PGMs week 7 Learning: Maximum Likelihood Estimation

We now have a fair idea as how to do inference in a graphical model. We have, however, far from exhausted the topic, especially for cases where there are loops in the model, or the marginalisation integrals are just not feasible. We will return to the idea of approximate inference towards the end of this study.

To do inference, we need a PGM and the corresponding parameters it depends on. We will assume that the structure of the PGM is known. We therefore now turn to investigate methods to determine the parameters of the model. We first concider Maximum Likelihood (ML) Estimation, probably the simplest approach we can take. This method sets the model parameters $\mathbf{\Theta}$ in such a way that the total likelihood of the data given the parameters is maximised, i.e. find

$$\mathbf{\Theta}^{\text{ML}} = \underset{\mathbf{\Theta}}{\text{argmax}}\ p(\mathcal{D}|\mathbf{\Theta}). \tag{1}$$

This is usually done by finding where the derivative of the likelihood function is equal to zero. Note that $\mathbf{\Theta}$ (and not $\mathcal{D}$) is the argument to p, changing it from a probability distribution to a *likelihood* function. In spite of its simplicity this technique is very useful and often forms a building block of more involved techniques.

- Watch the 03.4, 05.3, 15.1, 15.2, 16.1, 16.2 Koller videos, as well as the 13 series of videos (by Andrew Ng). Note that Koller's examples are for discrete random variables, where-as in the exercise (lower down) we will be looking at an example with continuous variables:
  - 03.4 Introduces the concept of a "Plate" in a graphical model, a mechanism used for repeating parts of the model. We use it here to handle repeated training feature vectors.
  - 05.3 Reminds us about the form of log-linear and particularly CRF models.
  - 15.1 covers the basics of MLE, 15.2 applies it to BNs. With fully observed data the estimates decomposes nicely.
  - Video 16.1 applies MLE to MRFs via log-linear models.
    - Firstly remember that MRFs, including their log-linear versions, model the joint distribution $p(\mathbf{X}|\mathbf{\Theta})$.
    - The presence of the normalizing partition function $Z(\mathbf{\Theta})$ couples the parameters and thus confounds the nice decomposing property we saw with BN's.
    - However, the log-likelihood function is concave and we can optimise it via gradient techniques.
    - For $\mathbf{X}$ the gradient is shown to be:

      $$\frac{\partial}{\partial \Theta_i}\frac{1}{M}l(\mathbf{\Theta}:\mathcal{D}) = \mathtt{E}_{\mathcal{D}}[f_i(\mathbf{X})] - \mathtt{E}_{\mathbf{\Theta}}[f_i], \tag{2}$$

      with

      $$\mathtt{E}_{\mathcal{D}}[f_i(\mathbf{X})] = \frac{1}{M}\sum_m f_i(\mathbf{x}[m]),$$

      and

      $$\mathtt{E}_{\mathbf{\Theta}}[f_i] = \sum_{\mathbf{x}} f_i(\mathbf{x})p(\mathbf{x}|\mathbf{\Theta}).$$

      I.e. the derivative is the difference between the empirical (or arithmetic) and the statistical average of f_i. Koller proves this for *discrete* variables. Question: Is this also applicable to continuous variables?
    - The first term requires one pass through the data to calculate the empirical average.
    - The second term requires that for every consecutive estimate of $\mathbf{\Theta}$ we infer $p(\mathbf{x}|\mathbf{\Theta})$ (note that the observed data no longer plays a role here). This is then used to calculate the required second expectation term. When the statistical expectations converge to the empirical expectations, we have found $\mathbf{\Theta}^{\text{ML}}$.
  - Video 16.2 applies MLE to estimate the parameters of CRFs.

  - Firstly remember that CRFs, including their log-linear versions, model the conditional joint distribution $p(\mathbf{Y}|\mathbf{x}, \mathbf{\Theta})$. We are not modelling the distribution over the $\mathbf{X}$'s, they are constrained to the values we observed.
  - This conditioning means that the joint distribution is now sliced at the observed value $\mathbf{x}$ and the normalization now needs to take this observed value into account. The normalizing partition function now becomes $Z_{\mathbf{x}}(\mathbf{\Theta})$ – it not only couples the parameters, but also is dependent on the specific observation made.
  - Note that the log-linear model is a rather specific form, for each weight parameter we need precisely one feature function. It is not mentioned in the videos, but to hammer our original likelihood function into this form we usually have to make use of indicator/Kronecker delta functions.
  - The log-likelihood function once again is concave and we can optimise it via gradient techniques.
  - When we have our log-likelihood function in log-linear form, there is a very elegant trick to calculate the derivatives while also avoiding the matrix calculus. To do this we need to know the values of the feature functions, and (via inference) the probabilities for our labels $Y[m]$ given the observations $\mathbf{x}[m]$ and our current parameter estimates. The gradient now reduces to:

$$\frac{\partial}{\partial \Theta_i} \frac{1}{M} l(\mathbf{\Theta} : \mathcal{D}) = \frac{1}{M} \sum_m \left(f_i(\mathbf{x}[m], \mathbf{y}[m]) - \mathtt{E}_{\mathbf{\Theta}}[f_i(\mathbf{x}[m], \mathbf{Y})] \right). \tag{3}$$

    Note that the last expectation is over $\mathbf{Y}$ while $\mathbf{x}$ is kept fixed at its observed value.
  - To determine this last expectation term we now require inference for *each* $\mathbf{x}[m]$ at *each* gradient step. This might seem rather expensive, but also bear in mind that the CRF avoids the cost of modelling $\mathbf{x}$.

- Quite often a training procedure can misuse the parameters of a model with many degrees of freedom, resulting in it specialising (overfitting) on the training data. Regularisation is a technique used to counter this. The 13 series of videos, lifted from Andrew Ng's Machine Learning course, discusses this. As discussed there it appears as a type of ad-hoc trick, but its origin (and much more flexibility in ways to use it) will become clear after we studied Bayesian estimation at a later stage.

- Read Barber chapter 8. Some things to note:
  - The empirical distribution describes the observed data and has use in parameter estimation. Note the use of Kronecker and Dirac delta functions to cover discrete and continous spaces respectively.
  - The Kullback-Leibler divergence measures the "difference" between two distributions. It plays an important role in some objective functions.
  - ML Estimation corresponds to:

$$\mathbf{\Theta}^{\text{ML}} = \underset{\mathbf{\Theta}}{\text{argmin}} \ \text{KL}(p(\mathcal{D}|\mathbf{\Theta})||p_E(\mathcal{D})), \tag{4}$$

    where p_E is the empirical distribution of the data.
  - Mutual information measures the reduction in the number of bits we need to code variable X if we know Y. It plays an important role in some objective functions.
  - Barber lists an extensive series of distribution functions. Many of them are conjugate priors for other densities – they will make more sense once we have covered Bayesian techniques. However, it is well worth perusing.
  - Maximum Likelihood Estimation is equivalent to minimising the Kullback-Leibler divergence between the empirical distribution and the distribution that we want to fit to the data. We will later see that fitting one distribution to another via the Kullback-Leibler distance, also has other uses in estimation.
  - The Barber version of Eq 3 is in Chapter 9, Eq 9.6.56 (pg 235). I find the notation he uses for expectation much more concise.

- Exercise:
  1. Estimating the parameters of a simple log-linear MRF:

     Let us consider the estimation of the parameters of a Gaussian model from a few training data examples. For simplicity we will stick to scalars (the same principles applies to the vector case). For training data, use Octave to generate $N = 100$ standard Gaussian random values.
     (a) Draw the plate model describing the likelihood function describing these random values, and write down a mathematical expression for this likelihood function. Then determine the ML values for the mean and variance by setting the derivative of the likelihood function w.r.t. the mean and variance respectively to zero (as in Eq 1 above). You should get the well-known standard expressions for mean and variance estimates that octave also employs when you call the "mean" and "var" functions (in the case of var specifically the 'biased' version i.e. normalized by N and not $N - 1$).

(b) We can rewrite the humble Gaussian as a log-linear model:

$$p(x|a,\sigma^2) = \frac{e^{(x-a)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}} = \frac{1}{Z(a,\sigma^2)} e^{x^2(\frac{-1}{2\sigma^2})+x(\frac{a}{\sigma^2})} \tag{5}$$

and use Eq 2 to determine the derivatives. (The summation now becomes integration.) Solve the parameters by setting the derivitaves to zero.

(c) Estimate the parameters by minimising the KL-divergence between the required Gaussian distribution and the empirical distribution for the data (i.e. the approach of Eq 4 above). For more detail see Barber Section 8.2.1 (pg 169-179) and 8.7.3 (pg 185). You should get the same result as before. (Remember to use Dirac delta's for continuous variables.)

(d) Although this problem has got nice closed form solutions, this is not always the case. Often we need an iterative gradient accent method to determine the optimal parameters. We do this by starting with a sensible initial guess Θ_0 for the parameters and then iterating it via:

$$\mathbf{\Theta}_t = \mathbf{\Theta}_{t-1} + \mu \frac{\partial}{\partial \mathbf{\Theta}} l(\mathbf{\Theta} : \mathcal{D}) \tag{6}$$

till we get convergence. μ is a small positive step length. Use this approach and see if you can converge on the same solution that the above closed form expressions gave.

2. Estimating the parameters of a simple CRF:

Let us consider the estimating the weight vector $\mathbf{w}$ of a logistic regression (two-class) classifier. The logistic regression function is given by classifier is given by:

$$p(Y = 1|\mathbf{x} : \mathbf{w}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+w_0)}} \tag{7}$$

We can view this as a potential on the observed feature vector $\mathbf{x}$ and class label Y given by:

$$\tilde{p}(Y|\mathbf{x} : \mathbf{w}) = \begin{cases} e^{\mathbf{w}^T\mathbf{x}+w_0} & Y = 1, \\ 1 & Y = 2 \end{cases} \tag{8}$$

Combined with a partition function of $Z(\mathbf{x}, \mathbf{w}) = 1 + e^{\mathbf{w}^T\mathbf{x}+w_0}$ this reduces to that given in Eq 7.

For simplicity we will use one-dimensional feature vectors i.e. scalars x. For training data, use Octave to generate 50 unity variance Gaussian random value samples centered around 0; these will have class label $Y = 1$. A further 50 samples with similar variance and centered around $+2$ will be the training data for the alternate class with class label $Y = 2$.

(a) From first principles derive the partial derivatives for the two parameters w_0 and w_1 (can you extrapolate the result to the vector case?). Then use the gradient ascent method described by Eq 6 to iteratively solve the parameter values.

(b) Use Eq 3 to determine the derivatives and compare with that obtained in the previous question. To do this, note that we have two parameters to estimate: w_0 and w_1. We therefore need two feature functions – we use Kronecker delta functions (indicator functions) to form them.

$$f_0(\mathbf{x}, y) = \delta_{y,1}$$
$$f_1(\mathbf{x}, y) = x\delta_{y,1}$$

The logistic regression function now becomes:

$$p(Y = y|x) = \frac{1}{Z(x, \mathbf{w})} e^{w_0 f_0(\mathbf{x},y)+w_1 f_1(\mathbf{x},y)} \tag{9}$$

(c) Try to duplicate the above by iteratively matching the logistic regression to the empirical distribution as described by Eq 4.

3. Now, based on the 13.x video series, consider the regularisation of the above estimates. Is the same type of regularisation applicable to both the mean, variance and the weight vector $\mathbf{w}$?

4. Can you extend the above CRF example to create something akin to an HMM, but without the requirement to explicitly model the distribution of the feature vectors? (You will have to make use Kronecker deltas again.)