PGMs week 9 Learning: Latent Variables

DRAFT

- Watch the 18 series of Koller videos.
  - The first situation leading to latent variables is when some of our training data is missing. The MAR principle has to apply to enable valid parameter estimates. Although of great practical importance in semi-supervised systems, we will not concern us much with this.
  - Quite often having latent states as part of our model, can make it much more powerful. Examples of this is the active basis pdf in a Gaussian Mixture Model (GMM, see Barber Chapter 20), the active state in a Hidden Markov Model (HMM) or in a temporal CRF.
  - However, this couples our parameters and causes our parameter space to have multiple/local optima.
  - We can optimise via the EM algorithm or directly using standard optimisation techniques.
- Read Barber chapter 11 (skip 11.5).
- Summary of the EM algorithm:

  We divide the variables in our model into three groups: $\mathbf{v}$ are the observed variables, $\mathbf{h}$ the latent/hidden variables and θ the parameters. We want to maximise:

$$p(\mathbf{v}|\theta) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}|\theta). \tag{1}$$

  Let $q(\mathbf{h})$ be an arbitrary distribution over the latent variables. We can easily show that :

$$\log p(\mathbf{v}|\theta) = \mathcal{L}(q, \theta) + \mathrm{KL}(q \parallel p), \tag{2}$$

  with

$$\mathcal{L}(q, \theta) = \sum_{\mathbf{h}} q(\mathbf{h}) \log \left[\frac{p(\mathbf{v}, \mathbf{h}|\theta)}{q(\mathbf{h})} \right] \tag{3}$$

$$\mathrm{KL}(q \parallel p) = -\sum_{\mathbf{h}} q(\mathbf{h}) \log \left[\frac{p(\mathbf{h}|\mathbf{v}, \theta)}{q(\mathbf{h})} \right] \tag{4}$$

  (You can verify this by substituting and then simplifying while remembering that $q(\mathbf{h})$ is a distribution.) We note that $\mathcal{L}(q, \theta)$ is a *functional* (i.e. it has an unspecified *function* q as parameter to optimise over – this is of importance too when we do variational inference) and $\mathrm{KL}(q \parallel p)$ is the Kullback-Leibler distance between distributions q and p. $\mathrm{KL}(q \parallel p) \geq 0$ with equality only if the two distributions coincide. This implies that $\mathcal{L}(q, \theta)$ is a lower bound for $\log p(\mathbf{v}|\theta)$. We can now optimise by iterating over following two-step method until convergence is achieved:

  1. In the E-step we want to maximise this lower bound w.r.t. $q(\mathbf{h})$. We do so by annihilating the KL distance i.e. by setting

$$q(\mathbf{h}) = p(\mathbf{h}|\mathbf{v}, \theta). \tag{5}$$

     This requires that we do inference to determine the distribution $q(\mathbf{h})$. In essence this step replaces the hidden data with a distribution for it.
  2. In the M-step we hold $q(\mathbf{h})$ fixed and maximise this lower bound $\mathcal{L}(q, \theta)$ w.r.t. θ. If not already at a maximum, this will cause the log-likelihood function $\log p(\mathbf{v}|\theta)$ to rise. Let us simplify a bit more:

$$\sum_{\mathbf{h}} q(\mathbf{h}) \log \left[\frac{p(\mathbf{v}, \mathbf{h}|\theta)}{q(\mathbf{h})} \right] = \sum_{\mathbf{h}} q(\mathbf{h}) \log p(\mathbf{v}, \mathbf{h}|\theta) - \sum_{\mathbf{h}} q(\mathbf{h}) \log q(\mathbf{h}).$$

The term on the right does not depend on θ and we can ignore it, leaving us with the auxiliary function

$$\mathcal{Q}(\theta) \equiv \sum_{\mathbf{h}} q(\mathbf{h}) \log p(\mathbf{v}, \mathbf{h}|\theta) = \langle \log p(\mathbf{v}, \mathbf{h}|\theta) \rangle_{q(\mathbf{h})} \tag{6}$$

to optimise w.r.t. θ. This is a particularly nice function to work with if our distribution is part of the exponential family. The $\log$ will cancel with the $\exp$ leaving us with a very clean function to optimise over.

We can also extend the M step to do a MAP estimate instead of an ML estimate – for this we need to maximise $\mathcal{L}(q, \theta) + \log p(\theta)$ w.r.t. θ.

- Exercise(s):
  - Create data corresponding to a simple scalar GMM (see Barber Chapter 20 for more details if required). Now learn the parameters of this GMM by applying the EM algorithm to the created data. Find initializations leading to different local optima. You might want to simplify this exercise even further by assuming the variances known.
  - Alternatively in Barber Example 11.2 on pg 261 is particularly nice to visualise the optimisation process. Implement and explore it.
  - If you feel the need for more excitement you can obtain an expression for the derivatives and then verify that it approaches zero as we progress towards convergence.
  - You can also directly optimise the parameters (using for instance steepest ascent optimisation) and compare it to the result obtained via the EM algorithm.