

Probabilistic Graphical Models

Huynh Xuan Phung - Coursera

Contents

1	Representation	5
1.1	Introduction	5
1.1.1	Distribution:Marginalization,Conditioning	6
1.1.2	Factors	6
1.2	Bayesian Network (Directed Models)	6
1.2.1	Semantics and Factorization	6
1.2.2	Reasoning Pattern	7
1.2.3	Flow of Probabilistic Influence: active trail	7
1.3	Bayesian Networks: In-dependencies	8
1.3.1	Conditional Independence	8
1.3.2	Independence in Bayesian Networks: d-separate, I-map	9
1.3.3	Naive Bayes Model	9
1.3.4	Assignment	10
1.4	Bayesian Networks: Knowledge Engineering	10
1.4.1	Medical Diagnosis	10
1.4.2	Knowledge Engineering: Example SAMIAM	10
1.4.3	Programming Assignment 1	10
1.5	Template Models	10
1.6	Temporal Models - DBNs	10
1.7	Temporal Models - HMMs	12
1.8	Plate Models	12
1.8.1	Modeling Repetition	13
1.9	Plate Dependency Model	13
1.10	Structured CPDs	14
1.11	Deterministic CPDs	14
1.12	Tree- structured CPDs	14
1.13	Independence of Causal Influence	15
1.13.1	Independence of Causal Influence	15
1.13.2	Sigmoid CPD	15
1.14	Continue Variables	15
1.15	Exercise: Tree CPD - context specific In-dependencies	15
1.16	Markov Networks: Undirected Models	16
1.16.1	Pairwise Markov Networks	16
1.16.2	General Gibbs Distribution	16

1.16.3	Induced Markov Network	17
1.16.4	Factorization	17
1.16.5	Flow of Influence	17
1.16.6	Active Trails	17
1.16.7	Task-specific prediction	17
1.16.8	Corelated Features	17
1.16.9	CRFs and Logistic Model	17
1.16.10	Independences in Markov Networks	18
1.16.11	Capturing Independence in P	18
1.16.12	Minimal I-map	18
1.16.13	Perfect Map	18
1.16.14	Uniqueness of Perfect Map	18
1.16.15	I-equivalence	18
1.17	Local Structure in Markov Networks	18
1.17.1	Log-Linear Representation	18
1.17.2	Ising Model	19
1.17.3	Metric MRFs	19
1.17.4	Shared Features in Log-Linear Models	19
1.17.5	Clique	19
2	Decision Making	21
2.1	Simple Decision Making	21
2.1.1	Simple Influence Diagram	21
2.1.2	Information Edges	22
2.1.3	Expected Utility with Information	22
2.1.4	Finding MEU Decision Rules	22
2.1.5	More Generally	22

Chapter 1

Representation

1.1 Introduction

Model

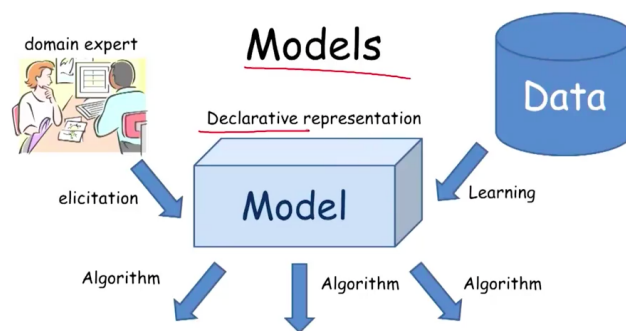


Figure 1.1: Model is a declarative representation of our understanding of the world

It is important because the same representation, that same model can be used in the the context of one algorithms that might answer different kind of questions. Or the same question in more efficient way.

We can construct methodologies the elicit these models from a human expert, or learn from data or combination.

Uncertainty

- Partial knowledges of state of the world
- Noisy observations
- Phenomena not covered by our model
- Inherent stochasticity

Probability Theory

- Declarative representation with clear semantics
- Powerful reasoning patterns: conditioning decision making

- Established learning methods
- Complex Systems
- Graphical Models: Bayesian Networks, Markov Networks (directed or undirected graphs)
- Graphical Representation
 - Intuitive and compact data structure
 - Efficient reasoning using general purpose algorithms
 - Sparse parameterization
 - feasible elicitation : by hand
 - learning from data automatically

1.1.1 Distribution:Marginalization,Conditioning

Joint Distribution: $P(I,D,G)$

Conditioning: observation 1 value of variable \rightarrow Reduction \rightarrow Renormalization $P(I,D,g^1) \rightarrow P(I,D|g^1)$

Marginalization: $\sum_I P(I,D) = P(D)$. Example: you have thrown two 6-sided dice, D_1 and D_2 . $P(D_1,D_2)$ is a joint probability distribution. The probability that $D_2 = 1$ is equals to $\sum_{i=1}^6 P(D_1 = i, D_2 = 1)$.

1.1.2 Factors

A factor is a function or table $\phi(X_1, \dots, X_k)$

$\phi : Val(X_1, \dots, X_k) \rightarrow R$

Scope = X_1, \dots, X_k

Joint distribution is a factor

Unnormalized measure is a factor

Conditional Probability Distribution (CPD) is a factor $P(G|I,D)$: G in columns while I,D in rows.

Why factors?

- Fundamental building block for defining distributions in high-dimensional spaces
- Set of basic operations for manipulating these probability distributions

1.2 Bayesian Network (Directed Models)

1.2.1 Semantics and Factorization

What does random variable depend on?

Draw nodes, edges, each node with a factor is CPD (conditional probability distribution)

A bayesian network is:

- A directed acyclic graph (DAG) G whose nodes represent the random variables
- For each node X_i a CPD $P(X_i|Par_G(X_i))$

The BN represent a joint distribution via the chain rule for Bayesian Networks

$$P(X_1, \dots, X_n) = \prod_i P(X_i | \text{Par}_G(X_i))$$

BN is a Legal distribution : $\sum P = 1$ and $P > 0$

1.2.2 Reasoning Pattern

Causal Reasoning: reasoning going down

Evidential Reasoning: reasoning going up

Inter-causal Reasoning: The probability of class is hard, if we observe the "C" grade and change the posterior probability of high intelligence is goes up

1.2.3 Flow of Probabilistic Influence: active trail

When can X influence Y?

$$X \rightarrow Y$$

$$X \leftarrow Y$$

Active Trails:

A trail $X_1 - \dots - X_k$ is active if: it has no v-structures $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$

When can X influence Y given evidence about Z?

A trail $X_1 - \dots - X_k$ is active given Z if:

— for any v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$ we have that X_i or one of its descendants $\in Z$

— no other X_i is in Z

Assignment

1. How many parameter to present a CPD. If X have m possibilities the $P(X)$ needs m-1 independent parameters

If Y have k possibilities, Z have l possibilities then $P(X \rightarrow Y, Z)$ has (m-1)*k*l independent parameters.

2. Inter-causal reasoning

To calculate the required values, we can apply Bayes' rule. For instance,

$$\begin{aligned} \frac{P(A=1|T=1, P=1)P(A=1, T=1, P=1)}{P(T=1, P=1)} \\ = \frac{P(A=1, T=1, P=1)}{(P(A=0, T=1, P=1) + P(A=1, T=1, P=1))} \end{aligned}$$

We can then use the chain rule of Bayesian networks to substitute the correct values in, e.g.,

$$P(A = 1, T = 1, P = 1) = P(P = 1) * P(A = 1) * P(T = 1 | P = 1, A = 1)$$

This example of inter-causal reasoning meshes well with common sense: if we see a traffic jam, the probability that there was a car accident is relatively high. However, if we also see that the president is visiting town, we can reason that the president's visit is the cause of the traffic jam; the probability that there was a car accident therefore drops correspondingly.

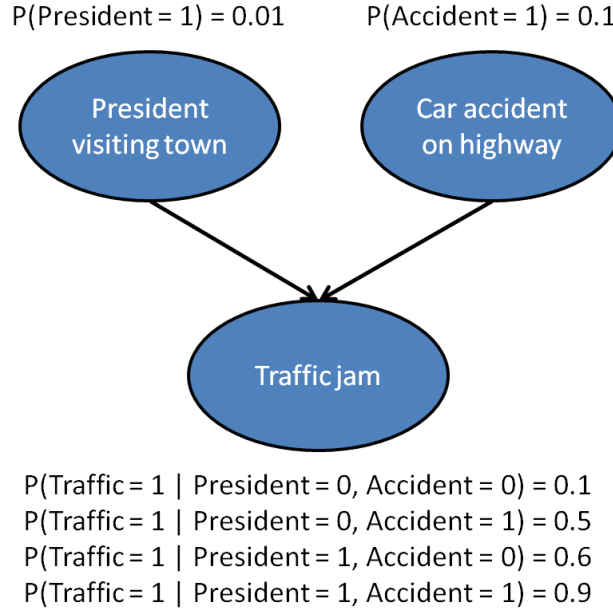


Figure 1.2:

1.3 Bayesian Networks: In-dependencies

1.3.1 Conditional Independence

Independence

For events α, β , P satisfy independence if:

- $P(\alpha, \beta) = P(\alpha) * P(\beta)$
- $P(\alpha|\beta) = P(\alpha)$
- $P(\beta|\alpha) = P(\beta)$

Conditional Independence

For random variables X, Y, Z ; P satisfy (X independence $Y \mid Z$) if

- $P(X, Y|Z) = P(X|Z) * P(Y|Z)$
- $P(X|Y, Z) = P(X|Z)$
- $P(Y|X, Z) = P(Y|Z)$
- $P(X, Y, Z) \propto \phi_1(X, Z) * \phi_2(Y, Z)$

We need to check for all cases

S and G are dependent but conditionally independent given I . I and D are independent but conditionally dependent given G , which active the V-structure here

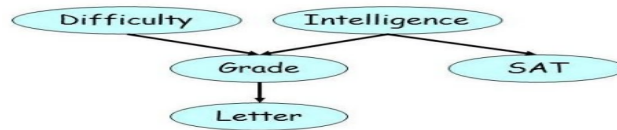


Figure 1.3: Example of student: independence

1.3.2 Independence in Bayesian Networks: d-separate, I-map

Flow of influence and d-separation

X and Y are d-separated in G given Z if there is no active trail in G between X and Y given Z. $d-sep_G(X, Y|Z)$

Factorization to Independences: BNs

Theorem If P factorizes over G, and $d-sep_G(X, Y|Z)$ then P satisfies $(X \perp Y|Z)$.

$P(S) = \sum_I P(I)P(S|I)$ (standard marginalization operation) because $P(S, I) = P(I) * P(S|I)$

If P factorizes over G, then in P, any variable is independent of its non-descendants given its parents

I-maps

d-separation in G \rightarrow P satisfies corresponding independence statement

$I(G) = (X \perp Y|Z) : d-sep_G(X, Y|Z)$

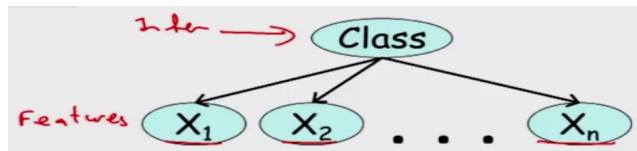
Definition: If P satisfies $I(G)$, we say that G is an I-map (in-dependency map) of P

Theorem: If P factorizes over G, then G is an I-map for P

Theorem: If G is an I-map for P, then P factorizes over G

1.3.3 Naive Bayes Model

Assumption: Given the class variable, each observed variable is independent of the other observed variables.

Figure 1.4: Naive Bayes: $(X_i \perp X_j|C)$ for all X_i, X_j

Naive Bayes Classifier

Bernoulli Naive Bayes: each observed variable is binary: 1 if appear and 0 otherwise

Multinomial Naive Bayes for Text: values of each random variable is the actual word in each document

1.3.4 Assignment

If there is no active trails between A and D, then they are independent

I-maps: if G is an I-map of P, all in-dependencies in G also in P. However, this does not mean that all in-dependencies in P also in G

I-maps can also be defined directly on graphs as follow: Let $I(G)$ be the set of in-dependencies encoded by a graph G. Then G_1 is an I-map for G_2 if $I(G_1) \subseteq I(G_2)$

I-map is not a function

1.4 Bayesian Networks: Knowledge Engineering

1.4.1 Medical Diagnosis

1.4.2 Knowledge Engineering: Example SAMIAM

1.4.3 Programming Assignment 1

Compute $P(A, B | C = C_i)$ means that observes the specific value of C and $P(C_i) = 1$ otherwise is 0

- Compute $P(A, B, C)$
- if $C_j \neq C_i$ then $P(A, B, C_j) = 0$
- $M = \sum_k P(A, B, C_k)$
- Normalize M

1.5 Template Models

Same model that can resolve multiple problems

Sharing between models + within models \rightarrow reuse parameters

Template variable $X(U_1, \dots, U_k)$ is instantiated (duplicated) multiple times

Template Models:

— Languages that specify how variables inherit dependency model from template

— Dynamic Bayesian Networks: temporal

— Object-relational models: Directed and Undirected models

1.6 Temporal Models - DBNs

Distributions over Trajectories

- Pick time granularity Δ

- $X^{(t)}$ - variable X at time $t \Delta$
- $X^{t:t'} = X^{(t)}, \dots, X^{(t')}(t \leq t')$
- Want to represent $P(X^{(t:t')})$ for any t, t'

Markov Assumption: time flows forward

$$P(X^{(0:T)}) = P(X^{(0)}) * \prod_{t=0}^{T-1} P(X^{(t+1)} | X^{(0:t)})$$

Assumption: $(X^{(t+1)} \perp X^{(0:t-1)} | X^{(t)})$: next step independent to the past if observed current state

Go back chain rules: $P(X^{(0:T)}) = P(X^{(0)}) * \prod_{t=0}^{T-1} P(X^{(t+1)} | X^{(t)})$

Time Invariance

- Template probability model $P(X' | X)$
- For all t :
- $P(X^{(t+1)} | X^{(t)}) = P(X' | X)$: for example: traffic time of day, week

Template Transition Model

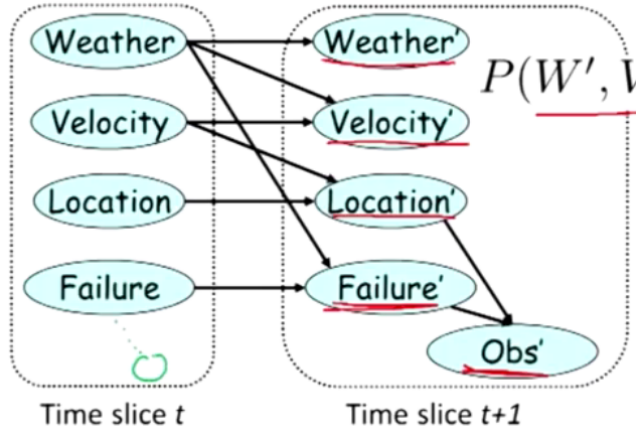


Figure 1.5: Template Transition Model

Based on network fragment, we need to compute the condition

$$P = P(W', V', L', F', O' | W, V, L, F)$$

We have CPD: $P = P(W' | W) * P(V' | V, W) * P(L' | L, V) * P(F' | F, W) * P(O' | F', L')$

We have dependence within (intra-time slice) and between (inter time slice).

Initial State Distribution: use chain rule

Ground Bayesian Network: copy the present of time 0 to time 1

2 time-slice Bayesian Network

— A transition model (2TBN) over X_1, \dots, X_n is specified as as BN fragment such that:

- The nodes includes X'_1, \dots, X'_n and a subset X_1, \dots, X_n
- Only the nodes X'_1, \dots, X'_n have parents and a CPD
- The 2 TBN defines a conditional distribution
- $P(X' | X) = \prod_{i=1}^n P(X'_i | Pa_{X'_i})$

Dynamic Bayesian Network

- A dynamic Bayesian network (DBN) over X_1, \dots, X_n is defined by a
 - 2TBN BN_{\rightarrow} over X_1, \dots, X_n
 - a Bayesian network $BN^{(0)}$ over $X_1^{(0)}, \dots, X_n^{(0)}$

Ground Network

- For a trajectory over $0, \dots, T$, ground (unrolled network)
 - The dependency model for $X_1^{(0)}, \dots, X_n^{(0)}$ is copied from $BN^{(0)}$
 - The dependency model for $X_1^{(t)}, \dots, X_n^{(t)}$ for all $t > 0$ is copied from BN_{\rightarrow}

1.7 Temporal Models - HMMs

Hidden Markov Models

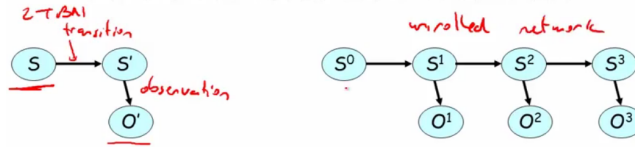


Figure 1.6: Unrolled network

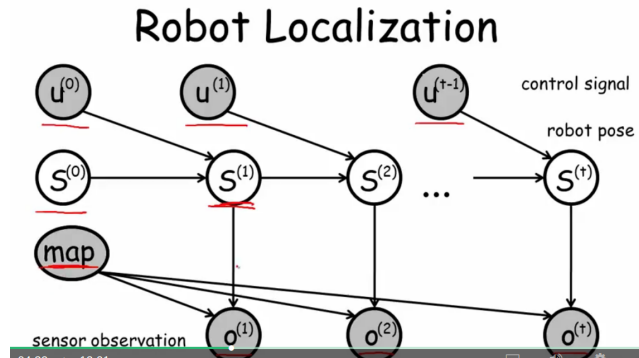


Figure 1.7:

HMM: self-transition (different with Bayesian Network): subclass of DBNs

1.8 Plate Models

Objects of the same type

1.8.1 Modeling Repetition

Box is put around Outcome variable: plate

Parameters outside of the plate (CPD)

For example: model of university with multiple students

Nested Plates

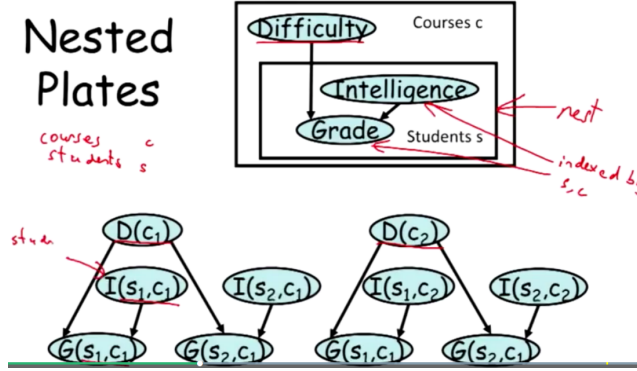


Figure 1.8: Nested Plates

Overlapping Plates

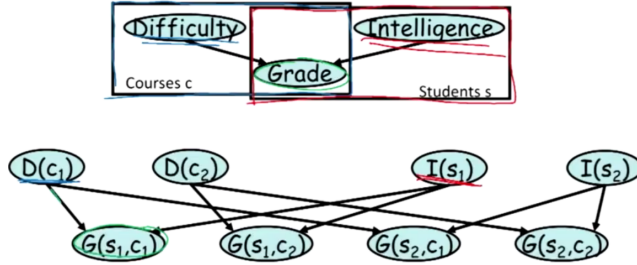


Figure 1.9: Overlapping Model

1.9 Plate Dependency Model

For a template variable $A(U_1, \dots, U_k)$:

— Template parents $B_i(U_1), \dots, B_m(U_m)$

— CPD $P(A|B_1, \dots, B_m)$

Ground Network

Plate Dependency Model

Template for an infinite set of BNs, each induced by a difference set of domain objects

Parameters and structures are reused within a BN and across different BNs

Models encode correlation across multiple objects, allowing collective inference

1.10 Structured CPDs

General CPD

- CPD $P(X|Y_1, \dots, Y_k)$ specifies distribution over X for each assignment y_1, \dots, y_k
- can use any function to specify a factor $\phi(X, Y_1, \dots, Y_k)$ such that
- $\sum_X \phi(x, y_1, \dots, y_k) = 1$ for all y_1, \dots, y_k

Context-Specific Independence

$P \models (X \perp_C Y|Z, c)$ assignment to c

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

1.11 Deterministic CPDs

non-tabular CPD arises when a variable X is deterministic function of its parents Pa_X . There is a function $f: Val(Pa_X) \rightarrow Val(X)$ such that

$P(x|pa_X) = 1$ only $x = f(pa_X)$, otherwise is 0

For example, the the case of binary-valued variables, X might be the "or" of its parents. In a continuous domain, we might want to assert in $P(X|Y, Z)$ that X is equal $Y+Z$

1.12 Tree- structured CPDs

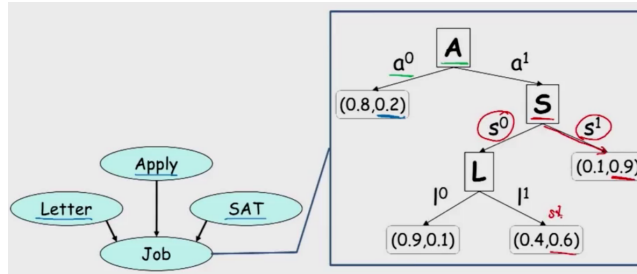


Figure 1.10: Tree-structured CPD

context-specific in-dependencies

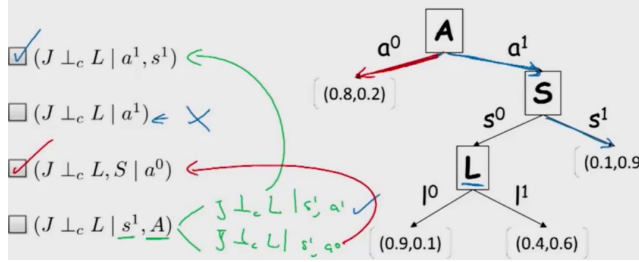


Figure 1.11: context-specific in-dependencies

1.13 Independence of Causal Influence

Noisy OR CPD

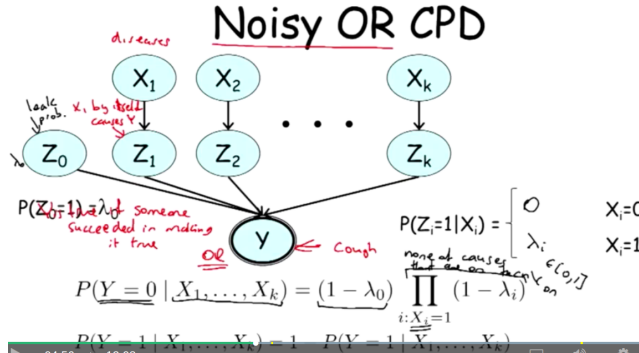


Figure 1.12: Noisy OR CPD

1.13.1 Independence of Causal Influence

1.13.2 Sigmoid CPD

1.14 Continue Variables

Use Gaussian model

1.15 Exercise: Tree CPD - context specific In-dependencies

Base on Fig 1.14: $(E \perp_c D | b^1)$ and $(E \perp_c D, B | a^1)$

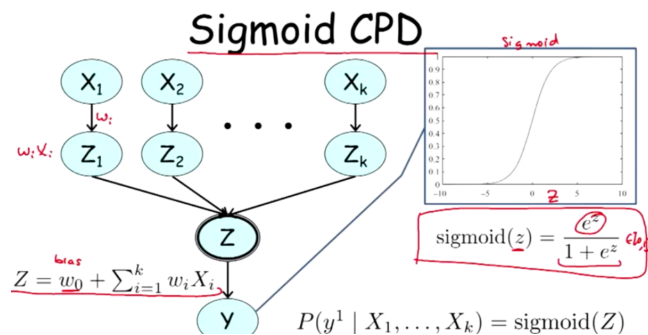


Figure 1.13: Sigmoid CPD

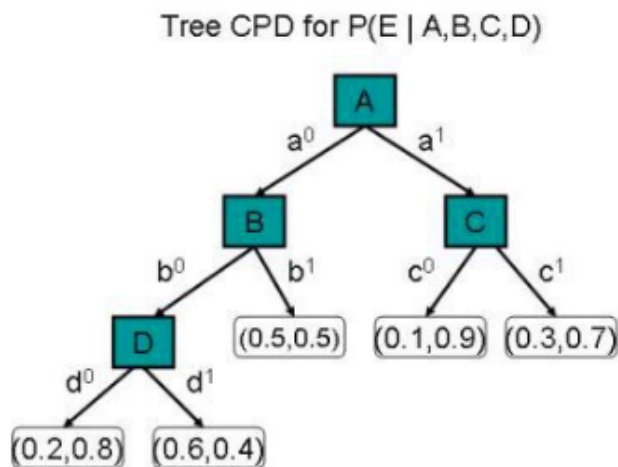


Figure 1.14: Which are context-specific independences?

1.16 Markov Networks: Undirected Models

1.16.1 Pairwise Markov Networks

is an undirected graph whose nodes are X_1, \dots, X_n and each edge $X_i - X_j$ is associated with a factor (potential) $\phi_{ij}(X_i X_j)$

iiiiiii HEAD =====

1.16.2 General Gibbs Distribution

Parameters: General factors $\phi_i(D_i)$

Gibbs distribution

Set of factors: $\Phi = \phi_1(D_1), \dots, \phi_k(D_k)$

$\widetilde{P}_\Phi(X_1, \dots, X_n) = \prod_{i=1}^k \phi_i(D_i)$

$$Z_\Phi = \sum_{X_1, \dots, X_n} \widetilde{P}_\Phi(X_1, \dots, X_n)$$

$$P_\Phi(X_1, \dots, X_n) = \frac{1}{Z_\Phi} \widetilde{P}_\Phi(X_1, \dots, X_n)$$

1.16.3 Induced Markov Network

Induced Markov network H_Φ has an edge $X_i - X_j$ whenever there exists a factor that includes X_i and X_j

1.16.4 Factorization

P factorizes over H if there exist $\Phi = \phi_1(D_1), \dots, \phi_k(D_k)$
 such that $P = P_\Phi$
 H is the induced graph for Φ

1.16.5 Flow of Influence

1.16.6 Active Trails

A trail $X_1 - \dots - X_n$ is active given Z if no X_i is in Z

1.16.7 Task-specific prediction

what are the input values, target variables

Examples: Image segmentation: inputs are pixels values and preprocessed features, target is the class for every pixel

1.16.8 Correlated Features

Naive Bayes assume that features are independent but they are correlated: add the edges to capture the correlation

Correlated Features Representation (CRF) representation

$\Phi = \phi_1(D_1), \dots, \phi_k(D_k)$: Gibbs distribution

$$\widetilde{P}_\Phi(X_1, \dots, X_n) = \prod_{i=1}^k \phi_i(D_i)$$

$$Z_\Phi(X) = \sum_Y \widetilde{P}_\Phi(X, Y)$$

$$P_\Phi(Y|X) = \frac{1}{Z_\Phi(X)} \widetilde{P}_\Phi(X, Y) : \text{a family of conditional distribution}$$

1.16.9 CRFs and Logistic Model

$$\phi_i(X_i, Y) = \exp(w_i X_i) = 1, Y = 1$$

Logistic is a very simple CRF

1.16.10 Independences in Markov Networks

$$I(H) = (X \perp Y|Z) : sep_H(X, Y|Z)$$

If P satisfies $I(H)$, we say that H is an I-map (independency map) of P

IF P factorizes over H , then H is an I-map of P

For a positive distribution P , if H is an I-map for P , then P factorizes over H

1.16.11 Capturing Independence in P

$$I(P) = (X \perp Y|Z) : Phold(X \perp Y|Z)$$

P factorizes over $G \rightarrow G$ is an I-map for P : $I(G) \subseteq I(P)$ but not always vice versa

1.16.12 Minimal I-map

I-map without redendant edges

Minimal I-map may still not capture $I(P)$

1.16.13 Perfect Map

Perfect map: $I(G) = I(P)$

— G perfectly captures independencies in P

Markov Network as a perfect map: $I(H) = I(P)$

— H perfectly captures independencies in P

1.16.14 Uniqueness of Perfect Map

1.16.15 I-equivalence

Definition: Two graphs G_1 and G_2 over X_1, \dots, X_n are I-equivalent if $I(G_1) = I(G_2)$

Converting BNs to MNs (vice versa) loses independencies

— BN to MN: v-structures

— MN to BN: must add triangulating edges to loops

1.17 Local Structure in Markov Networks

1.17.1 Log-Linear Representation

$$\tilde{P} = \prod_i \phi_i(D_i) \rightarrow \tilde{P} = \exp(-\sum_j w_j f_j(D_j))$$

— Each feature f_j has a scope D_j

— Different features can have same scope

— w_j is coefficient

1.17.2 Ising Model

$$E(x_1, \dots, x_n) = - \sum_{i < j} w_{i,j} x_i x_j - \sum_i u_i x_i$$

$$x_i \in -1, +1, f_{i,j}(X_i, X_j) = X_i * X_j$$

$$P(X) \propto e^{-1/T * E(X)}$$

T: temperature

1.17.3 Metric MRFs

- All X_i take values in label space V . want X_i and X_j to take similar values
 - Distance function $\mu : V \times V \rightarrow R^+$
 - Reflexivity: $\mu(v, v) = 0$ for all v
 - Symmetry: $\mu(v_1, v_2) = \mu(v_2, v_1)$ for all v_1, v_2
 - Triangle inequality: $\mu(v_1, v_2) \leq \mu(v_1, v_3) + \mu(v_3, v_2)$ for all v_1, v_2, v_3
- $f_{i,j}(X_i, X_j) = \mu(X_i, X_j)$
- $\exp(-w_{ij} f_{ij}(X_i, X_j)) w_{ij} > 0$
- values of X_i and X_j far in μ then lower probability

1.17.4 Shared Features in Log-Linear Models

In most MRFs, same feature and weight are used over many scopes

Ising Model

$$E(x_1, \dots, x_n) = - \sum_{(i,j) \in \text{Edges}} w x_i x_j - \sum_i u_i x_i$$

same weight for every adjacent pair

Repeated Features

- Need to specify for each feature f_k a set of scopes $\text{Scopes}[f_k]$
- For each $D_k \in \text{Scopes}[f_k]$ we have a term $w_k f_k(D_k)$ in the energy function

1.17.5 Clique

A subgraph over X is complete if every two nodes on X are connected by some edges. The set X is often called a clique.

The Markov Network can be factorized by some factors if scope of this factor is a clique

value of factor must greater than or equal zero.

Chapter 2

Decision Making

2.1 Simple Decision Making

A simple decision making situation D:

- A set of possible actions $Val(A) = a^1, \dots, a^k$
- A set of states $Val(X) = x^1, \dots, x^N$
- A distribution $P(X|A)$
- A utility function $U(X,A)$: take action how state is this, allow to evaluate the action

Expected Utility

$$EU[D[a]] = \sum_x P(x|a)U(x,a)$$

Want to choose action a that maximizes the EU

$$a^* = \operatorname{argmax}_a EU[D[a]]$$

2.1.1 Simple Influence Diagram

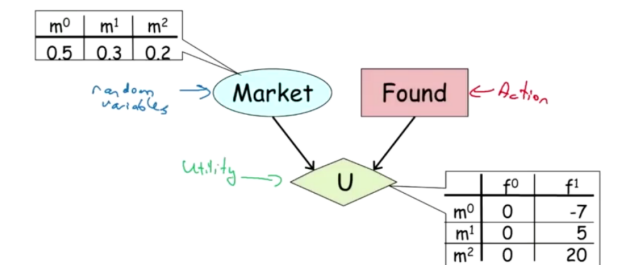


Figure 2.1: Simple decision. Found is action not a random variable so Found does not have CPD. f^0 : not found company, f^1 : found company

$$EU(f^0) = 0$$

$$EU(f^1) = 0.5 * (-7) + 0.3 * 5 + 0.2 * 20 > 0$$

found company is optimal action

2.1.2 Information Edges

Decision rule θ at action node A is a CPD: $P(A|Parents(A))$

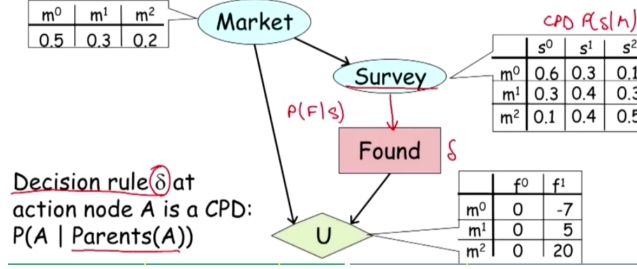


Figure 2.2: Decision rule at action node

2.1.3 Expected Utility with Information

$$ED[D[\delta_A]] = \sum_{x,a} P_{\delta_A}(x,a)U(x,a)$$

Want to choose the decision rules δ_A that maximizes ED

2.1.4 Finding MEU Decision Rules

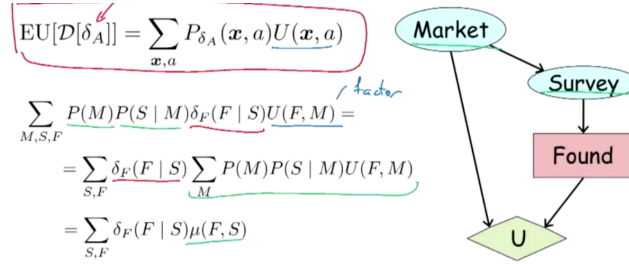


Figure 2.3: Finding MEU Decision Rules

$$P(f^1|s^0) = \sum_M P(M) * P(S = s^0|M) * U(F = f^1, M)$$

2.1.5 More Generally

$Z = Pa_A$ (observation prior to A) $W = X_1, \dots, X_n - Z$

$$EU[D[\delta_A]] = \sum_{x,a} P_{\delta_A}(x,a)U(x,a)$$

$$EU[D[\delta_A]] = \sum_{X_1, \dots, X_n, A} ((\prod_i P(X_i|Pa_{X_i}))U(Pa_U)\delta_A(A|Z))$$

$$EU[D[\delta_A]] = \sum_{Z,A} \delta_A(A|Z) \sum_W ((\prod_i P(X_i|Pa_{X_i}))U(Pa_U))$$

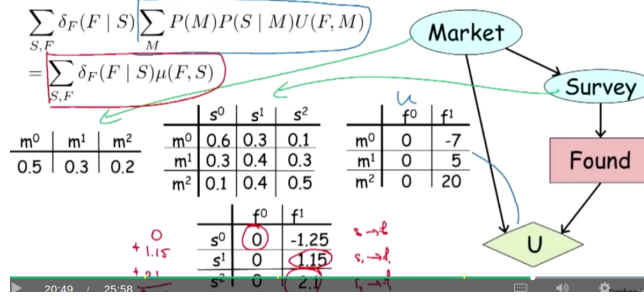


Figure 2.4: Finding MEU

$EU[D[\delta_A]] = \sum_{Z,A} \delta_A(A|Z)\mu(A, Z)$: eliminate all variable except A,Z
 $\delta_A^*(a|z) = 1$ if $a = \operatorname{argmax}_A \mu(A, z)$ otherwise is 0

2.2 Utility Function

2.2.1 Utilites and Preferences