidden node is responsible for some fraction of the error in each of the output nodes that it is connected to. The 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.

11 Support Vector Machines

1. **What is the idea of a "margin" in the context of support vector machines and how does it relate to a separating hyperplane?**
    a. The margin is the distance from the hyperplane to the the point on either side of the hyperplane.
    b. Provide a graphical example.
        i. 
Margin
$\boldsymbol{w}$ (normal to the hyperplane)
H_2
Hyperplane
$\boldsymbol{w}.\boldsymbol{x} + b = 0$
H_1
Origin
    c. What are the support vector in the same context?
        i. The training points, which lie on one of the hyperplanes (H1, H2) and whose removal would change the solution found are called support vectors
2. **What kind of problem do Support Vector machines have to solve in order to compute the separating hyperplane and the corresponding margin? Constraint Optimization.**
    a. What are the desirable properties of the resulting problem?
        i. (optimization)To maximize the width of the margin.
        ii. (constraint) Correctly classify the data points

    Solution to this problem depends only on dot products

3. **What is the idea in Support Vector Machines for addressing non-linear classification challenges?**
    a. What fundamental theorem do SVMs take advantage of?
        i. N data points are always linearly separable in an N-1 dimensional space
        ii. Idea: to map the points in a high enough dimension to separate them

    xi->F(xi), xj->F(xj), original low dimensional data points to high dimensional. Instead of explicitly doing the mapping, we prefer kernel function.

4. **What is the role of a kernel function in the context of a Support Vector Machine?**
        i. The role of kernel functions are to map the points in the data set to a higher dimensional space where it is possible to separate them with a linear hyperplane.

    b. What are some popular non-linear kernel functions?

    Polynomial: K(xi, xj)=F(xi)*F(xj)=>(1+xi*xj)^d

    Gaussian Radial Basis Function: K(u,v)=exp($\frac{-\|u-v\|^2}{2\sigma^2}$)

    Sigmoidal: (hyperbolic separating surface): K(u,v)=tanh(ku*v- δ)

5. **You may be provided a small non-linearly separable classification example and asked to illustrate its solution by SVMs.** NO. yes :(No…...I HOPE SO ->> FUCK ME IN THE ASS PLEASE!@!!! No thanks ->> are you sure??? ;) ooooooohhhohoooo alright fine when are you free →>STOP WATCHING ME no