Assignment 2
Statistical Methods in Applied Computer Science
DD2447

Jim Holmström 890503-7571
jimho@kth.se

November 24, 2012

Exercise A1 Compute $p(D|T \in \text{Polytree})$ with Bernoulli CPD's
Show how to compute $p(D|T)$ where T is a DGM which is a polytree and $D = \{x_1, ..., x_N\}$ (and x_i is an assignment of values to all variables of T). Assume that all variables are binary and all CPD's Bernoulli.

Solution.

Theorem 0.1. Chain rule

$$p(\textstyle\bigcap_k x_k) = \displaystyle\prod_k p(\, x_k \Big| \textstyle\bigcap_{j:j<k} x_j \,) \tag{1}$$

Proof. We can extend the rule

$$p(x_i, x_j) = p(\, x_i | x_j \,) p(x_j) \tag{2}$$

since x_i, x_j is just events we might as well have $x_j = \bigcap_{j'} x_{j'}$ which makes the expression look like this

$$p(x_i \cap \Big(\textstyle\bigcap_{j'} x_{j'}\Big)) = p(\, x_i \Big| \textstyle\bigcap_{j'} x_{j'} \,) p(\textstyle\bigcap_{j'} x_{j'}) \tag{3}$$

by tail-recursion on $p(\bigcap_{j'} x_{j'}) = p(x_a \cap \Big(\bigcap_{j' \neq a} x_{j'}\Big))$ we will exhaust all j' which leaves us with the base-case $p(x_a)$. Writing out the entire trace of the recursion will gives us the wanted expression for $p(\bigcap_k x_k)$. $\square$

The first goal is to break down $p(\, D|T \,)$ into the CPD's for each node

$$p(\, D|T \,) = \phi\left(\{p(\, x_t | \cdot \,)\}_t\right) \tag{4}$$

this gives an explicit expression for the $p(D|T)$.

Since T is a polytree $\subset$ DAG it is always possible to sort the data topologically according to T.[1] We arrange the indices of the data this way so that it will be topologically sorted according to T and even if it's bad practice in math we replace the old indices with the new sorted ones. This results in variables always having the property $i < j \Rightarrow i$ higher up or equally high as j.[2]

Now apply the chain rule

$$p(D|T) = p(\{x_t\}_k|T) = \prod_k p\left(x_k \middle| \{x_j\}_{j:j<k}, T\right) \tag{5}$$

Next we note that since we have that the data is topologically sorted we know that

$$\left(D \setminus \{x_j\}_{j:j<k}\right) \cap pa\,(x_k) = \emptyset \tag{6}$$

in other words the parent cannot be at the same or lower topological level. This results in

$$\{x_j\}_{j:j<k} \supset pa\,(x_k) \tag{7}$$

which basically states that the parents, denoted $pa(\cdot)$, always is at a higher level.

The next step is to use the conditional independence information from T, which is that a r.v. is only dependent on the parents r.v.'s[3], together with the fact (7) in (5) we get

$$p(D|T) = \prod_k p\left(x_k \middle| \{x_j\}_{j:j<k}, T\right) = \prod_k p(x_k|pa\,(x_k)) \tag{8}$$

Finally since we only have cpt's[4] f_t[5], we have that

$$x_t \sim Ber\,(f_t(pa(x_t))) \tag{9}$$

and as explicitly written out as possible

$$p(D|T) = \prod_k p(x_k|pa\,(x_k)) = \prod_k f_k\,(pa(x_k))^{x_k}\,(1 - f_k\,(pa(x_k)))^{1-x_k} \tag{10}$$

□

Exercise A2 Marginalize over non-observed variables
Assume instead that each x_i is an assignment to a subset of the variables say O. Show how to marginalize over $V \setminus O$ (i.e., the non-observed variables).

[1] How we sort the data within a topological level doesn't matter.

[2] "higher" in the context of topological sorting means that it is higher up in the sorted DAG that grows downwards.

[3] At least for a completely visible graph.

[4] According to mail correspondence with the teacher.

[5] Since each distribution has only two states and needs to be normalized to 1, this is the same thing as setting $p \in (0, 1)$ in a bernoulli

Solution.

Exercise 11.3 EM for the mixtures of Bernoullis

- Show that the M step for ML estimation of a mixture of Bernoullis is given by

$$\mu_{kj} = \frac{\sum_i r_{ik} x_{ij}}{\sum_i r_{ik}} \tag{11}$$

- Show that the M step for MAP estimation of a mixture of Bernoullis with a $\beta(\alpha, \beta)$ prior is given by

$$\mu_{kj} = \frac{(\sum_i r_{ik} x_{ij}) + \alpha - 1}{(\sum_i r_{ik}) + \alpha + \beta - 2} \tag{12}$$

Solution. The M step is to optimize the auxiliary function Q with respect to π, θ'. Q is the expected posterior log-likelihood[6] with respect to the last parameter θ and the observed data D. [7] The expression for this is

$$Q(\theta', \theta) = \mathrm{E}\left[\log \mathcal{L}_{MAP}(\theta')|D, \theta\right] = \mathrm{E}\left[\log\left(\mathcal{L}(\theta')p(\theta')\right)|D, \theta\right] = \tag{13}$$

$$= \mathrm{E}\left[\ell(\theta') + \log p(\theta')|D, \theta\right] = \mathrm{E}\left[\ell(\theta')|D, \theta\right] + \log p(\theta') \tag{14}$$

and to derive this expression we introducing the latent variable z_i which corresponds to the hidden or missing variables which basically is the *r.v.* for how x_i belongs to the class k. [8] We start with MAP and then derive ML by setting a uniform priori.

$$Q(\theta', \theta) = \mathrm{E}\left[\sum_i \log p(x_i, z_i|\theta')\right] + p(\theta') = \tag{15}$$

since E is linear and we can factor on the different classes and the factors become given with the parameters (π_k, θ'_k) of class k:

$$= \sum_i \left[\mathrm{E}\ \log\left(\prod_k (\pi_k p(x_i|\theta'_k))^{\mathbb{I}(z_i=k)}\right)\right] + \log \prod_k p(\theta'_k) = \tag{16}$$

then log the inner parts and let E operate on the expression

$$= \sum_i \sum_k \left[\mathrm{E}\ [\mathbb{I}(z_i = k)] \log\left[p(x_i|\theta'_k)\right]\right] + \sum_k \log p(\theta'_k) = \tag{17}$$

[6] In the book it's actually just log-likelihood, but this will work in the same way instead of using $Q(\theta', \theta) + \log p(\theta')$ without derivation, we will use posterior-Q as Q instead.

[7] It can be shown that Q is so that the new parameters is always better or as good as the last one, but exclude the proof for this since it's not needed by the exercise.

[8] Responsibility $r_{ik} \triangleq p(z_i = k|x_i, \theta)$

then we have that the expected $\mathrm{E}\left[\mathbb{I}(z_i = k)\right]$ will be $p(z_i = k|x_i, \theta) = r_{ik}$ which is the expected class-belonging of x_i given the previous parameters θ. The log product rule gives us

$$= \sum_i \sum_k r_{ik} \log \pi_k + \sum_i \sum_k \left\{ r_{ik} \log p(x_i|\theta'_k) \right\} + \sum_k \log p(\theta'_k) \tag{18}$$

Now since we have that $\sum_{i,k} r_{ik} \log \pi_k \perp \sum_{i,k} \{ r_{ik} \log p(x_i|\theta'_k) \} + \sum_k \log p(\theta'_k)$ we can optimize the parameters π_k and θ'_k separately and only the θ'_k-term is asked for in the exercise which we denote $\ell(\theta'_k)$.

The model parameters in this case is denoted by μ_k which is a vector and in index-notation μ_{kj}. [9]

We start with the MAP

$$\hat{\mu}'_k = \underset{\mu'_k}{argmax} \left\{ \sum_i \left\{ r_{ik} \log p(x_i|\mu'_k) \right\} + \log p(\mu'_k) \right\} \tag{19}$$

which we find by $\frac{\partial \ell(\mu'_k)}{\partial \mu'_k} = 0$, where $p(x_i|\mu'_k) = \prod_j p(x_{ij} \middle| \mu'_{kj})$ is the multivariate Bernoulli[10] distribution for class k. The priori in the same way is $p(\mu'_k) = \prod_j p(\mu'_{kj})$. In our case for MAP this is $\beta(\alpha, \beta) = \frac{{\mu'_{kj}}^{\alpha-1}(1-\mu'_{kj})^{\beta-1}}{B(\alpha,\beta)}$.

[9] Not using Einstein notation for tensor product.
[10] Often called multinoulli.

$$\frac{\partial(\ell(\mu'_k))}{\partial\mu'_{kj}} = \frac{\partial \sum_i \left\{ r_{ik} \log \prod_{j'} p(\,x_{ij'} \middle| \mu'_{kj'}\,) \right\} + \log \prod_{j'} p(\mu'_{kj'})}{\partial\mu'_{kj}} = \tag{20}$$

$$= \frac{\partial \sum_{i,j'} \left\{ r_{ik} \log p(\,x_{ij'} \middle| \mu'_{kj'}\,) \right\} + \sum_{j'} \log p(\mu'_{kj'})}{\partial\mu'_{kj}} = \tag{21}$$

$$= \left\{ \frac{\partial \sum_{j \neq j'} (\cdot)}{\partial\mu'_{kj}} = 0 \right\} = \frac{\partial \sum_i \left\{ r_{ik} \log \left({\mu'_{kj}}^{x_{ij}} (1-\mu'_{kj})^{(1-x_{ij})} \right) \right\} + \log \frac{{\mu'_{kj}}^{\alpha-1} (1-\mu'_{kj})^{\beta-1}}{B(\alpha,\beta)}}{\partial\mu'_{kj}} = \tag{22}$$

$$= \frac{\partial \sum_i r_{ik} \left(x_{ij} \log \mu'_{kj} + (1-x_{ij}) \log(1-\mu'_{kj}) \right)}{\partial\mu'_k} + \tag{23}$$

$$+ \frac{\partial(\alpha-1)\log\mu'_{kj} + (\beta-1)\log(1-\mu'_{kj}) - \log B}{\partial\mu'_k} = \tag{24}$$

$$= \frac{\left(\sum_i r_{ik}x_{ij}\right) + \alpha - 1}{\mu'_{kj}} - \frac{\left(\sum_i r_{ik}(1-x_{ij})\right) + \beta - 1}{1-\mu'_{kj}} = \tag{25}$$

$$= \frac{(1-\mu'_{kj})\left(\sum_i r_{ik}x_{ij}\right) - \mu'_{kj}\left(\sum_i r_{ik}(1-x_{ij})\right)}{\mu'_{kj}(1-\mu'_{kj})} + \tag{26}$$

$$+ \frac{(1-\mu'_{kj})(\alpha-1) - \mu'_{kj}(\beta-1)}{\mu'_{kj}(1-\mu'_{kj})} = \tag{27}$$

$$= \frac{\left(\sum_i r_{ik}x_{ij}\right) - \mu'_{kj}\left(\sum_i r_{ik}\right) + (\alpha-1) - \mu'_{kj}(\alpha+\beta-2)}{\mu'_{kj}(1-\mu'_{kj})} \tag{28}$$

and now find the zero of this expression

$$\frac{\left(\sum_i r_{ik}x_{ij}\right) - \mu'_{kj}\left(\sum_i r_{ik}\right) + (\alpha-1) - \mu'_{kj}(\alpha+\beta-2)}{\mu'_{kj}(1-\mu'_{kj})} = 0 \tag{29}$$

$$\left(\sum_i r_{ik}x_{ij}\right) - \mu'_{kj}\left(\sum_i r_{ik}\right) + (\alpha-1) - \mu'_{kj}(\alpha+\beta-2) = 0 \tag{30}$$

$$\mu'_{kj}\left[\left(\sum_i r_{ik}\right) + \alpha + \beta - 2\right] = \left(\sum_i r_{ik}x_{ij}\right) + \alpha - 1 \tag{31}$$

$$\mu'_{kj} = \frac{\left(\sum_i r_{ik}x_{ij}\right) + \alpha - 1}{\left(\sum_i r_{ik}\right) + \alpha + \beta - 2} \tag{32}$$

which is the same as the equation (12). $\square$

By just looking at the expression for $\beta(\cdot)$ we see that

$$\beta(1,1) = U(0,1) \tag{33}$$

and knowing that ML is a MAP with a uniform priori we can just set $\alpha, \beta = 1$ in the derived expression to get the ML.

$$\mu'_{kj} = \frac{(\sum_i r_{ik} x_{ij}) + 1 - 1}{(\sum_i r_{ik}) + 1 + 1 - 2} = \frac{\sum_i r_{ik} x_{ij}}{\sum_i r_{ik}} \tag{34}$$

which is the same as the equation (11). $\square$