732A96/TDDE15 Advanced Machine Learning

Graphical Models

Jose M. Peña
IDA, Linköping University, Sweden

Lecture 3: Parameter Learning

Contents

Literature

- Main source
  - Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. Chapters 8 and 9.
- Additional source
  - Koski, T. J. T. and Noble, J. M. A Review of Bayesian Networks and Structure Learning. *Mathematica Applicanda* 40, 51-103, 2012.

Parameter Learning for BNs: Maximum Likelihood

DAG	Parameter values for the conditional probability distributions
$\begin{tikzcd} \mathit{Sprinkler} \arrow[dr] & & \mathit{Rain} \arrow[dl] \arrow[dr] & \\ & \mathit{Wet\ Grass} & & \mathit{Wet\ Street} \end{tikzcd}$	$q(s) = (0.3, 0.7) = (\theta_{s_0}, \theta_{s_1})$ $q(r) = (0.5, 0.5) = (\theta_{r_0}, \theta_{r_1})$ $q(wg \mid r_0, s_0) = (0.1, 0.9) = (\theta_{wg_0 \mid r_0, s_0}, \theta_{wg_1 \mid r_0, s_0})$ $q(wg \mid r_0, s_1) = (0.7, 0.3) = (\theta_{wg_0 \mid r_0, s_1}, \theta_{wg_1 \mid r_0, s_1})$ $q(wg \mid r_1, s_0) = (0.8, 0.2) = (\theta_{wg_0 \mid r_1, s_0}, \theta_{wg_1 \mid r_1, s_0})$ $q(wg \mid r_1, s_1) = (0.9, 0.1) = (\theta_{wg_0 \mid r_1, s_1}, \theta_{wg_1 \mid r_1, s_1})$ $q(ws \mid r_0) = (0.1, 0.9) = (\theta_{ws_0 \mid r_0}, \theta_{ws_1 \mid r_0})$ $q(ws \mid r_1) = (0.7, 0.3) = (\theta_{ws_0 \mid r_1}, \theta_{ws_1 \mid r_1})$ $p(s, r, wg, ws) = q(s)q(r)q(wg \mid s, r)q(ws \mid r)$

- In general,

$$q(X_i = k \mid Pa_i = j) = \theta_{X_i = k \mid Pa_i = j}$$

- Recall that

$$p(X_i = k \mid Pa_i = j) = q(X_i = k \mid Pa_i = j)$$

Parameter Learning for BNs: Maximum Likelihood

- Given a sample $d_{1:N}$, the log likelihood function is

$$\log p(d_{1:N}|\theta, G) = \log \prod_l p(d_l|\theta, G) = \log \prod_l \prod_i p(d_l[X_i]|d_l[Pa_i], \theta)$$

$$= \log \prod_l \prod_i \theta_{X_i=d_l[X_i]|Pa_i=d_l[Pa_i]} = \log \prod_i \prod_j \prod_k \theta_{X_i=k|Pa_i=j}^{N_{ijk}}$$

$$= \sum_i \sum_j \sum_k N_{ijk} \log \theta_{X_i=k|Pa_i=j}$$

  where N_{ijk} is the number of instances in $d_{1:N}$ with $X_i = k$ and $Pa_i = j$.
- To maximize the log likelihood function subject to the constraint $\sum_k \theta_{X_i=k|Pa_i=j} = 1$ for all i and j, we maximize

$$\sum_i \sum_j \sum_k N_{ijk} \log \theta_{X_i=k|Pa_i=j} + \sum_i \sum_j \lambda_{ij} (\sum_k \theta_{X_i=k|Pa_i=j} - 1)$$

  where λ_{ij} are called Lagrange multipliers.[1]
- Setting to zero the derivative with respect to $\theta_{X_i=k|Pa_i=j}$ gives

$$\theta_{X_i=k|Pa_i=j} = -N_{ijk}/\lambda_{ij}$$

- Replacing in the constraint gives $\lambda_{ij} = -N_{ij}$ and $\theta_{X_i=k|Pa_i=j}^{ML} = N_{ijk}/N_{ij}$.

[1] Any stationary point of the Lagrangian function is a stationary point of the original function subject to the constraints. Moreover, the log likelihood function is concave.

Parameter Learning for BNs: Expectation Maximization Algorithm

- Let $d_{1:N}$ be an **incomplete sample**, i.e. $d_l[X_i] = ?$ for some i and l. Let $o_{1:N}$ denote the observed part of $d_{1:N}$, and $u_{1:N}$ the unobserved part.
- The log likelihood function over $o_{1:N}$ is

$$\log p(o_{1:N}|\theta, G) = \log \prod_l \sum_{u_l} p(o_l, u_l|\theta, G) = \sum_l \log \sum_{u_l} p(o_l, u_l|\theta, G)$$

- To maximize it subject to the constraint $\sum_k \theta_{X_i=k|Pa_i=j} = 1$ for all i and j, we maximize

$$\sum_l \log \sum_{u_l} p(o_l, u_l|\theta, G) + \sum_i \sum_j \lambda_{ij} (\sum_k \theta_{X_i=k|Pa_i=j} - 1)$$

- Its derivative with respect to $\theta_{X_i=k|Pa_i=j}$ is

$$\sum_l \frac{\sum_{u_l:c_l[X_i]=k, c_l[Pa_i]=j} \prod_{i'} \theta_{X_{i'}=c_l[X_{i'}]|Pa_{i'}=c_l[Pa_{i'}]}}{\theta_{X_i=k|Pa_i=j} \sum_{u_l} p(o_l, u_l|\theta, G)} + \lambda_{ij}$$

$$= \sum_l \sum_{u_l:c_l[X_i]=k, c_l[Pa_i]=j} \frac{p(u_l|o_l, \theta, G)}{\theta_{X_i=k|Pa_i=j}} + \lambda_{ij} = M_{ijk}/\theta_{X_i=k|Pa_i=j} + \lambda_{ij}$$

  where $c_l = \{o_l, u_l\}$ and $M_{ijk} = \sum_l \sum_{u_l:c_l[X_i]=k, c_l[Pa_i]=j} p(u_l|o_l, \theta, G)$.
- Setting the derivative to zero gives

$$\theta_{X_i=k|Pa_i=j} = -M_{ijk}/\lambda_{ij}$$

- Replacing this into the constraint gives $\lambda_{ij} = -M_{ij}$ and, thus, $\theta^{ML}_{X_i=k|pa_i=j} = M_{ijk}/M_{ij}$. **No closed form solution** but it suggests ...

Parameter Learning for BNs: Expectation Maximization Algorithm

---

EM algorithm

Set θ to some initial values
Repeat until θ does not change
    Compute $p(u_l|o_l, \theta, G)$ for all l /* E step */
    Compute M_{ijk}
    Set $\theta_{ijk} = M_{ijk}/M_{ij}$ /* M step */

- The EM algorithm increases $\log p(o_{1:N}|\theta, G)$ in each iteration. So, it is locally but not necessarily globally optimal.
- Note that computing $p(u_l|o_l, \theta, G)$ requires inference.
- Maximizing the log likelihood function over O is not only inefficient because no closed form solution exists, it is also ineffective due to **multimodality**, i.e. each completion of the data defines a unimodal function but their sum may be multimodal.

Parameter Learning for BNs: Expectation Maximization Algorithm

- Consider instead maximizing the expected log likelihood function over O

$$E[\log p(o_{1:N}, U_{1:N}|\theta, G)] = \sum_l \sum_{u_l} p(u_l|o_l, \theta, G) \log p(o_l, u_l|\theta, G)$$

$$= \sum_l \sum_{u_l} p(u_l|o_l, \theta, G) \sum_i \log \theta_{X_i = c_l[X_i] | Pa_i = c_l[Pa_i]}$$

  where $c_l = \{o_l, u_l\}$. Then

$$E[\log p(o_{1:N}, U_{1:N}|\theta, G)] = \sum_i \sum_j \sum_k M_{ijk} \log \theta_{X_i = k | Pa_i = j}$$

  where $M_{ijk} = \sum_l \sum_{u_l : c_l[X_i] = k, c_l[Pa_i] = j} p(u_l|o_l, \theta_G, G)$.

- Then, $\theta^{ML}_{X_i = k | Pa_i = j} = M_{ijk}/M_{ij}$. No closed form solution but it suggests the EM algorithm too.

Parameter Learning for MNs: Iterative Proportional Fitting Procedure

- Given a complete sample $d_{1:N}$, the log likelihood function is

$$\log p(d_{1:N}|\theta, G) = \sum_{K \in Cl(G)} \sum_k N_k \log \varphi(k) - N \log Z$$

where N_k is the number of instances in $d_{1:N}$ with $K = k$. Then

$$\log p(d_{1:N}|\theta, G)/N = \sum_{K \in Cl(G)} \sum_k p_e(k) \log \varphi(k) - \log Z$$

where $p_e(X)$ is the empirical probability distribution obtained from $d_{1:N}$.

- Let $Q \in Cl(G)$. The derivative with respect to $\varphi(q)$ is

$$\frac{\partial \log p(d_{1:N}|\theta, G)/N}{\partial \varphi(q)} = \frac{p_e(q)}{\varphi(q)} - \frac{1}{Z} \frac{\partial Z}{\partial \varphi(q)}$$

- Let $Y = X \setminus Q$. Then

$$\frac{\partial Z}{\partial \varphi(q)} = \sum_y \prod_{K \in Cl(G) \setminus Q} \varphi(k, \overline{k}) = \frac{Z}{\varphi(q)} \sum_y \prod_{K \in Cl(G) \setminus Q} \varphi(k, \overline{k}) \frac{\varphi(q)}{Z} = \frac{Z}{\varphi(q)} p(q|\theta, G)$$

where $\overline{k}$ denotes the elements of q corresponding to the elements of $K \cap Q$.

- Putting together the results above, we have that

$$\frac{\partial \log p(d_{1:N}|\theta, G)/N}{\partial \varphi(q)} = \frac{p_e(q)}{\varphi(q)} - \frac{p(q|\theta, G)}{\varphi(q)}$$

Parameter Learning for MNs: Iterative Proportional Fitting Procedure

- Setting the derivative to zero gives [2]

$$\varphi^{ML}(q) = \varphi(q)p_e(q)/p(q|\theta, G)$$

No closed form solution but ...

IPFP

Initialize $\varphi(k)$ for all $K \in Cl(G)$
Repeat until convergence
    Set $\varphi(k) = \varphi(k)p_e(k)/p(k|\theta, G)$ for all $K \in Cl(G)$

- IPFP increases $\log p(d_{1:N}|\theta, G)$ in each iteration. So, it is globally optimal.
- Iterative coordinate ascend method.
- Note that computing $p(k|\theta, G)$ in the last line requires inference. Moreover, the multiplication and division are elementwise.
- Note also that Z needs to be computed in each iteration, which is computationally hard. This can be avoided by a careful initialization.

[2] The log likelihood function is concave.

Contents

- Parameter Learning for BNs
  - Maximum Likelihood
  - Expectation Maximization Algorithm
- Parameter Learning for MNs
  - Iterative Proportional Fitting Procedure

Thank you