KRR - Learning the parameters of a Bayesian Network - Expectation Maximization

Alexandru Sorici

November 2019

When trying to quantify the way in which random variables in a Bayesian Network influence each other (i.e. determine their Conditional Probability Distributions - CPDs), the dataset used to estimate the CPDs may contain *unobserved/missing* variable values.

Example 1 In inferring a possible diagnosis for respitory problems, a patient may undergo several tests (e.g. body temperature, white blood cell count, sputum culture, chest X-ray). However, values for some of the tests may be missing in patient records.

Example 2 In diagnosing faults in an optical cable network (e.g. modeled dependencies between faults/root causes, intermediate faults on the network and observed alarms on end devices), the network management system fails to get some values from some network components (e.g. due to local network firewall, communication loss during reporting, missing upgrades in devices).

Categorizing missingness

In general the process by which missing values occur may be categorized in the following way:

- MCAR: data *missing completely at random*, i.e. some values are missing by a random sampling process. The fact that they are missing is completely unrelated to the data that *is* observed.
- MAR: data *missing at random*, i.e. the probability of being missing is the same only within groups defined by the observed data. For example, chest X-rays may be missing only for the subset of cases when temperature is normal, white blood cell count is normal and sputum is negative.
- NMAR: data *not missing at random* means that the probability of being missing varies for reasons that are unknown to us. Moreover, the "missingness" is due to conditions (unknown/unmodeled) related to the variable itself. For example, the chest X-ray may be missing, because the doctor determined that the lungs are healthy by some other *not modeled* modality (e.g. listening with the stethoscope). In this situation, the missingness contains information that cannot be inferred from observed values.

In this lab, we work under the assumptions of *MCAR* or *MAR*.
There are several ways for handling missing data:

- Discard samples with missing values
- Filling in missing values by different forms of averaging over samples with *known* values
- Derive a parameterized model of a Bayesian Network by maximizing the likelihood of the *observed* data.

In this lab we analyze the latter option, using an algorithm called Expectation Maximization.

Expectation Maximization (EM)

The EM algorithm is used to find (local) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. This can happen because values are *unobserved*, or because the model is assumed to have *latent variables* as part of the conceptual design (e.g. Gaussian Mixture Models, Hidden Markow Models).

Maximum Likelihood solutions typically involve taking derivatives of the likelihood function (L) with respect to *to all* unknown values, i.e. parameters *and* latent variables. This is often intractable, due to exponentially growing complexity of considering possible latent space value combinations.

The **Expectation Maximization (EM)** algorithm provides a means to *approximate* the likelihood of observed data by iteratively alternating between two steps: assume some complete parametrization of the statistical model and compute expectations of the *unobserved* variables; then use these expectations to compute better estimates for the parameters of the model.

Formalization We have a statistical model which generates a set $\mathbf{X}$ of observed variables and a set $\mathbf{Z}$ of missing/latent variables. The model is governed by a set of unknown parameters θ.

The likelihood function $L(\mathbf{X}, \mathbf{Z}; \theta) = p(\mathbf{X}, \mathbf{Z}|\mathbf{theta})$.

The MLE of the *observed* data is $L(X; \theta) = p(\mathbf{X}|\theta) = \int p(\mathbf{X}, \mathbf{Z}|\theta)\mathbf{dZ}$. This quantity is often intractable.

The EM algorithm tries to find the MLE of the marginal (observed data) likelhood ($L(X; \theta)$) by iteratively applying these two steps:

- *Expectation step (E)*: Define $Q(\theta|\theta^{(\mathbf{t})})$ as *the* ***expected value*** *of the* ***likelihood*** *function of* θ*, with respect to the current* ***conditional distribution*** *of* $\mathbf{Z}$ *given* $\mathbf{X}$ *and the current estimates of parameters* $\theta^{(\mathbf{t})}$:

$$Q(\theta|\theta^{(t)}) = \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^{(\mathbf{t})}}\left[logL(\mathbf{X}, \mathbf{Z}; \theta)\right]$$

- *Maximization step (M)*: Find the parameters θ that maximize $Q(\theta|\theta^{(\mathbf{t})})$, i.e.:

$$\theta^{(\mathbf{t+1})} = \underset{\theta}{argmax}\ Q(\theta|\theta^{(\mathbf{t})})$$

EM in Bayesian Networks

The EM algorithm can be adapted to the case of Bayesian Networks, when we are dealing with the case of unobserved variables. In this context, EM is used to estimate the probabilities $\theta_{i,j,k} = p(X_i = k|pa(X_i) = j)$ from *observed data* (i.e. only those nodes whose values are not missing).
From the previous lab, we remember that the form of the MLE over a whole dataset D for a Bayesian Network is:

$$L(D; \Theta) = \sum_{i,j,k} log(\theta_{i,j,k}) N_{i,j,k}$$

where $N_{i,j,k} = \sum_{d=1}^{card(D)} \mathbf{I}_{pa(X_i^{(d)})=j, X_i^{(d)}=k}$

We introduce the following notations:

- $X_i^{o,d}$ - the value for node X_i of the BN has been *observed* in the d^th sample, $d \in [1..card(D)]$
- $X_i^{m,d}$ - the value for node X_i of the BN is *missing* in the d^th sample, $d \in [1..card(D)]$

The Q Function for BN The Q function of the EM algorithm has the following form for Bayesian Networks:

$$Q(\theta, \theta^{(t)}) = \sum_{i,j,k} log(\theta_{i,j,k}) \hat{N}_{i,j,k}$$

where $\hat{N}_{i,j,k} = \mathbb{E}(\sum_{d=1}^{card(D)} \mathbf{I}_{pa(X_i^{(d)})=j, X_i^{(d)}=k}|\mathbf{X^o}, \theta^{(t)}) = \sum_{d=1}^{card(D)} \gamma_{i,j,k}^{(d)}$.

This means that $\hat{N}_{i,j,k}$ are the *expected* count of the number of records where the value of node X_i is k and the set of values of its parent nodes $pa(X_i)$ are j. This expectation is computed by the expression:

$$\gamma_{i,j,k}^{(d)} = p(X_i^{(d)} = k, pa(X_i)^{(d)} = j|\mathbf{X}^{o,d}; \theta^{(t)})$$

Each $\gamma_{i,j,k}^{(d)}$ represents the probability that $X_i^{(d)}$ is k and the set of values of its parent nodes $pa(X_i)^{(d)}$ are j, conditionally to the *observed* nodes in the d-th observation of the BN.

M step for BN If we have the estimates $\hat{N}_{i,j,k}$, the M step is maximization of $\sum_k log(\theta_{i,j,k})\hat{N}_{i,j,k}$, under constraint $\sum_k \theta_{i,j,k} = 1$ for all values i, j.

The result of the M step is

$$\theta_{i,j,k}^{(t+1)} = \frac{\hat{N}_{i,j,k}}{\sum_k \hat{N}_{i,j,k}}$$

E step for BN The E step computes, for each node X_i, the conditional joint probabilities of X_i and its parents $pa(X_i)$: $\gamma_{i,j,k}^{(d)}$, as well as the *expected* counts $\hat{N}_{i,j,k}$.

Computation of $\gamma_{i,j,k}^{(d)}$ is based on the *Junction Tree* algorithm *and* the propagation of evidence $\mathbf{X}^{o,d}$ on the junction tree.

The algorithm for the E step has the following steps:

1. Initialize parameters $\theta_{i,j,k}^0$ to random values, respecting the probability constraint $\sum_k \theta_{i,j,k} = 1$
2. Compute the *Junction Tree representation* $\mathcal{T}$ of the BN
3. E-step
   - Initialize counters $\hat{N}_{i,j,k} = 0$
   - Iterate through examples $d \in D$
     - Incorporate the *observed* nodes $X_i^{d,o}$ as evidence in the clique nodes of $\mathcal{T}$
     - Calibrate $\mathcal{T}$, i.e. run the belief-propagation algorithm
     - Compute $\gamma_{i,j,k}^{(d)} = p(X_i^{(d)} = k, pa(X_i)^{(d)} = j|\mathbf{X}^{o,d}; \theta^{(t)}) \propto \sum_{C_z \setminus \{X_i, pa(X_i)\}} \phi_{C_z}^*$ by *identifying the clique* C_z, such that $\{X_i, pa(X_i)\} \subset C_z$, marginalizing over $z \setminus X_i, pa(X_i)$ and *normalizing*
     - add $\gamma_{i,j,k}^{(d)}$ to $\hat{N}_{i,j,k}$
4. M-step: compute maximization as explained above
5. repeat E, M steps *nr_repeats* time