Lecture 11: Probabilistic graphical models

Meelis Kull
meelis.kull@ut.ee
Fall 2019
Why do we need ensemble methods?

- Bagging
- Random forest
- Weighted averaging
- Boosting
  - Intuitive explanation
  - AdaBoost algorithm
  - Alternative formulations
  - Interpretations
Acknowledgements

- Yaohang Li
  http://www.cs.odu.edu/~yaohang/cs480/ClassNotes/cs480-Lecture-24-12052017.pptx

- Svetlana Lazebnik
  http://slazebni.cs.illinois.edu/fall16/lec15_bayes_net_inference.pptx

- Christopher Bishop

- Gerton Lunter, Dirk Husmeier, Heng Li
  http://www.well.ox.ac.uk/dtc/bioinformatics_course/HMMDTC-2011.pptx

- Ying Nian Wu
  http://helper.ipam.ucla.edu/publications/gss2013/gss2013_11339.ppt
Lecture 11 – Probabilistic graphical models

• **Motivation**

• Factorisation of probability distributions

• Bayesian network example

• Inference example

• Bayesian networks and learning

• Markov model

• Hidden markov model (HMM)

• Markov random fields

• Directed vs undirected graphical models

• Implementation
Probabilistic models

- Let $X$ denote the variables we know about (feature values)
- Let $Y$ denote the target variables we are interested in (label)
- How to model relationship between $X$ and $Y$?
- Statistician’s approach:
  - Assume some underlying random process that generates the values for these variables
  - Use data to estimate the distribution of $X$ and $Y$
Probabilistic supervised learning

- $P(Y)$ – prior probability of the label $Y$
- $P(X|Y)$ – probability or density of features given the label (also known as the likelihood of the label)
- $P(Y|X)$ – posterior probability of the label given the features
- $P(X,Y)$ – joint distribution of features and label
Probabilistic models

• **Discriminative models** – modelling conditional distribution $P(\text{label} \mid \text{features})$
  – Logistic regression, neural nets with softmax
  – Conditional random fields

• **Generative models** – modelling joint distribution $P(\text{features}, \text{label})$
  – Naïve Bayes models
  – Gaussian mixture models
  – Bayesian networks
  – Gaussian process models
Probabilistic models

- **Discriminative models** – modelling conditional distribution $P(\text{label} | \text{features})$
  - Logistic regression, neural nets with softmax
  - Conditional random fields

- **Generative models** – modelling joint distribution $P(\text{features, label})$
  - Naïve Bayes models
  - Gaussian mixture models
  - Bayesian networks
  - Gaussian process models

How can we use generative models for classification?
Bayes classifier [from lecture 08]

• Bayes classifier (or Bayes-optimal classifier) is the best possible classifier:

\[ f^*(x) = \arg \max_i p_i^*(x) = \arg \max_i P(Y_i = 1 \mid X = x) \]

• No classifier can achieve accuracy higher than Bayes classifier
Maximum a posteriori (MAP)  
[from lecture 02]  

- Therefore, optimal is to follow the maximum a posteriori (MAP) decision rule

\[
\hat{y} = f(x) = \arg \max_{y \in Y} P(Y = y | X = x)
\]

- That is, predict the class that has the highest probability conditional to the given feature values
Why joint probabilities are enough for calculating MAP

\[ y_{MAP} = \arg\max_y P(Y = y | X = x) = \arg\max_y \frac{P(X = x, Y = y)}{P(X = x)} \]

\[ = \arg\max_y \frac{P(X = x, Y = y)}{\sum_{y'} P(X = x, Y = y')} \]

Therefore, we can now concentrate on calculating the joint probability distribution:

\[ P(X = x, Y = y) = P(X_1 = x_1, \ldots, X_m = x_m, Y = y) \]
Tennis dataset [from lecture 02]

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temp</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
## Tennis dataset – too few data

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temp</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
</tbody>
</table>

14 instances but $3 \times 3 \times 2 \times 2 \times 2 = 72$ different possible combinations of feature and label value combinations:

- \{Overcast, Rain, Sunny\}$ \times \{Cool, Hot, Mild\}$
- \{High, Normal\}$ \times \{Strong, Weak\}$
- \{Yes, No\}

Not enough data to estimate the joint distribution directly by counting.
Issue: not enough data

- The amount of required data grows exponentially with the number of features
- Not enough data to learn the joint probability distribution
- Need to use domain knowledge for reducing the amount of needed data
  - For example, if we know that the features are independent conditioned on the class, then we can do Naïve Bayes
- We will now consider probabilistical graphical models as a more general approach to solve this issue
Lecture 11 – Probabilistic graphical models

✓ Motivation

- **Factorisation of probability distributions**
- Bayesian network example
- Inference example
- Bayesian networks and learning
- Markov model
- Hidden markov model (HMM)
- Markov random fields
- Directed vs undirected graphical models
- Implementation
Reduction in data requirements

• Key idea to reducing the amount of required data:
  – Make different kind of independence assumptions (conditional and unconditional)
  – The more assumptions, the less data needed
Naïve Bayes [from lecture 02]

- Assume that any pair of features are conditionally independent given the label:
  \[ X_i \perp X_j \mid Y \]

- Then MAP decision rule gives us:

\[
\hat{y} = f(x) = \arg\max_{y \in \mathcal{Y}} P(Y = y \mid X = x)
= \arg\max_{y \in \mathcal{Y}} P(X = x \mid Y = y) P(Y = y)
= \arg\max_{y \in \mathcal{Y}} P(X_1 = x_1, \ldots, X_m = x_m \mid Y = y) P(Y = y)
= \arg\max_{y \in \mathcal{Y}} P(Y = y) \prod_{i=1}^{m} P(X_i = x_i \mid Y = y)
\]
Naïve Bayes [from lecture 02]

• Predict the class according to the following rule:

\[ \hat{y} = f(x) = \arg \max_{y \in \mathcal{Y}} P(Y = y) \prod_{i=1}^{m} P(X_i = x_i \mid Y = y) \]
Factorising a probability distribution

• Chain rule (applies for any random variables, e.g. $X_1, \ldots, X_n, Y$):

$$P(A_1, \ldots, A_k) = P(A_1 | A_2, A_3, \ldots, A_k) \cdot P(A_2 | A_3, \ldots, A_k) \cdots P(A_{k-1} | A_k) \cdot P(A_k)$$

  – For example, with $k=4$:

$$P(A_1, A_2, A_3, A_4) = P(A_1 | A_2, A_3, A_4) \cdot P(A_2 | A_3, A_4) \cdot P(A_3 | A_4) \cdot P(A_4)$$

• Conditional independencies allow to omit some variables from the conditioning

  – E.g., if $A_1 \perp A_3 \mid A_4$ and $A_2 \perp A_4 \mid A_3$ then:

$$P(A_1, A_2, A_3, A_4) = P(A_1 | A_2, A_4) \cdot P(A_2 | A_3) \cdot P(A_3 | A_4) \cdot P(A_4)$$
Lecture 11 –
Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
  • Bayesian network example
  • Inference example
  • Bayesian networks and learning
  • Markov model
  • Hidden markov model (HMM)
  • Markov random fields
  • Directed vs undirected graphical models
  • Implementation
Alarm story

- I have a burglar alarm installed at home
  - It is fairly reliable at detecting a burglary, but also responds on occasion to minor earth quakes.

- I also have two neighbors, John and Mary
  - They have promised to call me at work when they hear the alarm
    - John always calls when he hears the alarm, but sometimes confuses the telephone ringing with the alarm and calls then, too.
    - Mary likes rather loud music and sometimes misses the alarm altogether.

- Five beliefs
  - A: Alarm; B: Burglary; E: Earthquake; J: JohnCalls; M: MaryCalls
Example tasks

• Suppose Mary and John both call, what is the probability of a burglary?
• Suppose only John calls, what is the probability of a burglary?
• More generally, observing values for J (JohnCalls) and M (MaryCalls), what is the probability of a burglary?
A Simple Bayesian Network

Intuitive meaning of arrow from x to y: “x has direct influence on y”

Directed acyclic graph

Nodes are beliefs

Burglary

Earthquake

Alarm

JohnCalls

MaryCalls

causes

effects
A Simple Bayesian Network

**IMPORTANT NOTE!**

The arrows in the Bayesian network don’t necessarily follow the order of cause to effect. Often they do, but only because the network can become more intuitive this way.
Factorisation in this example

• Let us see how factorization of the joint distribution works for this example

• Although the required probabilities can be learned from data, currently we assume these as given (for easier understanding of the factorization)
Assigning Probabilities to Roots

<table>
<thead>
<tr>
<th></th>
<th>P(B)</th>
<th>P(not B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burglary</td>
<td>0.001</td>
<td>0.999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>P(E)</th>
<th>P(not E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earthquake</td>
<td>0.002</td>
<td>0.998</td>
</tr>
</tbody>
</table>
conditional probability table (CPT): each row contains the conditional probability of each node value for a conditioning case (a possible combination of values for the parent nodes).
Conditional Probability Table

**Redundant information!**

Burglary

<table>
<thead>
<tr>
<th>P(B)</th>
<th>P(not B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>0.999</td>
</tr>
</tbody>
</table>

Earthquake

<table>
<thead>
<tr>
<th>P(E)</th>
<th>P(not E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.002</td>
<td>0.998</td>
</tr>
</tbody>
</table>

JohnCalls

MaryCalls

| B   | E  | P(A|B,E) | P(not A | B,E) |
|-----|----|---------|----------|
| T   | T  | 0.95    | 0.05     |
| T   | F  | 0.94    | 0.06     |
| F   | T  | 0.29    | 0.71     |
| F   | F  | 0.001   | 0.999    |

Conditional probability table (CPT): each row contains the conditional probability of each node value for a conditioning case (a possible combination of values for the parent nodes).
After removing redundancy

Conditional probability table (CPT):
each row contains the conditional probability of each node value for a conditioning case (a possible combination of values for the parent nodes).
\[ P(\text{not } A \mid B, \text{not } E) = \ldots \]

A. 0.95
B. 0.94
C. 0.29
D. 0.06
E. 0.05
F. 0.001
G. None of the above
H. I don’t know

| B | E | P(A|B,E) |
|---|---|---------|
| T | T | 0.95    |
| T | F | 0.94    |
| F | T | 0.29    |
| F | F | 0.001   |

Response Counter
After removing redundancy

conditional probability table (CPT): each row contains the conditional probability of each node value for a conditioning case (a possible combination of values for the parent nodes).

| B | E | P(A|B,E) |
|---|---|---------|
| T | T | 0.95    |
| T | F | 0.94    |
| F | T | 0.29    |
| F | F | 0.001   |
Conditional Probability Tables

conditional probability table (CPT): each row contains the conditional probability of each node value for a conditioning case (a possible combination of values for the parent nodes).

<table>
<thead>
<tr>
<th></th>
<th>P(B)</th>
<th></th>
<th>P(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.001</td>
<td></td>
<td>0.002</td>
</tr>
</tbody>
</table>

| B | E | P(A|B,E) |
|---|---|---------|
| T | T | 0.95    |
| T | F | 0.94    |
| F | T | 0.29    |
| F | F | 0.001   |

| A | P(J|A) |
|---|------|
| T | 0.90  |
| F | 0.05  |

| A | P(M|A) |
|---|------|
| T | 0.70  |
| F | 0.01  |
Conditional Probability Tables

conditional probability table (CPT): each row contains the conditional probability of each node value for a conditioning case (a possible combination of values for the parent nodes).

Size of the conditional probability table (CPT) for a node with k parents: $2^k$

| B | E | P(A|B,E) |
|---|---|---------|
| T | T | 0.95    |
| T | F | 0.94    |
| F | T | 0.29    |
| F | F | 0.001   |

P(B) | 0.001

P(E) | 0.002

P(J|A) | 0.90

P(M|A) | 0.70

P(M|A) | 0.01
Dependencies and independencies

• **Presence of an arrow** between nodes in a Bayesian network **does not mean** that the connected nodes are **dependent** (can still be independent)

• **Lack of an arrow** **does not mean** that the nodes are independent (can be dependent)

• **Lack of an arrow** means **conditional independence, given all parents** of one or both nodes

• **Presence of an arrow** **means** that there is **no information** about conditional independence
Which independencies are encoded by this Bayesian network?

- Burglary
- Earthquake
- JohnCalls
- MaryCalls

Diagram:

- Burglary → Alarm
- Earthquake → Alarm
- JohnCalls → Alarm
- MaryCalls → Alarm
Which independencies are encoded by this Bayesian network?

Lack of each of these red arrows encodes some (conditional) independence
Conditional independencies

Lack of this arrow means: JohnCalls is independent of Burglary given Alarm or ¬Alarm
Lack of this arrow means: MaryCalls is independent of Earthquake given Alarm or ¬Alarm
Conditional independencies

Lack of these arrows means:
MaryCalls is independent of JohnCalls given Alarm or ¬Alarm
Lack of these arrows means:
Burglary is independent of Earthquake
(unconditionally)
The following two Bayesian networks encode the same independence structure

A. Yes
B. No
C. I don’t know
The following two Bayesian networks encode the same independence structure.

A. Yes
B. No
C. I don’t know

Response Counter

1

A
B
C

A
B
C

Yes
No
I don’t know
The following two Bayesian networks encode the same independence structure

A. Yes
B. Yes
C. I don’t know
What is the difference?

- A and C are (unconditionally) independent
- A and C are independent conditional to B
Calculation of Joint Probability

\[ P(J \land M \land A \land \lnot B \land \lnot E) = ?? \]

| B | E | P(A|B,E) |
|---|---|---------|
| T | T | 0.95    |
| T | F | 0.94    |
| F | T | 0.29    |
| F | F | 0.001   |

| A | P(J|A) |
|---|------|
| T | 0.90 |
| F | 0.05 |

| A | P(M|A) |
|---|------|
| T | 0.70 |
| F | 0.01 |
- \(P(J \land M \land A \land \neg B \land \neg E)\)
  \[= P(J \land M | A, \neg B, \neg E) \times P(A \land \neg B \land \neg E)\]
  \[= P(J | A, \neg B, \neg E) \times P(M | A, \neg B, \neg E) \times P(A \land \neg B \land \neg E)\]
  \((\text{because } J \text{ and } M \text{ are independent given } A)\)

- \(P(J | A, \neg B, \neg E) = P(J | A)\)
  \((J \text{ and } \neg B \land \neg E \text{ are independent given } A)\)

- \(P(A \land \neg B \land \neg E) = P(A | \neg B, \neg E) \times P(\neg B) \times P(\neg E)\)
  \((\neg B \text{ and } \neg E \text{ are independent})\)

- \(P(J \land M \land A \land \neg B \land \neg E) = P(J | A)P(M | A)P(A | \neg B, \neg E)P(\neg B)P(\neg E)\)
Calculation of Joint Probability

\[ P(J \land M \land A \land \neg B \land \neg E) = P(J|A)P(M|A)P(A|\neg B, \neg E)P(\neg B)P(\neg E) = 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062 \]
Calculation of Joint Probability

General formula:
\[ P(x_1, x_2, ..., x_n) = \prod_{i=1,...,n} P(x_i|\text{Parents}(X_i)) \]

P(J \land M \land A \land \neg B \land \neg E)
= P(J|A)P(M|A)P(A|\neg B, \neg E)P(\neg B)P(\neg E)
= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998
= 0.00062
Bayesian Network – any directed acyclic graph (DAG)

\[ p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \]
\[ p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5) \]

General Factorization

\[ p(x) = \prod_{k=1}^{K} p(x_k|\text{pa}_k) \]

\text{pa}_k - all parent nodes of } x_k
Example
How is Naive Bayes represented as Bayesian network?
Size of probability table of NB

- Suppose that the label and all m features are binary
- Then the table for joint probability distribution $P(X_1, \ldots, X_m, Y)$ has $2^{m+1}$ rows
- If the Naïve Bayes assumption holds we can represent it by $P(Y) \prod_{i=1}^{m} P(X_i \mid Y)$
- This has $2 + 4m$ rows only:
  - 2 rows for $P(Y)$ for $Y=t$ and $Y=f$ (true and false)
  - 4 rows for $P(Xi\mid Y)$ for $(X_i,Y)=$(t,t), (t,f), (f,t), (f,f)
Size of probability table of NB

• Since all variables are binary we need less rows, as \( P(A_i = \text{false}) = 1 - P(A_i = \text{true}) \)

• Then the table for joint probability distribution \( P(X_1, \ldots, X_m, Y) \) has \( 2^{m+1} - 1 \) rows

• If the Naïve Bayes assumption holds we can represent it by \( P(Y) \prod_{i=1}^{m} P(X_i \mid Y) \)

• This has \( 1 + 2m \) rows only:
  – 1 row for \( P(Y=t) \)
  – 2 rows for \( P(X_i \mid Y) \) for \((X_i, Y)=(t,t), (t,f)\)
Inference in Bayesian networks

- Set E of evidence variables that are observed with new probability distribution, e.g., \{JohnCalls, MaryCalls\}
- Query variable X, e.g., Burglary, for which we would like to know the posterior probability distribution \(P(X|E)\)

| J | M | P(B|...) |
|---|---|----------|
| T | F | ?        |
| F | F | ?        |

Distribution conditional to the observations made
Inference in Bayesian networks

- Set \( E \) of **evidence variables** that are observed with new probability distribution, e.g., \( \{\text{JohnCalls, MaryCalls}\} \)
- **Query variable** \( X \), e.g., Burglary, for which we would like to know the posterior probability distribution \( P(X|E) \)

In machine learning terms:
- Evidence variables = features
- Query variable = target variable
Lecture 11 – Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
  • **Inference example**
  • Bayesian networks and learning
  • Markov model
  • Hidden markov model (HMM)
  • Markov random fields
  • Directed vs undirected graphical models
  • Implementation
Another story

- Dr. Watson goes out in the morning and sees that the grass is wet. He wonders if it rained at night or if he had left the sprinkler on last evening. He checks that the sprinkler is off and concludes that it must have rained.
The involved variables

• Suppose we have 4 binary variables:
  – Cloudy (is it cloudy or not)
  – Sprinkler (was the sprinkler on at night or not)
  – Rain (did it rain at night or not)
  – WetGrass (is the grass wet or not)

• Which variables depend on which?
Factorised joint probability distribution

- Variables: *Cloudy, Sprinkler, Rain, WetGrass*

Note that we have omitted the complementary probabilities such as $P(\text{not cloudy})$ as this can be calculated as $1 - P(\text{cloudy})$. 
From joint distribution to predictions

• Once we know the joint distribution, we can calculate any probabilities under any conditioning
  – E.g. \( P(Y|X_1, X_2, \ldots, X_m) \)

• Such calculations are called inference in probabilistic modelling
Inference example

- Given that the grass is wet, what is the probability that it has rained?
Inference example

• Given that the grass is wet, what is the probability that it has rained?

\[ P(r \mid w) \]

\[ P(r, w) = \sum_{C=c, S=s} P(c, s, r, w) \]
Inference example

- Given that the grass is wet, what is the probability that it has rained?

\[
P(r \mid w) = \frac{P(r, w)}{P(w)} = \frac{P(r, w)}{P(r, w) + P(\neg r, w)}
\]

\[
P(r, w) = \sum_{C=c, S=s} P(c, s, r, w)
\]

\[
= \sum_{C=c, S=s} P(c)P(s \mid c)P(r \mid c)P(w \mid r, s)
\]

\[
= \sum_{C=c} P(c)P(r \mid c)\sum_{S=s} P(w \mid r, s)P(s \mid c)
\]
Bayesian network inference: Big picture

• **Exact inference is intractable**
  - There exist techniques to speed up computations, but worst-case complexity is still exponential except in some classes of networks (polytrees)

• **Approximate inference**
  - Sampling, variational methods, message passing / belief propagation...
Lecture 11 –
Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
✓ Inference example

• **Bayesian networks and learning**
• Markov model
• Hidden markov model (HMM)
• Markov random fields
• Directed vs undirected graphical models
• Implementation
Inference vs Parameter learning

- **Inference problem**: given values of evidence variables \( E = e \), answer questions about query variables \( X \) using the posterior \( P(X \mid E = e) \)

- **Learning problem**: estimate the parameters of the probabilistic model \( P(X \mid E) \) given a *training sample* \( \{(x_1, e_1), \ldots, (x_n, e_n)\} \)
Parameter learning

- Suppose we know the network structure (but not the parameters), and have a training set of complete observations

\[
\begin{array}{c|c|c|c|c}
\text{Sample} & \text{C} & \text{S} & \text{R} & \text{W} \\
1 & T & F & T & T \\
2 & F & T & F & T \\
3 & T & F & F & F \\
4 & T & T & T & T \\
5 & F & T & F & T \\
6 & T & F & T & F \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\end{array}
\]
Parameter learning

• Suppose we know the network structure (but not the parameters), and have a training set of complete observations
  
  - \( P(X \mid \text{Parents}(X)) \) is given by the observed frequencies of the different values of \( X \) for each combination of parent values
Parameter learning

- Incomplete observations

- **Expectation maximization (EM) algorithm** for dealing with missing data – not covered in the course

<table>
<thead>
<tr>
<th>Sample</th>
<th>C</th>
<th>S</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>?</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td>?</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Parameter learning

- What if the network structure is unknown?
  - *Structure learning* algorithms exist, but they are pretty complicated...

<table>
<thead>
<tr>
<th>Sample</th>
<th>C</th>
<th>S</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

...  ...  ...  .....  ...

Training set
Lecture 11 –
Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
✓ Inference example
✓ Bayesian networks and learning
  • **Markov model**
  • Hidden markov model (HMM)
  • Markov random fields
  • Directed vs undirected graphical models
  • Implementation
Markov model

- A particular kind of Bayesian network

- All variables are observed

- Suitable for modeling dependencies within sequences

\[ P(S_n \mid S_1, S_2, \ldots, S_{n-1}) = P(S_n \mid S_{n-1}) \]

*Markov property*

\[ P(S_1, S_2, S_3, \ldots, S_n) = P(S_1) P(S_2 \mid S_1) \ldots P(S_n \mid S_{n-1}) \]
Markov model

- **States**: letters in English words
- **Transitions**: which letter follows which

\[
P(S_n = y \mid S_{n-1} = x) = \frac{\text{frequency of } xy}{\text{frequency of } x}
\]

Here, \( S_1 = M \), \( S_2 = R \), \( S_3 = \text{<space>} \), \( S_4 = S \), and \( S_5 = H \).
Markov model

- States: **triplets** of letters
- Transitions: which (overlapping) triplet follows which

\[
P(S_n = \text{xyz} | S_{n-1} = \text{wxy}) = \frac{P(\text{wxyz})}{P(\text{wxy})}
\]

\[
P(S_1 = \text{MR}<\text{space}>)
\]
\[
P(S_2 = \text{R}<\text{space}>S)
\]
\[
P(S_3 = <\text{space}>\text{SH})
\]
\[
P(S_4 = \text{SHE})
\]
\[
P(S_5 = \text{HER})
\]

MR SHERLOCK HOLMES WHO WAS USUALLY VERY LATE IN THE MORNINGS
SAVE UPON THOSE NOT INFREQUENT OCCASIONS WHEN HE WAS UP ALL....

THERE THE YOU SOME OF FEELING WILL PREOCCUPATIENCE CRESASON LITTLED
MASTIFF HENRY MALIGNATIVE LL HAVE MAY UPON IMPRESENT WARNESTLY
Markov model

• States: **word** pairs

• Text from: [http://www.gutenberg.org/etext/1105](http://www.gutenberg.org/etext/1105)

When thou thy sins enclose!
That tongue that tells the story of thy love
Ay fill it full with feasting on your sight
Book both my wilfulness and errors down
And on just proof surmise accumulate
Bring me within the level of your eyes
And in mine own when I of you beauteous and lovely

youth

When that churl death my bones with dust
shall cover
And shalt by fortune once more re-survey
These poor rude lines of life thou art forced
to break

a twofold truth

Hers by thy deeds

Then churls their thoughts (although their eyes were kind)
To thy fair appearance lies
To side this title is impanelled
A quest of thoughts all tenants to the sober west
As those gold candles fixed in heaven's air
Let them say more that like of hearsay well
I will drink
Potions of eisel 'gainst my strong infection
No bitterness that I was false of heart
Though absence seemed my flame to qualify
As easy might I not free
Lecture 11 –
Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
✓ Inference example
✓ Bayesian networks and learning
✓ Markov model
  • **Hidden markov model** (HMM)
  • Markov random fields
  • Directed vs undirected graphical models
  • Implementation
Example: occasionally corrupt casino

- Casino has one fair and one loaded dice
- Fair:
  - 1/6 probability for each result
- Loaded:
  - Result 6 is more likely, others uniform
- Player does not know whether the fair or loaded dice is used
- Every now and then casino swaps the dice
- Goal for player: guess which dice is used
Example: The Occasionally Corrupt Casino

\[ P \]

\[ P \]

\[ \bullet \] \[ \bullet \] \[ \bullet \] \[ \text{dice} \] \[ \text{dice} \] \[ \text{dice} \] \[ \text{dice} \] \[ \text{dice} \] \[ \text{dice} \] \[ \text{dice} \]
Hidden Markov model

- HMM = probabilistic observation of a Markov chain
- Another special kind of Bayesian network

\[ S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5 \rightarrow S_6 \rightarrow S_7 \rightarrow S_8 \rightarrow \ldots \]

\[ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \]

\[ Y_1 \rightarrow Y_2 \rightarrow Y_3 \rightarrow Y_4 \rightarrow Y_5 \rightarrow Y_6 \rightarrow Y_7 \rightarrow Y_8 \rightarrow \ldots \]

- \( S_i \) form a Markov chain as before, but states are unobserved (e.g. fair vs loaded dice)
- Instead, \( y_i \) (dependent on \( S_i \)) are observed (dice results)
- Generative viewpoint: state \( S_i \) “emits” symbol \( y_i \)
- \( y_i \) do not form a Markov chain (= do not satisfy Markov property)
  They exhibit more complex (long-range) dependencies
Example: HMM
Inference in HMMs

So HMMs can describe complex (temporal, spatial) relationships in data. But how can we use the model?

- A number of (efficient) *inference algorithms* exist for HMMs:
  - Viterbi algorithm: most likely state sequence, given observables
  - Forward algorithm: likelihood of model given observables
  - Backward algorithm: together with Forward, allows computation of posterior probabilities
  - Baum-Welch algorithm: parameter estimation given observables
The Most Likely State Sequence

Find the mode of

$$P(S_1, \ldots, S_N | y_1, \ldots, y_N) = \frac{P(S_1, \ldots, S_N, y_1, \ldots, y_N)}{P(y_1, \ldots, y_N)}$$

$$\propto P(S_1, \ldots, S_N, y_1, \ldots, y_N)$$

$$S_n \in \mathcal{H} \quad \Rightarrow \quad (S_1, \ldots, S_N) : |\mathcal{H}|^N \text{ terms.}$$
Conditional Independence in HMMs

\[
P(y_t | y_1, \ldots, y_{t-1}, y_{t+1}, \ldots, y_N, S_1, \ldots, S_N) = P(y_t | S_t)
\]

\[
P(S_{t+1} | S_1, \ldots, S_t, y_1, \ldots, y_t) = P(S_{t+1} | S_t)
\]
Factorisation in HMMs

\[ P(y_1, \ldots, y_N, S_1, \ldots, S_N) = \prod_{t=1}^{N} P(y_t|S_t) \prod_{t=2}^{N} P(S_t|S_{t-1}) P(S_1) \]

- Emission probabilities
- Transition probabilities
- Starting probabilities
Probabilistic graphical models

• Bayesian networks (including HMMs) are an example of **directed graphical models**

• There exist also **undirected graphical models**
  – Markov random fields are a main example
Lecture 11 – Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
✓ Inference example
✓ Bayesian networks and learning
✓ Markov model
✓ Hidden markov model (HMM)
  • Markov random fields
  • Directed vs undirected graphical models
  • Implementation
Illustration: Image De-Noising (1)
Markov Chains

\[
\text{Pr}(X_{t+1} = y | X_t = x, X_{t-1}, X_{t-2}, \ldots) = \text{Pr}(X_{t+1} = y | X_t = x) = K(x, y)
\]

\[
P(x_0, x_1, \ldots, x_t) = P(x_0)P(x_1 | x_0) \ldots P(x_t | x_{t-1})
\]

Pr(future | present, past) = Pr(future | present)
future \perp past | present

**Markov property**: conditional independence
limited dependence
Makes modeling and learning possible
Markov Random Fields

Markov Property

\[ P(I_{x,y} | I_{-(x,y)}) = P(I_{x,y} | I_{\mathcal{N}(x,y)}) \]

\( -(x, y) \) all the other nodes

\[ \mathcal{N}(x, y) = \{(x - 1, y), (x + 1, y), (x, y - 1), (x, y + 1)\} \]

Nearest neighborhood, first order neighborhood

From Slides by S. Seitz - University of Washington
Markov Random Fields

Can be generalized to any undirected graphs (nodes, edges)
Neighborhood system: each node is connected to its neighbors
neighbors are reciprocal
Markov property: each node only depends on its neighbors
Markov Random Fields

\[ P(I_{x,y} \mid I_{-(x,y)}) = P(I_{x,y} \mid I_{N(x,y)}) \]

What is \( P(I) \)?

\[ I = (I_{x,y}, \forall(x, y)) \]
Illustration: Image De-Noising (1)

Original Image

Noisy Image
Illustration: Image De-Noising (2)

\[ E(x, y) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i \]

\[ p(x, y) = \frac{1}{Z} \exp\{-E(x, y)\} \]
Illustration: Image De-Noising (3)

Noisy Image

Restored Image

(ICM – Iterated conditional models)
Illustration: Image De-Noising (4)

Restored Image
(ICM – Iterated conditional models)

Restored Image (Graph cuts)
This is an example of structured output prediction: task where the target variable is more than just one real or discrete value – in image de-noising the target value is a matrix of pixels.
Lecture 11 –
Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
✓ Inference example
✓ Bayesian networks and learning
✓ Markov model
✓ Hidden markov model (HMM)
✓ Markov random fields

• Directed vs undirected graphical models
• Implementation
Converting Directed to Undirected Graphs

\[ p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2) \cdots p(x_N|x_{N-1}) \]

\[ p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N) \]
Directed vs. Undirected Graphs (1)

\[ P = \text{all possible conditional independence structures over a fixed set of variables} \]
\[ D = \text{those structures that can be represented by directed graphical models} \]
\[ \text{(Bayesian networks)} \]
\[ U = \text{those structures that can be represented by undirected graphical models} \]
\[ \text{(Markov random fields)} \]
Directed vs. Undirected Graphs (2)

A Bayesian network (=directed graphical model)
that cannot be represented as Markov random field

\[ A \perp B \mid \emptyset \]

\[ A \not\perp B \mid C \]

A Markov random field (=undirected graphical model)
that cannot be represented as a Bayesian network

\[ A \not\perp B \mid \emptyset \]

\[ A \perp B \mid C \cup D \]

\[ C \perp D \mid A \cup B \]
Lecture 11 – Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
✓ Inference example
✓ Bayesian networks and learning
✓ Markov model
✓ Hidden markov model (HMM)
✓ Markov random fields
✓ Directed vs undirected graphical models

• Implementation
Implementation of PGMs

• PGMs (probabilistic graphical models) are implemented either in a specific form:
  – Naïve Bayes
  – HMM

• Or alternatively, using a probabilistic programming language, such as:
  – Edward (http://edwardlib.org)
  – Infer.net (http://infernet.azurewebsites.net/)
**WetGrass example in Edward**

```
rain = edm.Bernoulli(probs=0.2)
p_sprinkler = tf.where(tf.cast(rain, tf.bool), 0.01, 0.4)
sprinkler = edm.Bernoulli(probs=p_sprinkler)
p_grass_wet = tf.where(tf.cast(rain, tf.bool),
    tf.where(tf.cast(sprinkler, tf.bool), 0.99, 0.8),
    tf.where(tf.cast(sprinkler, tf.bool), 0.9,
              0.00000001))
grass_wet = edm.Bernoulli(probs=p_grass_wet)

# Inference
q_rain =
edm.Bernoulli(probs=tf.nn.sigmoid(tf.Variable(tf.random_normal([]))))
ed.get_session()
inf = edi.KLpq({rain: q_rain}, data={grass_wet: tf.constant(1, dtype=tf.int32)})
inf.run(n_samples=50)
print(q_rain.probs.eval())
```

See full Python notebook at:
Lecture 11 – Probabilistic graphical models

✓ Motivation
✓ Factorisation of probability distributions
✓ Bayesian network example
✓ Inference example
✓ Bayesian networks and learning
✓ Markov model
✓ Hidden markov model (HMM)
✓ Markov random fields
✓ Directed vs undirected graphical models
✓ Implementation