Lecture 5 Propositional Logic

1. **Inferencing Methods**

   $$\frac{\alpha \to \beta, \alpha}{\beta} \qquad \frac{\alpha \wedge \beta}{\alpha}$$

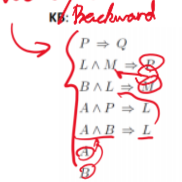
   1.1 Model Checking
   Exhaustive truth table enumeration (brute-force)

   1.2 Proof by Deduction
   a. Modus Ponens (top right corner 1)
   b. AND-Elimination rules (trc2)
   c. Inferencing rules (right)
   d. **Forward/Backward Chaining**

   $$\frac{\alpha \leftrightarrow \beta}{(\alpha \to \beta) \wedge (\beta \to \alpha)} \qquad \frac{(\alpha \to \beta) \wedge (\beta \to \alpha)}{\alpha \leftrightarrow \beta}$$

   1.3 Proof by Contradiction
   Resolution Theorem
   Converts inferencing problem into SAT problem with clauses as constraints.
   Relies on two tools: Refutation and Resolution

   Forward/Chaining

   a. **Resolution**

   Given either X or Y are true.
   If we know not X is true, then we can resolve Y as true.
   Formally

   Query: Q

   $$\{X \vee Y, \neg X\} \models Y$$

   or

   $$\{(X \vee Y) \wedge (\neg X)\} \models Y$$

   b. **Refutation** (Proof by contradiction)

   > $KB \models \alpha$ if and only if $KB \wedge \neg\alpha$ is a contradiction
   > or
   > $KB \models \alpha$ if and only if $KB \wedge \neg\alpha \models false$

2. **Grammar for CNF**
   Disjunction of (conjunction of laterals)
3. **Propositional Definite Clauses**
   Must contain **exactly 1 positive literal**
   Definite clauses can be written as implications because

   $$X \to Y \equiv (\neg X \vee Y)$$

   Example:

   $$(\neg X \vee \neg Y \vee Z) \equiv X \wedge Y \to Z$$

   - Examples:
     - $(X \vee Y)$ is not a definite clause
       - Two positive literals
     - $(\neg X \vee Y)$ is a definite clause
       - One positive literal
     - $(\neg X \vee \neg Y)$ is not a definite clause
       - No positive literal
     - X is a definite clause
       - One positive literal
4. **Horn Clauses:** Allows at most 1 positive literal
   - Closed under resolution
     - Resolvent of two Horn clauses is a Horn clause
   - Example:
     - $((\neg X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee \neg Y \vee Z)) \models (\neg X \vee \neg Y)$
   - Fact
     - Horn clause with single positive lite
     - Ex: X
       - $X \equiv (True \Rightarrow X)$
   - Goal Clause
     - Horn clause with no positive literal
     - Ex: $(\neg X \vee \neg Y)$
       - $(\neg X \vee \neg Y) \equiv (X \wedge Y \Rightarrow False)$
5. **Fact and Goal Clauses (right)**
6. **Forward & Backward Chaining**

   6.1 Backward Chaining
   a. **Start with query alpha**
   b. Find the implications whose **conclusion is alpha**
   c. Recursively prove the **antecedents** of each implication
   d. If at **least one of the antecedents can not be proved to be true,** return False

   6.2 Which method to choose?
   (a) Forward Chaining: Known facts are available. Goal-independent reasoning. Exhaustively derive conclusions.
   (b) Backward Chaining: Specific goal or query is given. Goal-oriented reasoning. Efficiency gained by pruning the search space.
   (c) If the KB contains NON-definite clauses (like A ⇔ B V E), we can't use forward/backward chaining. We should use resolution-refutation instead.

Lecture 6 Probabilistic Reasoning

1. **Independent Events**

   If variables X and Y are independent, i.e. if X perpendicular to Y

   $$P(X|Y) = P(X)$$
   $$P(Y|X) = P(Y)$$
   $$P(X,Y) = P(X)P(Y)$$

   Math Preliminary: Bayes' Rule

   $P(B|A) = \dfrac{P(A|B) \cdot P(B)}{P(A)}$

   If X and Y are conditionally independent given a variable Z then

   $P(A,B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$.

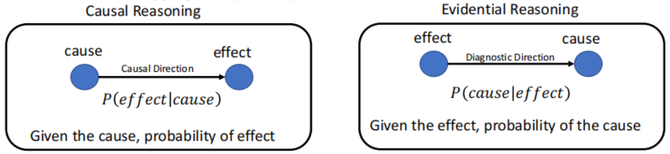
   ② Law of total probability:

   $P(A) = \sum P(A|B_n) \cdot P(B_n)$

   $$P(X|Y,Z) = P(X|Z)$$
   $$P(Y|X,Z) = P(Y|Z)$$
   $$P(X,Y|Z) = P(X|Z)P(Y|Z)$$

2. **Evidential Reasoning and Causal Reasoning**

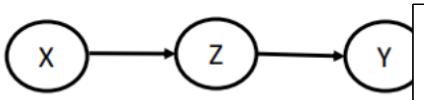
   Causal Direction: $P(symptoms|disease)$ — Causal Reasoning
   Diagnostic Direction: $P(disease|symptoms)$ — Evidential Reasoning

   Causal knowledge
   Diagnostic knowledge

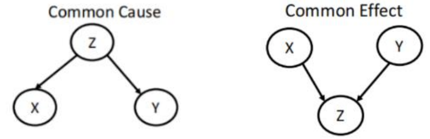
3. **Components of Bayesian Networks**

   3.1 Casual Chain

   $$P(X,Y,Z) = P(X)P(Z|X)P(Y|Z)$$

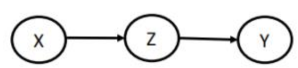
   3.2 Common Cause and Common Effe

   $P(X,Y,Z) = P(Z)P(X|Z)P(Y|Z)$ $\qquad$ $P(X,Y,Z) = P(X)P(Y)P(Z|X,Y)$

4. **Dependencies in Causal Chains**

   Causal Chain: Indirect Causal Effect

   $$P(X,Y,Z) = P(X)P(Z|X)P(Y|Z)$$

   No

5. **d-Separation**

Tips for converting natural language into propositional symbols:

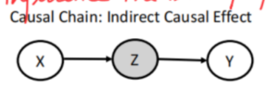
1. Unless A ,does not B = 不 A 就不 B = B 就 A = **B->A**
2. A only if B = **A->B**
3. A if B = **B -> A**
4. A if and only if B = **A ⇔ B**

for a 2x2 matrix the inverse is:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

5.1 d-Separation in Causal Chains

⇒ X doesn't influence the belief of y given z



$$P(X,Y,Z) = P(X)P(Z|X)P(Y|Z)$$

Assume evidence on Z, i.e. Z is known

$$P(Y|X,Z) = \frac{P(X,Y,Z)}{P(X,Z)} = \frac{P(X)P(Z|X)P(Y|Z)}{P(Z|X)P(X)} = P(Y|Z)$$

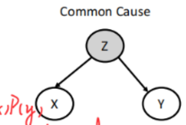
5.2 d-Separation in Common Cause

Assumption: Z is known

$$\boxed{P(X,Y|Z)} = \frac{P(X,Y,Z)}{P(Z)} = \frac{P(X|Z)P(Y|Z)P(Z)}{P(Z)} = \boxed{P(X|Z)P(Y|Z)}$$

X is independent of Y given Z
(or)
$X \perp Y \mid Z$
(or)

$P(X,Y) \neq P(X)P(Y)$
⇒ Marginally Independent

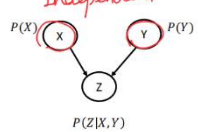


5.3 d-Separation in Common Effect

Z is unknown

$P(X,Y) = \sum_Z P(X)P(Y)P(z|X,Y)$ ⟶ Marginalize unknown (Z)
$= P(X)P(Y)\ \sum_z P(z|X,Y)$
$= P(X)P(Y)$

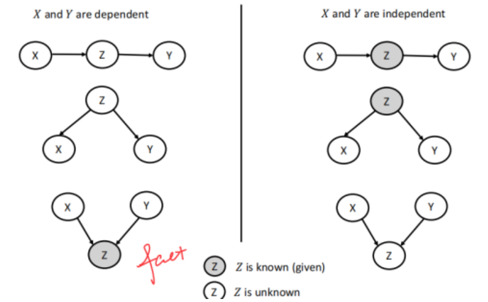
Marginally Independent



If Z is unknown, we are not sure about the events X and Y
Knowledge about one event does not reduce uncertainty about other event

If Z is known, then knowledge of X reduces the uncertainty about Y

5.4 d-Separation summary



5.5 Identifying d-separation

known facts
"Query"

- Given a set of nodes $\boldsymbol{Z}$, are the set of nodes $\boldsymbol{X}$ conditionally independent of set of nodes $\boldsymbol{Y}$?
  1. Consider the ancestral subgraph consisting of $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$
  2. Construct Moral Graph, i.e., add links between any unlinked pair of nodes that share a common child
  3. Replace directed links by undirected links
  4. If $\boldsymbol{Z}$ blocks all paths between $\boldsymbol{X}$ and $\boldsymbol{Y}$ in the resulting graph, then $\boldsymbol{Z}$ d-separates $\boldsymbol{X}$ and $\boldsymbol{Y}$

Lecture 7 Hidden Markov Models

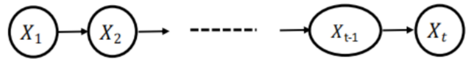
1. **Markov Property**

Irrelevant

$$P(X_t|X_{t-1},\boxed{\cancel{X_{t-2},\dots,E_{t-1},E_{t-2},\dots}}) = P(X_t|X_{t-1})$$

2. **Joint distribution**

▼ Markov Models: Joint Distribution



Joint distribution of the first n variables:

$$P(X_1, X_2, X_3, \dots, X_t) = P(X_1)\prod_{i=2}^{n} P(X_i|X_{i-1})$$

3. **Transition in Markov Chains**

3.1 Transition model representations

rs:

$high) = 1.0$

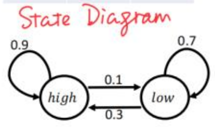
1

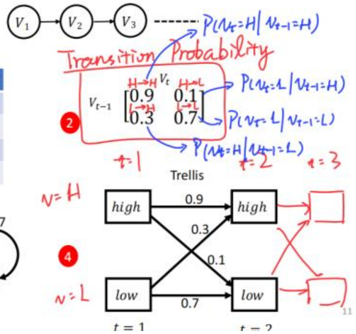
V_{t-1}	V_t	$P(V_t\|V_{t-1})$
high	high	0.9
high	low	0.1
low	high	0.3
low	low	0.7

ate
els

State Diagram

3





3.2 Transition matrix

Same previous state -> same row in matrix

$$\boldsymbol{T} = \begin{bmatrix} P(V_2 = high|V_1 = high) & P(V_2 = low|V_1 = high) \\ P(V_2 = high|V_1 = low) & P(V_2 = low|V_1 = low) \end{bmatrix}$$

3.3 Calculating the probability distribution of the next state

a. **Next state**

$$\boxed{\boldsymbol{P}(V_2) = \boldsymbol{T}^T\boldsymbol{P}(V_1)}$$

$[P(V_2 = high)]$

$$\begin{bmatrix} Id & B \\ C & D \end{bmatrix}^T = \begin{bmatrix} Id & C \\ B & \end{bmatrix}$$

b. **Next n states**

$$\boldsymbol{P}(X_{n+1}) = \boldsymbol{T}^T\boldsymbol{P}(X_n) = (\boldsymbol{T}^T)^n\boldsymbol{P}(X_1)$$

3.4 Stationary Distribution properties

Stationary distribution is independent of the initial distribution.

Properties of stationary distribution:

$$P_\infty(X) = P_{\infty+1}(X)$$
$$P(X_\infty) = T^T P(X_\infty)$$
$$P_\infty(X) = \sum_x P(X|x)P_\infty(x)$$
$$\sum_x P_\infty(X) = 1$$

In practice, we usually use these equations:

$$\begin{aligned} P(X_\infty = H) &= P(H|H)P(X_\infty = H) + P(H|L)P(X_\infty = L) \\ P(X_\infty = L) &= P(L|H)P(X_\infty = H) + P(L|L)P(X_\infty = L) \\ P(X_\infty = H) + P(X_\infty = L) &= 1 \end{aligned}$$

4. **Hidden Markov Models**

4.1 Observation property

Current observation is independent of everything else given the current state

$$P(E_t|E_1, \dots, E_{t-1}, E_{t+1} \dots, E_n, X_{0:n+1}) = P(E_t|X_t)$$

4.2 Joint probability distribution

Joint distribution   Product of transition distribution

$$P(E_1, \dots, E_n, X_0, X_1, \dots, X_n, X_{n+1}) = \underbrace{P(X_0)}_{\text{Initial Probability}} \underbrace{\prod_{i=1}^{n+1} P(X_i|X_{i-1})}_{\text{Transition Distribution}} \underbrace{\prod_{i=1}^{n} P(E_i|X_i)}_{\text{Emission Distribution}}$$

4.3 Bayesian Recursive Filtering

- **Objective:**
  - Calculation of $B(X_{t+1}) = P(X_{t+1}|\boldsymbol{e}_{1:t+1})\ \forall t$

$B(X_t) = P(X_t|e_{1:t})$

- **Solution:** Recursive Filtering

f1.4   Base case (Prior Distribution)

$P(X_4) \to P(X_3) \to P(X_2) \to P(X_1)$

$$\boldsymbol{f}_{1:t+1} = \alpha\, \boldsymbol{O}\, \boldsymbol{T}^T \boldsymbol{f}_{1:t}$$

α : Normalization constant

$$\boldsymbol{f}_{1:t+1} = \begin{bmatrix} B(X_{t+1} = 1) \\ \vdots \\ B(X_{t+1} = |S|) \end{bmatrix}$$ Probability distribution of states at time $t+1$

$$\boldsymbol{f}_{1:t} = \begin{bmatrix} B(X_t = 1) \\ \vdots \\ B(X_t = |S|) \end{bmatrix}$$ Probability distribution of states at time t

Transition Matrix

$$T = \begin{array}{c|ccccc} & S & Id & B & C & D \\ \hline O & 1 & 0 & 0 & 0 & 0 \\ S & 0.3 & 0.2 & 0.1 & 0.1 & 0.3 \\ Id & 0.25 & 0.1 & 0.05 & 0.15 & 0.25 \\ B & 0.05 & 0.4 & 0.15 & 0.2 & 0.2 \\ C & 0.5 & 0.2 & 0.2 & 0.05 & 0.05 \end{array}$$

actually

$$\boldsymbol{O} = \boldsymbol{diag}([\mathbf{P}(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1} = 1) \ \dots \ \mathbf{P}(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1} = |S|)])$$ Observation matrix

$$\boldsymbol{T} = \begin{bmatrix} T_{11} & \dots & T_{1|S|} \\ \vdots & \ddots & \vdots \\ T_{|S|1} & \dots & T_{|S||S|} \end{bmatrix}, T_{ij} = \mathbf{P}(X_{t+1} = j|X_t = i)$$ Transition Matrix

33

Hint: for P(X1|E1) try converting to joint p. First, then break it down

Lecture 8 Markov Decision Process

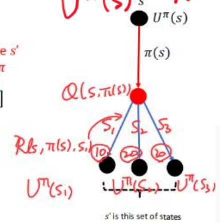
1. **Bellman's Update Equation for Utility**

- Expected utility of state s with a policy π

If deterministic next state:

$U^\pi(s) = R(s) + \gamma U(s')$

$$\underbrace{U^\pi(s)}_{\text{Utility of state } s \text{ with policy } \pi} = \sum_{s'} \underbrace{P(s'|\pi(s), s)}_{\substack{\text{Probability of moving to state } s' \text{ with} \\ \text{action } \pi(s) \text{ at state } s}}[\underbrace{R(s, \pi(s), s')}_{\substack{\text{Reward for taking an action } \pi(s) \\ \text{at state } s \text{ and moving to state } s'}} + \gamma \underbrace{U^\pi(s')}_{\text{Utility of state } s' \text{ with policy } \pi}]$$



$P_1(1+10) + P_2(2+20) + P_3(3+30)$

Utility of a state is expressed in terms of utility of neighbors

Q(s,a): expected value of action a at state s

$$Q(s, a) = \sum_{s'} P(s'|a, s)[R(s, a, s') + \gamma\, U(s')]$$

$$U(s) = \max_{a \in A(s)} Q(s, a)$$

$\pi_1 \to U_1 \to \pi_2 \to U_2 \dots$

2. **Policy Iteration**

a) Start with a **random policy** pi_i

b) Policy evaluation: **calculate utility values** for the policy pi_i

c) Policy improvement: Create **new policy pi_{i+1}** based on utility values

d) Repeat the above steps **until no change in policy**

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

3. **Value Iteration**

▼ Value Iteration (VI)

- Start with U_i(s) = 0 for all s
- Calculate Q values and update Utility values
- Once utility values converge to optimal values, select the corresponding policy

Lecture 9&10 Introduction to Learning, PLA

1. **Linear Regression**

X usually needs to be augmented with a column of 1 at left!

1.1 Choosing cost functions

Outliers are due to human errors – MAE

Outlier detection is critical - MSE

1.2 What can be done to the value of the learning rate to enable better convergence: Start from **large steps** to reach the optimal value faster, as optimal value is reached, **reduce step size** to allow for more gradual convergence

1.3 Gradient Descent

$$\boldsymbol{w} = \boldsymbol{w} - \alpha\, \nabla J(\boldsymbol{w})$$

vector

learning rate

scalar

Objective: Find w s.t. $J(w)$ is minimized

where,

$\frac{d}{dw} J(w)$

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$\nabla J(w) = \begin{bmatrix} \frac{\partial J(w)}{w_0} \\ \frac{\partial J(w)}{w_1} \end{bmatrix}$$

vector

2. **Logistic Regression (binary classification)**

2.1 One hypothesis for each class

$$z_1^{(i)} = \boldsymbol{w}_{cat} \cdot \boldsymbol{x}^{(i)}$$ and so on so forth

2.2 Output of the classifier

$$Q = \begin{bmatrix} P(\hat{y}^{(i)} = 1) \\ P(\hat{y}^{(i)} = 2) \\ P(\hat{y}^{(i)} = 3) \end{bmatrix} = \begin{bmatrix} \hat{y}_1^{(i)} \\ \hat{y}_2^{(i)} \\ \hat{y}_3^{(i)} \end{bmatrix} = Softmax\left(\begin{bmatrix} z_1^{(i)} \\ z_2^{(i)} \\ z_3^{(i)} \end{bmatrix}\right) = \begin{bmatrix} softmax(z_1^{(i)}) \\ softmax(z_2^{(i)}) \\ softmax(z_3^{(i)}) \end{bmatrix}$$

$$softmax\left(z_j^{(i)}\right) = \frac{e^{z_j^{(i)}}}{\sum_j e^{z_j^{(i)}}}$$

$Z = W \cdot X^T =$ w_1 w_2 w_3 3×3 3×3

3. **Perceptron Learning Algorithm**

Difference between PLA & Gradient Descent? ① Can put the learning rate ② Derivative-free

Perceptron Learning Algorithm (PLA)

Frank Rosenblatt (1943)

1. Initialize weights w_i
   - Could be all zero, or random small values
2. For each instance i with features $x^{(i)}$
   - Classify $\hat{y}^{(i)} = sgn(w^T x^{(i)})$ +1 −1
3. Select one misclassified instance
   - Update weights: $w \leftarrow w + \Delta w$ — How do we update w?
4. Iterate steps 2 to 3 until
   - Convergence (classification error < threshold), or
   - Maximum number of iterations

$$\underbrace{w}_{\text{new weight}} \leftarrow \underbrace{w}_{\text{old weight}} + \underbrace{\eta}_{\text{learning rate}}\underbrace{(y - \hat{y})}_{\text{learning error}}x$$

Expected output   Actual output