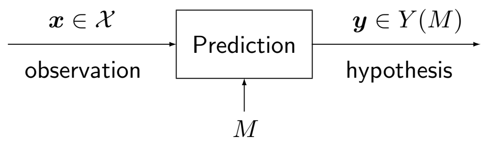
# 2. Models for Statistical Structured Prediction










- Real life
  - **Fragility**: Lack of robustness due to an imperfect representation of input objects [ Uncertainty regarding representations of inputs ]
  - **Ambiguity**: Several interpretations (hypothesis) are plausible due to variability and difficulty of the task. [ Uncertainty regarding outputs ]
  - **Incompleteness**: Inadequate models due to incomplete and insufficient knowledge [ uncertainty ! ]
- (Statistical) Solution Based on statistical decision theory $f : \mathcal{X} \to Y(M)$

$$\widehat{\boldsymbol{y}} = f(\boldsymbol{x}) = \underset{\boldsymbol{y} \in Y(M)}{\arg\max} \; P(\boldsymbol{y} \mid \boldsymbol{x})$$





Discriminative models (conditional probability) $P(y \mid x)$

$$f(x) = \underset{y \in Y(M)}{\arg\max} \; P(y \mid x)$$

Generative models (join probability) $P(x, \, y)$

$$f(x) = \underset{y \in Y(M)}{\arg\max} \; P(x, \, y) = \underset{y \in Y(M)}{\arg\max} \; P(x \mid y) \cdot P(y)$$





## Three Key Components

- Statistical model: used to replace the unknown real models

  $P(y \mid x)$ or $P(x \mid y)$ and $P(y)$ vs $P_\theta(y \mid x)$ or $P_\theta(x \mid y)$ and $P_\theta(y)$

  Discriminant Function, Neural networks, Gaussian mixtures, models with hidden variables (HMM, alignment models), log-linear models, etc.

- Decision rule (search or decoding): Sometimes hard !

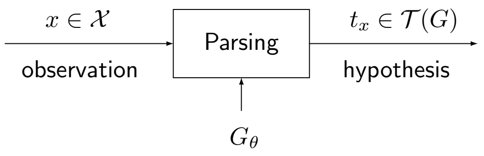
  Dynamic Programming, A* search, etc.

- Training criterion to learn the unknown parameters $\theta$ from training data

  Maximum Likelihood Estimation, Maximum a Posterior Estimation, Minimum Classification error, etc.

Specific algorithms depend on statistical models

## Statistical Structured Prediction: parsing




$$t_x \in \mathcal{T}(G): \ \texttt{yield}(t) = x \quad \textbf{iff} \quad S \stackrel{+}{\Rightarrow} x \quad \textbf{iff} \quad x \in L(G)$$

(Statistical) Solution based on statistical decision theory: **Probabilistic models**

$$x \in L(G) \quad \Longrightarrow \quad P(x\,;G_\theta) \ \equiv \ P(x\,;\theta) \ \equiv \ P_\theta(x)$$

where $\theta$ is the parameter vector of the probabilistic model $G_\theta$



## Probabilistic Languages

- Language $L \subseteq \Sigma^*$, given an alphabet $\Sigma$
- Language generated by a grammar $L(G) \subseteq \Sigma^*$, given an alphabet $\Sigma$ and a grammar $G = (\Sigma, N, S, \mathcal{P})$

  $$L(G) = \{x \mid x \in \Sigma^* : \ S \stackrel{+}{\Rightarrow} x\}$$

- Probabilistic language $(L, \phi)$, given an alphabet $\Sigma$
  - $L \subseteq \Sigma^*$ characteristic language
  - $\phi : \Sigma^* \longrightarrow [0,1]$ computable probabilistic function:
    - i) $x \notin L \implies \phi(x) = 0 \qquad \forall x \in \Sigma^*$
    - ii) $x \in L \implies 0 < \phi(x) \leq 1 \qquad \forall x \in \Sigma^*$
    - iii) $\sum_{x \in L} \phi(x) = 1$



## Probabilistic Languages

- Example [Booth-Thompson,73]

  Given the alphabet $\Sigma = \{a, b\}$, the following language is defined:

  $L = \{a^n b^n \mid n \geq 0\}$, where $\phi(x) = 0$, $\forall x \notin L$ and $\phi(a^n b^n) = \frac{1}{e\ n\,!}$

  $$\sum_{x \in L} \phi(x) = \sum_{0 \leq n \leq \infty} \frac{1}{e\ n\,!} = \frac{1}{e} \sum_{0 \leq n \leq \infty} \frac{1}{n\,!} = \frac{1}{e} e = 1$$

- Theorem [Wetherell,80]

  Let $(L, \phi)$ be an infinite probabilistic language, then for each $\epsilon > 0$ there exists an $n \geq 0$, such as:

  $$\mid x \mid \geq n \Longrightarrow \epsilon > \phi(x)$$



## Course Coverage

| Problems | Models | |
|---|---|---|
| Non-structured | Generative | Naive Bayes classifier |
| | Discriminative | Logistic Regression<br>Perceptron algorithm<br>Support Vector Machines<br>Neural networks |
| Structured | Generative | Probabilistic Finite State Automata<br>Hidden Markov Models<br>Probabilistic Context-Free Grammars |
| | Discriminative | Conditional Random Fields<br>Structured Perceptron<br>Structured Support Vector Machines<br>(Encoder-Decoder) Recurrent Neural Network |

## PROBABILISTIC FINITE-STATE AUTOMATON

**Definition**. A probabilistic automaton $A$ over a probabilistic semiring is a tuple $A = (\Sigma, Q, \delta, I, F, P)$, where

- $\Sigma$ is the alphabet;
- $Q$ is a finite set of states;
- $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is a set of transitions;
- $I : Q \to \mathbb{R}^+$ (initial-state probabilities);
- $F : Q \to \mathbb{R}^+$ (final-state probabilities);
- $P : \delta \to \mathbb{R}^+$ (transition probabilities).

$I$, $F$, and $P$ are probabilistic functions that must satisfy:

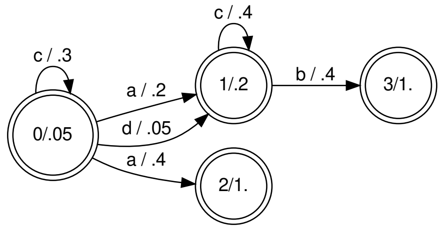
$$\sum_{q \in Q} I(q) = 1,$$

$$\forall q \in Q \qquad F(q) + \sum_{a \in \Sigma; q' \in Q} P(q, a, q') = 1.$$



## PROBABILISTIC FINITE-STATE AUTOMATON

$$I(q_0) = 1$$
$$\forall i \neq 0, \; I(q_i) = 0$$




$$P_A(acb,\; q_0q_1q_1q_3) = I(q_0) \cdot P(q_0, a, q_1) \cdot P(q_1, c, q_1) \cdot P(q_1, b, q_3) \cdot F(q_3) = 0{,}032$$

$$P_A(a,\; q_0q_1) = I(q_0) \cdot P(q_0, a, q_1) \cdot F(q_1) = 0{,}04$$
$$P_A(a,\; q_0q_2) = I(q_0) \cdot P(q_0, a, q_2) \cdot F(q_2) = 0{,}4$$

$$P_A(a) = P_A(a,\; q_0q_1) + P_A(a,\; q_0q_2) = 0{,}44$$



## PROBABILISTIC FINITE-STATE AUTOMATON

- The input string $x \in \Sigma^*$ will be accepted by the PFSA $A$, if there exists a path $\pi$ (sequence of transitions), $s_0 s_1 \dots s_k$ $(|x| \leq k)$, such that:

$$\pi = (s_0, x_1, s_1) \cdot (s_1, x_2, s_2) \cdots (s_{k-1}, x_k, s_k)$$

- The probability that the PFSA $A$, will accept the input string $x \in \Sigma^*$ through the path (sequence of transitions) $\pi$ is:

$$P_A(x, \pi) = I(s_0) \cdot \left( \prod_{i=1}^{k} P(s_{i-1}, x_i, s_i) \right) \cdot F(s_k)$$

- The probability that the PFSA $A$, will accept the input string $x \in \Sigma^*$

$$P_A(x) = \sum_{\pi \in \Pi_A(x)} P_A(x, \pi)$$



## PROBABILISTIC FINITE-STATE AUTOMATON

- Probability of a path

$$P_A(x, \pi) = I(s_0) \cdot \left( \prod_{i=1}^{k} P(s_{i-1}, x_i, s_i) \right) \cdot F(s_k)$$

- Probability of a string $\qquad P_A(x) = \sum_{\pi \in \Pi_A(x)} P_A(x, \pi)$

- Probability of the best path $\qquad \widehat{P_A}(x) = \max_{\pi \in \Pi_A(x)} P_A(x, \pi)$

- Best path $\qquad \widehat{\pi}(x) = \arg\max_{\pi \in \Pi_A(x)} P_A(x, \pi)$

- Language accepted by a probabilistic automaton

$$L(A) = \{x \in L(A) \mid P_A(x) > 0\}$$

## probabilistic finite-state automaton: properties

➢ Consistency of PFSA: A probabilistic automaton $A$ is consistent **iff**:

$$\sum_{x \in L(A)} P_A(x) = 1$$

➢ **Definition:** A state of a PFSA $A$ is useful if it appears in at least one valid path of $\Pi_A$.

➢ **Proposition:** A PFSA is consistent if all its states are useful.



## other finite-state models: HMM

**Definition**. A hidden Markov model (HMM) $M$ over a probabilistic semiring is a tuple $M = (\Sigma,\ Q,\ F,\ I,\ T,\ E)$, where

➢ $\Sigma$ is the alphabet;
➢ $Q$ is a finite set of states;
➢ $q_f \in F \subseteq Q$ is a special (final) state;
➢ $I : (Q - \{q_f\}) \to \mathbb{R}^+$ is an initial state probability function;
➢ $T : (Q - \{q_f\}) \times Q \to \mathbb{R}^+$ is a state state probability function;
➢ $E : (Q - \{q_f\}) \times \Sigma \to \mathbb{R}^+$ is a state-based symbol emision probability function.

$I$, $T$, and $E$ are probabilistic functions that must satisfy:

$$\begin{aligned} \sum_{q \in (Q - \{q_f\})} I(q) &= 1, \\ \forall q \in (Q - \{q_f\}) \quad \sum_{q' \in Q} T(q, q') &= 1 \\ \forall q \in (Q - \{q_f\}) \quad \sum_{a \in \Sigma} E(q, a) &= 1 \end{aligned}$$



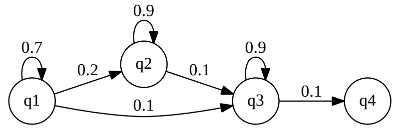
## Hidden Markov Models

**Symbol emission probabilities**

| | |
|---|---|
| a | 0.9 |
| b | 0.1 |

| | |
|---|---|
| a | 0.1 |
| b | 0.9 |

| | |
|---|---|
| a | 0.9 |
| b | 0.1 |




**State transition probabilities**
**Initial state probabilities** $I(q_1) = 1;\quad I(q_2) = I(q_3) = I(q_4) = 0$

$$P_M(aba,\ q_1q_2q_3q_4) = I(q_1)\ E(a,q_1)\ T(q_1,q_2)\ E(b,q_2)\ T(q_2,q_3)\ E(a,q_3)\ T(q_3,q_4) = 1{,}458 \cdot 10^{-3}$$

$$P_M(aba,\ q_1q_1q_3q_4) = I(q_1)\ E(a,q_1)\ T(q_1,q_1)\ E(b,q_1)\ T(q_1,q_3)\ E(a,q_3)\ T(q_3,q_4) = 0{,}567 \cdot 10^{-3}$$

$$P_M(aba,\ q_1q_3q_3q_4) = I(q_1)\ E(a,q_1)\ T(q_1,q_3)\ E(b,q_3)\ T(q_3,q_3)\ E(a,q_3)\ T(q_3,q_4) = 0{,}729 \cdot 10^{-3}$$

$$P_M(aba) = 2{,}754 \cdot 10^{-3}$$



## Hidden Markov Models

➢ The probability that the HMM $M$, will accept the input string $x = x_1 \dots x_k \in \Sigma^*$ through the path (sequence of transitions) $\pi = s_1 \dots s_k$ is:

$$P_M(x, \pi) \;=\; I(s_1) \cdot \prod_{i=2}^{k} T(s_{i-1}, s_i) \cdot \prod_{i=1}^{k} E(x_i, s_i)$$

➢ The probability that the HMM $M$, will accept the input string $x \in \Sigma^*$

$$P_M(x) \;=\; \sum_{\pi \in \Pi_A(x)} P_M(x, \pi)$$

➢ Given an HMM $M$, there exists a PFSA $A$, such that $P_M(x) \;=\; P_A(x) \quad \forall x \in \Sigma^*$

# Probabilistic Context-Free Grammars

➢ Probabilistic Context-Free Grammars: $G_\theta = (G, P)$

- ➢ $G = (\Sigma, N, S, \mathcal{P})$ characteristic grammar
- ➢ $P : \mathcal{P} \to ]0,1]$ probabilities of rules:

$$\forall A_i \in N \qquad P(A_i \to \alpha_j) \equiv P(r_{ij}) \equiv P(A_i \to \alpha_j \mid A_i) \equiv P(\alpha_j \mid A_i)$$

where $n_i$ is the number of rules with $A_i$ in the left side of rule:

$$\sum_{1 \le j \le n_i} P(\alpha_j \mid A_i) = 1$$



# Probabilistic Context-Free Grammars

➢ Probabilistic derivation

Given a sequence of stochastic events:

$$S = \alpha_0 \overset{r_1}{\Rightarrow} \alpha_1 \overset{r_2}{\Rightarrow} \alpha_2 \cdots \alpha_{m-1} \overset{r_m}{\Rightarrow} \alpha_m = x$$

the probability of $x$ being generated by $G_\theta = (G, P)$ from the rule sequence $t_x = r_1 \cdots r_m$, is:

$$P_\theta(x, t_x) = P(r_1) \cdot P(r_2 \mid r_1) \cdots P(r_m \mid r_1 \cdots r_{m-1})$$

- ➢ **problem**: computation of the probabilities
- ➢ **restriction**: $P(r_j \mid r_1 \cdots r_{j-1}) = P(r_j)$

$$P_\theta(x, t_x) = \prod_{j=1 \cdots m} P(r_j)$$



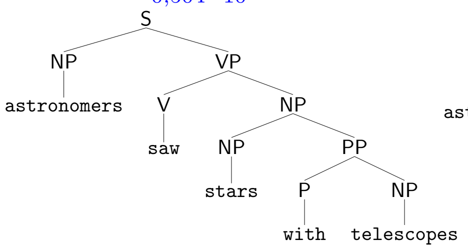
# Probabilistic Context-Free Grammars: example

Example: A simple Context-Free Grammars [Manning and Schütze, 2002]

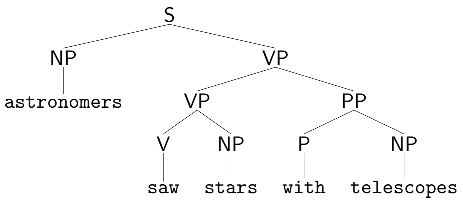
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| S → NP VP | $1{,}0$ | VP → V NP | $0{,}7$ | V → `saw` | $1{,}0$ | NP → `saw` | $0{,}04$ |
| NP → NP PP | $0{,}4$ | VP → VP PP | $0{,}3$ | NP → `astronomers` | $0{,}1$ | NP → `stars` | $0{,}18$ |
| PP → P NP | $1{,}0$ | P → `with` | $1{,}0$ | NP → `ears` | $0{,}18$ | NP → `telescopes` | $0{,}1$ |

S $\overset{1,0}{\Rightarrow}$ NP VP $\overset{0,1}{\Rightarrow}$ `astronomers` VP $\overset{0,7}{\Rightarrow}$ `astronomers` V NP $\overset{1,0}{\Rightarrow}$ `astronomers saw` NP $\overset{0,4}{\Rightarrow}$
`astronomers saw` NP PP $\overset{0,18}{\Rightarrow}$ `astronomers saw stars` PP $\overset{1,0}{\Rightarrow}$ `astronomers saw stars` P NP $\overset{1,0}{\Rightarrow}$
`astronomers saw stars with` NP $\overset{0,1}{\Rightarrow}$ `astronomers saw stars with telescopes` $= 0{,}504 \cdot 10^{-3}$

$0{,}504 \cdot 10^{-3}$




$0{,}378 \cdot 10^{-3}$






# Probabilistic Context-Free Grammars

➢ Probability of a parse tree

$$P_\theta(x, t_x) = \prod_{j=1 \cdots m} P(r_j)$$

➢ Probability of a string

$$P_\theta(x) = \sum_{t_x \in \mathcal{T}_x} P_\theta(x, t_x)$$

➢ Probability of the best parse tree

$$\widehat{P_\theta}(x) = \max_{t_x \in \mathcal{T}_x} P_\theta(x, t_x)$$

➢ Language generated by a probabilistic grammar

$$L(G_\theta) = \{x \in L(G) \mid P_\theta(x) > 0\}$$

## Probabilistic Context-Free Grammars: properties

➢ Consistent grammars

A probabilistic gramar $G_\theta = (G, P)$ is consistent **iff**:

$$\sum_{x \in L(G)} P_\theta(x) = 1$$

➢ Theorem [Booth-Thompson,73]

There exist probabilistic languages $(L, \phi)$ that can not be generated by a probabilistic grammar $G_\theta = (G, P)$

➢ Example.- Let $L = \{a^n b^n \mid n \geq 0\}$ be a probabilistic language:

$$\phi(a^n b^n) = \frac{1}{e\ n!}$$

There is not any $G_\theta$ such that $\phi(x) = P_\theta(x) \quad \forall x \in L$



## Probabilistic Graphical Models

**Probabilistic Graphical Models** are an elegant framework that combines uncertainty (probabilities) and logical structure (independence constraints) to make simplifying assumptions about which variables affect which other variables.

These assumptions can often be represented graphically, leading to whole sets of models known collectively as Graphical Models (GMs)

Why do we need Graphical Models?

➢ GMs are the basis for the probabilistic approach to *Intelligent Systems*.

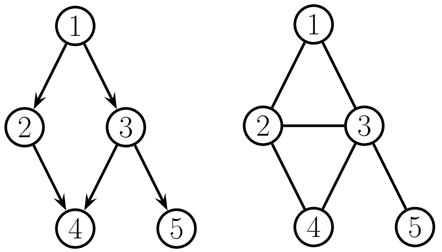
➢ GMs are a compact representation of joint probability distributions using graphs, constituting a perfect match between **Probability Theory** and **Graph Theory**.

➢ GMs allow us to abstract out the conditional independence relationships between the variables from the details of their parametric forms.

➢ GMs generalize to Neural Networks and Hidden Markov Models, among others.



## Probabilistic Graphical Models




[from K.P. Murphy, 2012]

**Nodes** represent random variables.

The **graph** represents a set of independences and factorizes a distribution.

Graphical models: taxonomy

➢ Bayesian Networks based on **Directed Graphs**.

➢ Markov Random Fields based on **Undirected Graphs**.



## Probabiistic Graphical Models

Key components

➢ **Inference**: deduce probability distributions from other given.

➢ **Learning**: obtain the probabilistic model from observations.

Applications

➢ Medical diagnosis, fault detection, ...

➢ **Computer Vision**: image segmentation, 3D reconstruction, scene analysis, ...

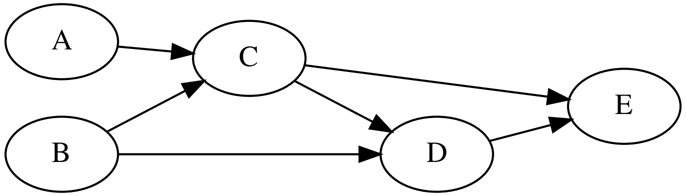
➢ **Natural Language Processing**: speech recognition, information extraction, machine translation, ...

➢ **Robotics**: planning, detection, ...

## Bayesian Networks

A Bayesian Network (BN) is a graphical model that represents a set of random variables and their conditional dependencies via a Directed Acyclic Graph (DAG).

**Example**




**Nodes** represent to random variables (discrete or continuous)

**Edges** represent statistical dependencies between the variables



## Bayesian Networks

➢ There are two extreme cases:

1) $p(x_1, x_2, \ldots, x_T) = p(x_1) \cdot p(x_2|x_1) \ldots p(x_T|x_1, \ldots, x_{T-1})$

$$= p(x_1) \prod_{t=2}^{T} p(x_t|x_1, \ldots, x_{t-1}) \qquad \text{requires } 2^T - 1 \text{ parameters}$$

2) $p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t)$ requires $T$ parameters

➢ If not all dependencies are possible, the complete factorization will be reflected in the graph (DAG) associated with the BN.

➢ A BN with nodes $x_1, \ldots, x_T$ defines a joint probability distribution:

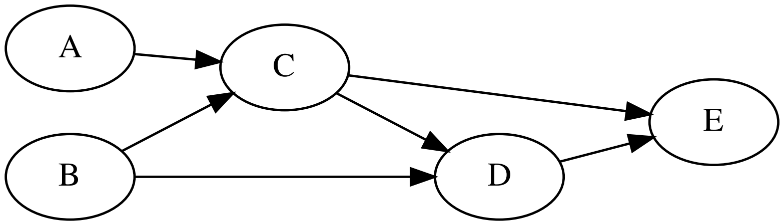
$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t \,|\, \varphi(x_t))$$

where $\varphi(x_t)$ denotes the dependencies associated with node $x_t$



## Bayesian Networks

### Example




$$p(A, B, C, D, E) = p(A) \cdot p(B) \cdot p(C|A, B) \cdot p(D|B, C) \cdot p(E|C, D)$$



## BNs: conditional independence

Representing knowledge through the notion of conditional independence

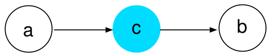
➢ Two events, $a$ and $b$, are **(unconditionally) independent** and we denote it as $(a \perp\!\!\!\perp b)$ if:

$$p(a, b) = p(a) \cdot p(b) \qquad \text{or} \qquad p(a \mid b) = p(a)$$

➢ Two events, $a$ and $b$, are **conditionally independent** given an event $c$, and we denote it as $(a \perp\!\!\!\perp b \mid c)$ if:

$$p(a \mid b, c) = p(a \mid c) \qquad \text{or} \qquad p(a, b \mid c) = p(a \mid c) \cdot p(b \mid c)$$

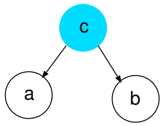
# BNs: rules of conditional and unconditional independence




**Causal direction**: $(a \perp\!\!\!\perp b \mid c) \quad (a \not\!\perp\!\!\!\perp b)$

$$P(a,b \mid c) = \frac{P(a)P(c \mid a)P(b \mid c)}{P(c)} = P(a \mid c)P(b \mid c)$$

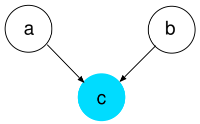
but $\quad P(a,b) \neq P(a)P(b)$




**Common parent**: $(a \perp\!\!\!\perp b \mid c) \quad (a \not\!\perp\!\!\!\perp b)$

$$P(a,b \mid c) = \frac{P(c)P(a \mid c)P(b \mid c)}{P(c)} = P(a \mid c)P(b \mid c)$$

but $\quad P(a,b) \neq P(a)P(b)$




**V-structure**: $(a \not\!\perp\!\!\!\perp b \mid c) \quad (a \perp\!\!\!\perp b)$

$$P(a,b \mid c) \neq P(a \mid c)P(b \mid c)$$

but $\quad P(a,b) = \sum_c P(a)P(b)P(c \mid a,b) = P(a)P(b)$



# BNs: inference

The purpose is to estimate the posterior probability of some variable $x$ from the joint distributions associated with a BN, given some evidence $y$ (regardless of the values of the rest of the variables $z$).

$$P(x \mid y) = \frac{P(x,y)}{P(y)} \quad \text{con:} \quad P(x,y) = \sum_z P(x,y,z); \quad P(y) = \sum_{x,z} P(x,y,z);$$

The goal is to efficiently calculate $P(x,y)$ and $P(y)$.

➢ The usefulness of calculating the posterior probability.

Prediction.- What is the probability of observing a symptom knowing that the patient has a particular disease?

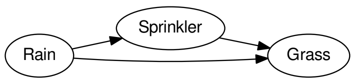
Diagnosis.- What is the probability that a particular disease is a correct diagnosis given some symptoms?



# BNs: a detailed example

$P(S \mid R)$

| Rain (R) | Sprinkler (S) s | r |
|---|---|---|
| n | 0.60 | 0.40 |
| y | 0.99 | 0.01 |




$P(R)$

| Rain (R) n | y |
|---|---|
| 0.8 | 0.2 |

| | | |
|---|---|---|
| Sprinkler | s: | stoped |
| | r: | running |
| Grass | d: | dry |
| | w: | wet |
| Rain | n: | not rain |
| | y: | yes it rains |

$P(G \mid S,R)$

| Rain (R) | Sprinkler (S) | Grass (G) d | w |
|---|---|---|---|
| n | s | 0.99 | 0.01 |
| n | r | 0.10 | 0.90 |
| y | s | 0.20 | 0.80 |
| y | r | 0.01 | 0.99 |

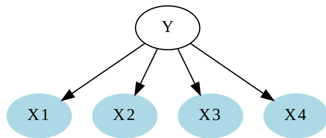
Joint distribution: $\quad P(R,S,G) = P(R)\ P(S \mid R)\ P(G \mid R,S)$

**Exercise**: What is the probability that the sprinkler will work if there is no rain and the grass is wet?



# BNs: examples (generative models)

## Classifiers




### Naive Bayes

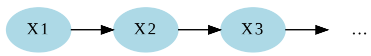
➢ Predicting a single class $y$ given a vector of features $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$

➢ Assumption: once the class label is known all the features are independent

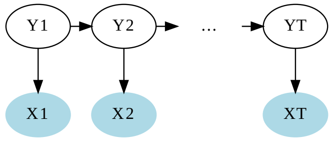
$$p(x_1, x_2, \ldots, x_T, y) = p(y) \prod_{t=1}^{T} p(x_t \mid y)$$

## BNs: examples (generative models)

### Sequential Prediction




### Sequence Labeling




**Markov model**

$$p(x_1, x_2, \ldots, x_T) \;=\; p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_2) \ldots \;=\; p(x_1) \prod_{t=1}^{T} p(x_t|x_{t-1})$$

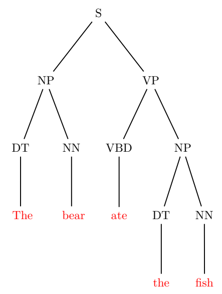
**(Bigram) Hiddden Markov Model**

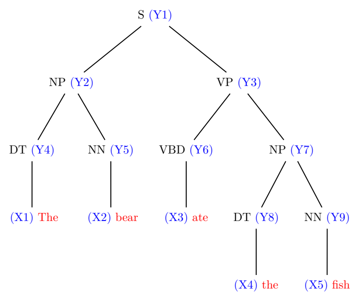
$$P(x_1^T, y_1^T) \;=\; \prod_{t=1}^{T} q(y_t|y_{t-1}) \cdot e(x_t|y_t)$$



## Bayesian Networks: examples (generative models)

### Parsing




**Probabilistic Context-Free Grammars**

$$t_x \;=\; [\, S = \alpha_0 \overset{r_1}{\Rightarrow} \alpha_1 \overset{r_2}{\Rightarrow} \alpha_2 \cdots \alpha_{m-1} \overset{r_m}{\Rightarrow} \alpha_m = \mathbf{x} \,] \qquad P(x_1^T, t_x) = \prod_{i=1 \cdots m} P(r_i);$$

$$P(r_i) \;=\; P(Y_j\, Y_k | Y_i) \quad \textbf{or} \quad P(r_i) \;=\; P(X_l | Y_i)$$



## Markov Random Fields

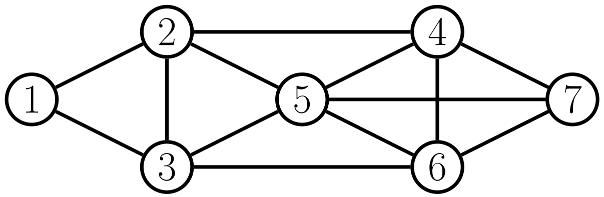
Bayesian Networks cannot perfectly represent all distributions.

Markov Random Fields (MRF) are graphical models based on Undirected Graphs.

The **nodes** represent the random variables.

The **edges** represent some notion of probabilistic interaction between neighboring nodes. Edges show which variables depend on each other.

**Example** [Kevin P. Murphy, 2012]





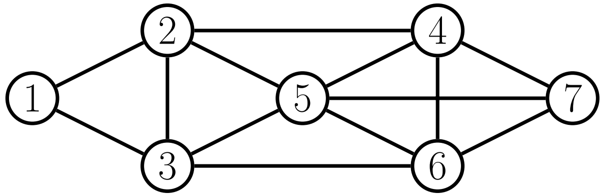

## Markov Random Fields

Conditional independence properties: **global Markov property** for MRFs

MRFs define conditional independence relationships via simple graph separation: for sets of nodes $A$, $B$, and $C$, we say,

$$A \perp B \,|\, C \quad \Leftrightarrow \quad C \text{ separates } A \text{ from } B \text{ in the graph.}$$

**Example**:



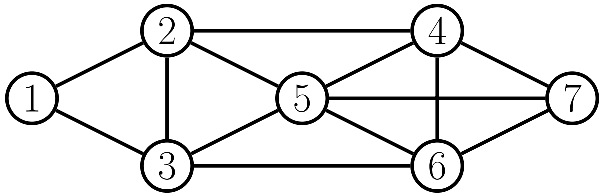

$$\{1,2\} \perp \{6,7\} \;|\; \{3,4,5\}$$

## Markov Random Fields

Definitions: A **clique** of an undirected graph is a fully connected subgraph.

A **maximal clique** is a clique that is not a subgraph of any other clique.

**Example**:



Cliques: $\{1,2\}; \ldots \{1,2,3\}; \{2,3,5\}; \{2,4,5\}; \{3,5,6\}; \{4,5,6\} \ldots \{4,5,6,7\};$

No-cliques: $\{1,2,3,5\}; \{2,3,4,5,6\}; \ldots$

maximal cliques: $\{1,2,3\}; \{2,3,5\}; \{2,4,5\}; \{3,5,6\}; \{4,5,6,7\};$



## Markov Random Fields: factorization

Definition.- Let $G$ be the undirected graph that underlies an MRF. A probability distribution $P_G$ defines a `factorization` over $G$ if it is associated with:

$$P_G(X_1, \ldots, X_T) = \frac{1}{Z} \prod_{C \in Q} \psi_C(V_C)$$

Where,

$V = \{X_1, X_2, \ldots, X_T\}$ is the set of random variables;

$Q$ is the set of all the (maximal) cliques of $G$;

$V_C$ is the subset of variables from clique $C$;

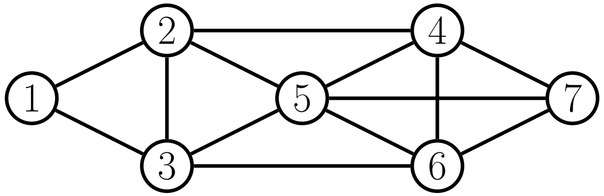
$\psi_C : Q \to \mathbb{R}^{>0}$ is the **potential function** for clique $c$, and

$Z$ is a normalization factor (constant) defined as:

$$Z = \sum_{X_1, \ldots, X_T} \prod_{C \in Q} \psi_C(V_C)$$



## Markov Random Fields: factorization



**Example** $Q = \{\{1,2,3\}; \{2,3,5\}; \{2,4,5\}; \{3,5,6\}; \{4,5,6,7\};\}$

$$P_G(1,2,3,4,5,6,7) = \frac{1}{Z} \psi_1(1,2,3) \cdot \psi_2(2,3,5) \cdot \psi_3(2,4,5) \cdot \psi_4(3,5,6) \cdot \psi_5(4,5,6,7)$$

MRFs are more powerful than BNs but are more challenging to deal with computationally.



## Markov Random Fields: factorization

It is often helpful to specify these potential functions by using exponential transformations,

$$\begin{aligned} P_G(X_1, \ldots, X_T) &= \frac{1}{Z} \prod_{C \in Q} \psi_C(V_C) \\ &= \frac{1}{Z} \prod_{C \in Q} \exp(-E_C(V_C)) \\ &= \frac{1}{Z} \exp(-\sum_{C \in Q} E_C(V_C)) \end{aligned}$$

Where $E_C : Q \to \mathbb{R}$ is a function, which is called the **energy function**.

There are types of energy functions that can be defined by *generalized linear functions*:

$$E_C(V_C) = -\sum_k \theta_{C,k} \; f_{C,k}(V_C)$$

## structured prediction: CRFs

Given $x = x_1, x_2, \dots, x_T \in \mathcal{X}^*$ and $y = y_1, y_2, \dots, y_T \in \mathcal{Y}^*$

➢ Discriminative models: **Conditional Random Fields (CRF)**

[Lafferty, McCallum, Pereira, 2001].

➢ Model parameters $\theta$

➢ Compatibility function $\phi(x, y; \theta) \rightarrow \mathcal{R}$

that that gives high positive scores to compatibles pairs $(x, y)$

➢ Conditional distribution:

$$p(y \mid x\,;\,\theta) = \frac{\exp\{\,\phi(x, y\,;\,\theta)\,\}}{Z(x\,;\,\theta)}$$

Where

$$Z(x\,;\,\theta) = \sum_{y' \in \mathcal{Y}^*} \exp\{\phi(x, y'\,;\,\theta)\}$$



## CRFs: feature engineering

### Compatibility Functions from (Bigram) Indicator Features

**Example** Named-Entity Recognition [C.Sutton and A.McCallum, 2012]

Entities = { (P) people, (O) organizations, (L) locations, (M) other}

$\mathcal{Y}$ = { B-P, I-P, B-O, I-O, B-L, I-L, B-M, I-M, O } [ CoNLL 2003 data set ]

| U.N. | official | Ekeus | heads | for | Baghdad |
|---|---|---|---|---|---|
| B-O | O | B-P | O | O | B-L |

➢ Label–label features

$$f^{LL}_{ij}(y_{t-1}, y_t, x_t) = \begin{cases} 1 & \textbf{if} \quad y_{t-1} = i \ \textbf{ and } \ y_t = j \quad \forall i, j \in \mathcal{Y} \\ 0 & \text{otherwise} \end{cases}$$

There are 9 different labels, so there are 81 label–label features.



## CRFs: feature engineering

➢ Label-word features

$$f^{LW}_{iv}(y_{t-1}, y_t, x_t) = \begin{cases} 1 & \textbf{if} \quad y_t = i \ \textbf{ and } \ x_t = v \quad \forall i \in \mathcal{Y},\ \forall v \in \mathcal{V} \\ 0 & \text{otherwise} \end{cases}$$

$\mathcal{V}$ is the set of all corpus words. For the CoNLL 2003 data set, $|\mathcal{V}| = 21,249$ words so there are $191,241$ label-word features.

➢ Label-observation features

$$f^{LO}_{ib}(y_{t-1}, y_t, x_t) = \begin{cases} 1 & \textbf{if} \quad y_t = i \ \textbf{ and } \ g_b(x_t) \quad \forall i \in \mathcal{Y} \\ 0 & \text{otherwise} \end{cases}$$

All observation functions are binary functions and are heuristically defined depending on the task



## CRFs: factored compatibility functions

➢ Let $f(y_{i-1}, y_i, x_i)$ be a vector of features

$$(\, f_1(y_{i-1}, y_i, x_i),\ f_2(y_{i-1}, y_i, x_i),\ \dots,\ f_K(y_{i-1}, y_i, x_i)\,)$$

➢ Given $f(.)$ and let $\theta \in \mathcal{R}^K$ be a weight matrix

$$\begin{aligned} \phi(x, y; \theta) &= \sum_{t=1}^{T} \theta\, f(y_{t-1}, y_t, x_t) \qquad (1) \\ &= \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k\, f_k(y_{t-1}, y_t, x_t) \end{aligned}$$

➢ Let $y_0 = \text{NULL}$, so the bigram $y_0\, y_1$ is defined for $t = 1$.

➢ This factorization will allow efficient algorithms since if $y \neq y'$ share bigrams then they will share scores

## Conditional Random Fields

$$p(y|x;\theta) = \frac{\exp\{\, \phi(x,y;\theta)\, \}}{Z(x;\theta)}$$

$$= \frac{\exp\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k \; f_k(y_{t-1},\; y_t,\; x_t)\ \}}{Z(x;\theta)} \qquad (2)$$

Where

$$Z(x;\theta) = \sum_{y' \in y^*} \exp \left\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k \; f_k(y'_{t-1},\; y'_t,\; x_t) \right\} \qquad (3)$$

- Features $f(.)$ are given (problem-dependent).
- $\theta \in \mathcal{R}^K$ are the parameters of the model.
- CRFs are log-linear models on the feature functions.



## CRFs: three problems

- Inference: Compute the probability of an output sequence $y$ for $x$

$$p(y\,|\,x\,;\,\theta) = \frac{1}{Z(x\,;\,\theta)} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k \; f_k\; (y_{t-1},\; y_t,\; x_t) \right\}$$

- Decoding: predict the best output sequence for $x$

$$\arg\max_{y \in \mathcal{Y}^*} p(y\,|\,x\,;\,\theta) = \arg\max_{y} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k \; f_k(y_{t-1},\; y_t,\; x_t) \right\}$$

- Prameter estimation: learn parameters $\theta$, given training data

$$\{\ (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\ \}$$



## BIBLIOGRAPHIC REFERENCES


- Probabilistic Context-Free Grammars
  - T.L.Booth and R.A.Thompson: Applying Probability Measures to Abstract Languages. IEEE Trans. on Compututers, 22(5):442–450. May 1973.
  - C.S.Wetherell: Probabilistic Languages: A Review and Some Open Questions. ACM Computing Surveys, 12(4):361–379. December 1980. `doi:10.1145/356827.356829`.
  - M.Collins: Probabilistic Context-Free Grammars. Course notes,Columbia University, 2011.
- Graphical Models
  - C.M.Bishop: Pattern Recognition and Machine Learning. Chap.8: Graphical Models. Springer, 2006.
  - D.Koller, N.Friedman, L.Getoor and B.Taskar: Graphical Models in a Nutshell in *An Introduction to Statistical Relational Learning*. MIT Press, 2007.
- Conditional Random Fields
  - J.Lafferty, A.McCallum and F.C.N.Pereira: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. ICMI:282-289. 2001.
  - C.Sutton and A.McCallum: An introduction to conditional random fields for relational learning in *An Introduction to Statistical Relational Learning*. MIT Press, 2007.