4F10: Deep Learning and Structured Data

Graphical Models and Conditional Independence

Mark Gales
Slides - José Miguel Hernández–Lobato
Department of Engineering
University of Cambridge

Michaelmas Term

Basics of Probability

Everything needed follows from just two rules:

Sum rule:

$$p(X) = \sum_Y p(X, Y)\,.$$

Product rule:

$$p(X, Y) = p(Y|X)p(X) = p(X|Y)p(Y)\,.$$

They can be combined to obtain **Bayes' rule:**

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} = \frac{p(X|Y)p(Y)}{\sum_Y p(X, Y)}\,.$$

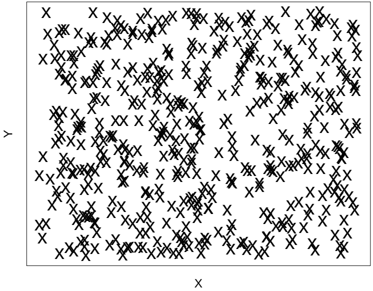
Independence of X and Y ($X \perp Y$): $p(X, Y) = p(X)p(Y)$.

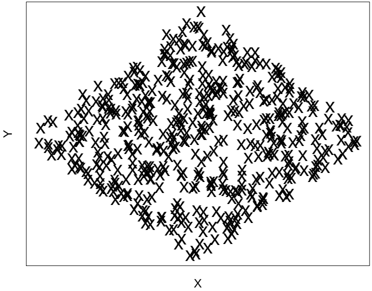
Conditional independence of X and Y given Z: $(X \perp Y|Z)$
$p(X, Y|Z) = p(X|Z)p(Y|Z)$.

Independence Examples

Recall that $X \perp Y$ implies $p(X, Y) = p(X)p(Y)$ and assume that $(X, Y) \in [0, 1] \times [0, 1]$.

$X \perp Y$?



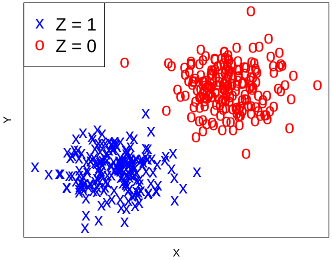


	Yes	No
	$p(X, Y) = 1$	$p(X, Y) = 2\mathbb{I}\left[\left\|X - \frac{1}{2}\right\| + \left\|Y - \frac{1}{2}\right\| < \frac{1}{2}\right]$
	Distribution is simple	**Distribution is more complicated**

Conditional Independence Examples

Recall that $X \perp Y | Z$ implies $p(X, Y|Z) = p(X|Z)p(Y|Z)$. Let $(X, Y) \in \mathbb{R}^2$ and $Z \in 0, 1$.

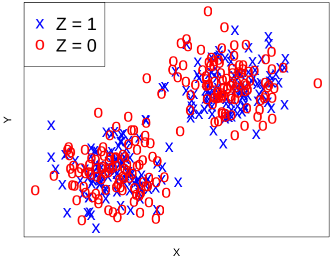
$X \perp Y | Z$?



Yes

$$p(X, Y, Z) = \frac{Z}{2}\mathcal{N}(X|2, 1)\mathcal{N}(Y|2, 1) + \frac{1-Z}{2}\mathcal{N}(X|-2, 1)\mathcal{N}(Y|-2, 1)$$

Distribution is simple



No

$$p(X, Y, Z) = \frac{Z}{2}\left[\frac{1}{2}\mathcal{N}(X|2, 1)\mathcal{N}(Y|2, 1) + \frac{1}{2}\mathcal{N}(X|-2, 1)\mathcal{N}(Y|-2, 1)\right] + \frac{1-Z}{2}\left[\frac{1}{2}\mathcal{N}(X|2, 1)\mathcal{N}(Y|2, 1) + \frac{1}{2}\mathcal{N}(X|-2, 1)\mathcal{N}(Y|-2, 1)\right].$$

Distribution is more complicated

Motivation for Conditional Independencies

A distribution satisfying conditional independencies is simpler and can be represented **more compactly**.

Example with binary variables: $p(A, B, C, D) = p(A|C)p(B|C)p(C)p(D)$.

What independencies occur in this distribution? $C \perp D$, $D \perp A$, $D \perp B$ and $A \perp B | C$.

p(A,B,C,D)

A B C D	prob
0 0 0 0	0.000375
0 0 0 1	0.001125
0 0 1 0	0.008750
0 0 1 1	0.026250
0 1 0 0	0.003375
0 1 0 1	0.010125
0 1 1 0	0.026250
0 1 1 1	0.078750
1 0 0 0	0.007125
1 0 0 1	0.021375
1 0 1 0	0.035000
1 0 1 1	0.105000
1 1 0 0	0.064125
1 1 0 1	0.192375
1 1 1 0	0.105000
1 1 1 1	0.315000-

=

p(A|C)

A C	prob
0 0	0.05
0 1	0.20
1 0	0.95
1 1	0.80

x

p(B|C)

B C	prob
0 0	0.10
0 1	0.25
1 0	0.90
1 1	0.75

x

p(C)

C	prob
0	0.30
1	0.70

x

p(D)

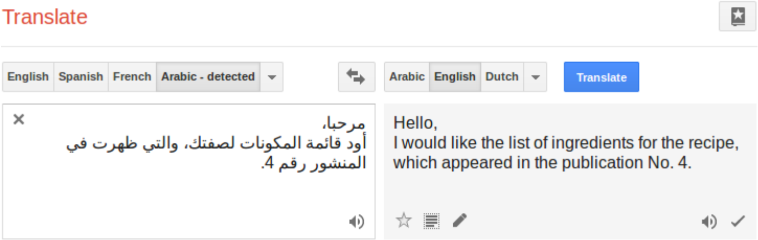
D	prob
0	0.25
1	0.75

A probability table of size 16 is represented using smaller tables (factors) of size 4, 4, 2 and 2. In high-dimensional probability distributions the gains would be much higher!

Working with distributions that factorize in terms of simple factors and the introduction of conditional independence assumptions is equivalent.

Additional Motivation: Language Model Example

A language model is a probability distribution $p(W_1, \ldots, W_T)$ over sequences of words, e.g. of length T. Useful, for example, in automatic language translation.



How to specify and fit $p(W_1, \ldots, W_T)$ to data? Possible approach: use **product rule** and fit the individual factors by **maximum likelihood**:

$$p(W_1, \ldots, W_T) = p(W_1)p(W_2|W_1)p(W_3|W_2, W_1) \cdots p(W_T|W_1, \ldots, W_{T-1}) \,.$$

Learning requires computing frequencies of **long sub-sequences** in the training data.

Most frequencies for long word sub-sequences will be zero in the training data!

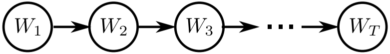
Solution: simplify factors by introducing **conditional independencies.**

Markov Models

First order Markov:

parameters tied

$$p(W_1, W_2, \ldots, W_T) = p(W_1)p(W_2|W_1)p(W_3|W_2)\cdots p(W_T|W_{T-1})$$



Markov model = product rule + conditional independence

$$W_{t+1} \perp W_1, \ldots, W_{t-1} | W_t$$

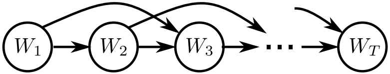
given present

Future

independent of past

Second order Markov:

$$p(W_1, W_2, \ldots, W_T) = p(W_1)p(W_2|W_1)p(W_3|W_2, W_1)\cdots p(W_T|W_{T-1}, W_{T-2})$$



Learning done by computing frequencies of only up to 2 or 3 consecutive words.

The Big Picture

Modeling data requires us to specify a **high-dimensional distribution** $p(X_1, \ldots, X_d)$.

However, working with fully flexible joint distributions is **intractable**! :-(

Instead, we can work with **structured distributions**, where $p(X_1, \ldots, X_d)$ is written as a **product of simpler factors** evaluated only on a **subset** of $X_1, \ldots, X_d$:-)

By using simple factors, the random variables interact directly only with few others: the factorization introduces **conditional independencies**. This

- Results in a **compact** representation of the distribution.
- Simplifies the fit of the distribution parameters to data (**learning**).
- Allows us to **sum out** variables efficiently (see slide 11), e.g. to compute the **normalization constant** in Bayes rule.

Graphical Models: factorizations can be encoded one-to-one as **graphs** in which

- Nodes are random variables.
- Edges connect variables for which **no conditional independencies exist**.

Bayesian Networks (Directed Graphical Models)

Markov models are a type of graphical model called **Bayesian networks**.

A Bayesian network $\mathcal{G}$ is a **directed acyclic graph** whose nodes are random variables $X_1, \ldots, X_d$.

Let $\mathrm{PA}_{X_i}^{\mathcal{G}}$ be the **parents** of X_i in $\mathcal{G}$.
The network is annotated with the **conditional distributions** $p(X_i | \mathrm{PA}_{X_i}^{\mathcal{G}})$.

Factorization:

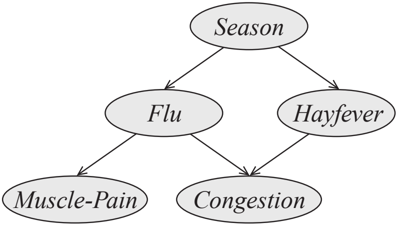
$\mathcal{G}$ encodes the factorization $p(X_1, \ldots, X_d) = \prod_{i=1}^{d} p(X_i | \mathrm{PA}_{X_i}^{\mathcal{G}})$.

Conditional Independencies (CI):

Let $\mathrm{ND}_{X_i}^{\mathcal{G}}$ be the variables in $\mathcal{G}$ which are **non-descendants** of X_i in $\mathcal{G}$.
$\mathcal{G}$ encodes the conditional independencies $(X_i \perp \mathrm{ND}_{X_i}^{\mathcal{G}} | \mathrm{PA}_{X_i}^{\mathcal{G}})$, $i = 1, \ldots, d$.

Example of Bayesian Network

Graph:



Factorization:

$p(S, F, H, M, C) = p(S)p(F|S)p(H|S)p(C|F, H)p(M|F)$

Conditional Independencies:

$(F \perp H|S)$, $(C \perp S|F, H)$, $(M \perp H|F)$, $(M \perp C|F)$, ...

Figure source [Koller et al. 2009].

Efficient Marginalization in Graphical Models

Given the Bayesian network (BN) $A \to B \to C \to D$ we want to compute $p(d)$.

Using the **factorization** given by the BN and the **sum rule** of probability theory, we have $p(d) = \sum_a \sum_b \sum_c p(d|c)p(c|b)p(b|a)p(a)$. When variables are discrete, taking n different values each, the cost of this operation is $\mathcal{O}(n^4)$ because we first construct a table of size n^4 and then sum its entries.

Reordering operations results in $p(d) = \sum_c p(d|c) \sum_b p(c|b) \sum_a p(b|a)p(a)$. In this latter case, the cost is $\mathcal{O}(n^2)$, given by the **largest factor** or probability table generated during this alternative summation process.

Selecting a specific **order** of computations can produce **large savings**!

Approach

The BN expresses the joint distribution as a **product of factors** which depend only on a **small number of variables**.

Exploit the BN factorization, so that we avoid generating very large factors (**probability tables**) during the summation process.

Motivation for Undirected Graphical Models

In some cases, having to choose a **direction** for the edges is rather awkward.

Recall that a multivariate Gaussian with mean $\mathbf{0}$ and cov. matrix $\mathbf{\Sigma}$ is given by

$$p(X_1, \ldots, X_d) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}|}} \exp\left\{(X_1, \ldots, X_d)^\mathsf{T} \mathbf{\Sigma}^{-1} (X_1, \ldots, X_d)\right\} .$$

Let us assume that the precision matrix $\mathbf{\Lambda} = \mathbf{\Sigma}^{-1}$ is **sparse** with $\lambda_{i,j} \neq 0$ if $(i,j) \in \mathcal{E}$. Then we obtain the following factorization:

$$p(X_1, \ldots, X_d) \propto \exp\left\{-\frac{1}{2} \sum_{(i,j)\in\mathcal{E}} \lambda_{i,j} X_i X_j\right\} = \prod_{(i,j)\in\mathcal{E}} \exp\left\{-\frac{1}{2}\lambda_{i,j} X_i X_j\right\} .$$

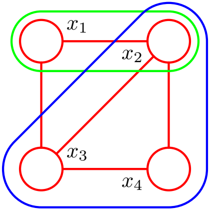
In this case, the factors are **symmetric** in X_i and X_j: using an **undirected graph** with no edge orientations seems a better option than a Bayesian network.

Markov Networks (Undirected Graphical Models)

A Markov Network (MN) is an **undirected** graph $\mathcal{G}$ whose nodes are the r.v. $X_1, \ldots, X_d$.

It is annotated with the **positive potential functions** $\phi_1(\mathbf{D}_1), \ldots, \phi_k(\mathbf{D}_k)$, where $\mathbf{D}_1, \ldots, \mathbf{D}_k$ are sets of variables, each forming a **clique** of $\mathcal{G}$.

A **Clique** is a fully connected subset of nodes.



Clique examples.

Factorization:

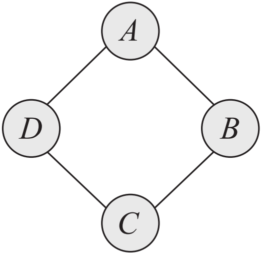
$\mathcal{G}$ encodes the factorization $p(X_1, \ldots, X_d) = Z^{-1} \prod_{i=1}^{k} \phi_i(\mathbf{D}_i)$, where Z is the **partition function** or normalization constant: $Z = \sum_{X_1,\ldots,X_d} \prod_{i=1}^{k} \phi_i(\mathbf{D}_i)$.

Conditional Independencies (CIs):

$\mathcal{G}$ encodes the CIs $(A \perp B | C)$ for any sets of nodes A, B and C such that C **separates** A **from** B **in** $\mathcal{G}$ (C blocks all paths in $\mathcal{G}$ between A and B).

Example of Markov Network

Graph:



Factorization:

$$p(A, B, C, D) = \frac{1}{Z}\phi_1(A, B)\phi_2(B, C)\phi_3(C, D)\phi_4(A, D)$$

Conditional Independencies:

$(A \perp C | B, D)$, $(B \perp D | A, C)$

Figure source [Koller et al. 2009].

Markov Network Example: Potts Model

Useful for image segmentation.

Let $x_1, \ldots, x_n \in \{1, \ldots, C\}$,

$$p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{i \sim j} \phi_{ij}(x_i, x_j)\,,$$

where

$$\log \phi_{ij}(x_i, x_j) = \begin{cases} \beta > 0 & \text{if} \quad x_i = x_j \\ 0 & \text{otherwise} \end{cases},$$

Segmentation example

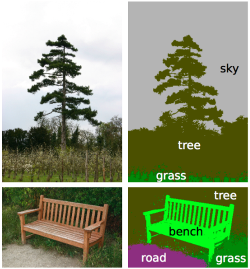


Figure: Krähenbühl and Koltun

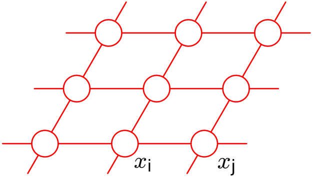
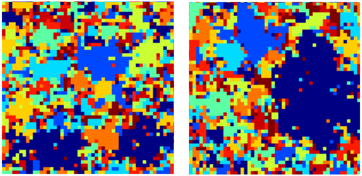


Figure: C. Bishop.



Samples from Potts model, figure: Erik Sudderth.

Summary

Probability theory is a natural and principled tool for dealing with uncertainty.

In practice, we have to work with compact and structured probability distributions.

Graphical models encode such compact distributions by specifying several CIs which also correspond to a factorization of the joint probability distribution.

Bayesian and Markov networks are two types of graphical models which express different types of CIs.

Marginalization may require to sum an exponentially large number of terms.

We can avoid that by exploiting a factorization and caching intermediate results.