Lecture 11 - *Probabilistic Graphical Models*

OBJECTIVE: Probabilistic graphical models (aka Bayes nets) combine probability theory and graphs in order to represent large domains of random variables. We will tackle two tasks: inference and learning. In inference, we assume we have the conditional probability tables and focus on estimating the probability of a group of variables given the other variables. In learning, we compute the conditional probability tables from data.

Textbook: Missing section.

Let $\mathbf{x}$ denote two random variables $\mathbf{x} = (x_1, x_2)$, each taking 3 possible values. That is, $x_i \in E = \{1, 2, 3\}$. We can represent the marginal, conditional and joint distributions with the following tables:

⋆

$\star$

$$x_i \in E = \{1, \ldots, r\} \qquad \text{for } i = 1 : n$$

$$size(\text{ joint probability table }) =$$

We can exploit conditional independencies and graph theory to replace large tables by a group of smaller tables.

A **directed graph** is a pair $G = (x, e)$ with nodes $x_{1:n}$ and directed edges $e = \{(x_i, x_j) : i \neq j\}$. The nodes will correspond to r.v.s and the edges to conditional probabilities. We assume that G is acyclic.

$\star$

In general:

$$p(x_{1:n}) = \prod_{i=1}^{n} p(x_i | parents(x_i))$$

The size of each table is r^{m_i+1}, where m_i is the number of parents of node x_i.

Graphical models are often used as expert systems:

★

Conditional Independence Statements

★

Inference in DAGs

Suppose we are interested in computing $P(x_1|x_6 = 1)$ in the following model:

⋆

$p(x_1|x_6 = 1) =$

⋆

The idea of replacing sums of products ($ac+ab$) by products of sums ($a(b+c)$) is at the heart of most inference algorithms. For exact inference, in Gaussian and discrete networks of reasonable size, we use the **junction tree algorithm**. This algorithm involves two steps:

1. Converting the directed graph to an undirected graph called the junction tree.
2. Running belief propagation. That is, replace sums of products by products of sums.

Dynamic Bayesian Networks and HMMs

⋆

A General Framework

As hinted by the previous example, many algorithms can be placed in the framework of graphical models.

⋆

Learning in Graphical Models

We consider two paradigms: frequentist and Bayesian.

Frequentist Learning

It assumes that there is a true model (say a parametric model with parameters θ_0). The estimate is denoted $\widehat{\theta}$. It can be found by maximising the **likelihood**:

⋆

$$\widehat{\theta} = \arg\max_{\theta} \quad p(x_{1:n}|\theta)$$

For **identical and independent distributed** (i.i.d.) data:

$$p(x_{1:n}|\theta) =$$

$$\mathcal{L}(\theta) = \log p(x_{1:n}|\theta) =$$

⋆ Let $x_{1:n}$, with $x_i \in \{0,1\}$, be i.i.d. Bernoulli:

$$p(x_{1:n}|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

With $m \triangleq \sum x_i$, we have

$$\mathcal{L}(\theta) =$$

Differentiating, we get

We can now go back to graphical models and learn the **conditional probability tables** (CPTs):

⋆ Let the DAG be

And assume we have collected the data:

c	r	**g**
0	0	0
0	0	0
1	0	1
1	1	1
1	1	1

⋆

The conditional probabilities are:

$$\begin{aligned} p(c|\gamma) &\propto \\ p(r|\alpha_1, c=0) &\propto \\ p(r|\alpha_2, c=1) &\propto \\ p(g|\beta_1, c=0) &\propto \\ p(g|\beta_2, c=1) &\propto \end{aligned}$$

and hence, the ML estimates are:

$$\begin{aligned} \gamma &= \\ \alpha_1 &= \\ \alpha_2 &= \\ \beta_1 &= \\ \beta_2 &= \end{aligned}$$

Now we can carry out inference to answer queries like $p(g|r=1)$.

⋆

$p(g=0|r=1) =$

Frequentist Model Selection

How about using another model to represent the same data?

⋆

$$
\begin{aligned}
p(c|\gamma) &\propto \\
p(r|\alpha_1, c=0) &\propto \\
p(r|\alpha_2, c=1) &\propto \\
p(g) &\propto
\end{aligned}
$$

$$
\begin{aligned}
\gamma &= \\
\alpha_1 &= \\
\alpha_2 &= \\
\beta &=
\end{aligned}
$$

How do we know which model provides the most satisfiable answer? An answer to this question is to have some **test data** and check which model predicts this data best. That is, we use **cross-validation** again.

⋆ Let the test data point be $x_{test} = (1, 1, 1)$ and the two DAGs be denoted M_1 and M_2. Then

$$p(x_{test}|\theta_1, M_1) =$$

$$p(x_{test}|\theta_2, M_2) =$$

The current approach has a few short-comings:

- There is no mechanism for incorporating *a priori* knowledge.
- The model selection strategy is very dependent on the parameter estimates. If we have few data points, the parameter estimates can be misleading.
- Model selection requires extra data (the test dataset).

The Bayesian learning paradigm helps surmount these difficulties.

Bayesian Learning

Given our **prior** knowledge $p(\theta)$ and the data model $p(\cdot|\theta)$, the Bayesian approach allows us to update our prior using the new data x as follows:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

where $p(\theta|x)$ is the **posterior distribution**, $p(x|\theta)$ is the likelihood and $p(x)$ is the **marginal likelihood** (evidence). Note

$$p(x) = \int p(x|\theta)p(\theta)d\theta$$

For a particular model structure M_i, we have

$$p(\theta|x, M_i) = \frac{p(x|\theta, M_i)p(\theta|M_i)}{p(x|M_i)}$$

Models are selected according to their posterior:

$$P(M_i|x) \propto P(x|M_i)p(M_i) = P(M_i)\int p(x|\theta, M_i)p(\theta|M_i)d\theta$$

The ratio $P(x|M_i)/P(x|M_j)$ is known as the **Bayes Factor**. Typically, $p(M)$ is uniform (the same for all models), so what decides what model we should be using is $p(x|M_i)$.

⋆ Let $x_{1:n}$, with $x_i \in \{0,1\}$, be i.i.d. Bernoulli: $x_i \sim \mathcal{B}(1,\theta)$

$$p(x_{1:n}|\theta) = \prod_{i=1}^{n} p(x_i|\theta) = \theta^m (1-\theta)^{n-m}$$

Let us choose the following **Beta** prior distribution:

$$p(\theta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

where Γ denotes the Gamma-function. For the time being, α and β are fixed **hyper-parameters**. The posterior distribution is proportional to:

$$p(\theta|x) \propto$$

with normalisation constant

Since the posterior is also Beta, we say that the Beta prior is **conjugate** with respect to the binomial likelihood. Conjugate priors lead to the same form of posterior.

Different hyper-parameters of the Beta $\mathcal{B}e(\alpha,\beta)$ distribution give rise to different prior specifications:

⋆

The generalisation of the Beta distribution is the Dirichlet distribution $\mathcal{D}(\alpha_i)$, with density

$$p(\theta) \propto \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

where we have assumed k possible thetas. **Note that the Dirichlet distribution is conjugate with respect to a Multinomial likelihood.**

Bayesian Prediction

We predict by marginalising over the posterior of the parameters

$$\begin{aligned} p(x_{n+1}|x_{1:n}) &= \int p(x_{n+1}, \theta | x_{1:n}) d\theta \\ &= \int p(x_{n+1}|\theta) p(\theta | x_{1:n}) d\theta \end{aligned}$$