PGMs session 9 Learning: Bayesian estimation

In session 8, we estimated the parameters of a model by finding the set of parameter values that maximise the likelihood function, $p(\mathcal{D}|\boldsymbol{\Theta})$; however, this maximum-likelihood estimation was an ad-hoc approach that does not work well all scenarios. A more principled approach – the Bayesian approach – is to model *everything* you are uncertain about as RVs. According to this approach, the parameters, $\boldsymbol{\Theta}$, are now considered RVs, and estimating the parameters from data, $\mathcal{D}$, amounts to calculating the posterior distribution

$$p(\boldsymbol{\Theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}).$$

We call this approach to parameter estimation the *full Bayesian approach.* For this approach, we have to specify a prior distribution $p(\boldsymbol{\Theta})$ for the parameters; this prior distribution also contains parameters, which are called *hyperparameters*.

In the full Bayesian approach, we represent our knowledge of the parameters as probability distributions, not specific values. This could make inference difficult and slow. An approach that avoids the complexity of the full Bayesian approach, but one that is more principled than MLE is *maximum a posteriori (MAP) estimation*, where we find the set of parameter values that maximise the posterior distribution over the parameters, or

$$\begin{aligned}\boldsymbol{\Theta}^{\text{MAP}} &= \underset{\boldsymbol{\Theta}}{\text{argmax}} \quad p(\boldsymbol{\Theta}|\mathcal{D}) \\ &= \underset{\boldsymbol{\Theta}}{\text{argmax}} \quad p(\mathcal{D}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}).\end{aligned}$$

In the practical part of this assignment, we will use both MAP estimation and the full Bayesian approach. For both approaches, we have to choose a prior distribution, $p(\boldsymbol{\Theta})$. For a specific likelihood function, $p(\mathcal{D}|\boldsymbol{\Theta})$, we try to choose the prior distribution such that the posterior distribution, $p(\boldsymbol{\Theta}|\mathcal{D})$, is in a convenient form and easy to calculate; this requires a number of new distributions (e.g. the Dirichlet and gamma distributions; see also the concept of a *conjugate distribution* below). It is important to have an idea of how to choose prior parameter distributions as well as how to work with them; however, we will sidestep this complexity for the practical part by looking at a problem for which we can use a Gaussian prior distribution for the parameters.

1 Preparation: Watch the Koller videos on Bayesian estimation

Watch the following videos in the Coursera course Probabilistic Graphical Models 3: Learning, Week 1:

- *Bayesian Parameter Estimation for BNs: Bayesian Estimation (15:27)*
- *Bayesian Parameter Estimation for BNs: Bayesian Prediction (13:40)*
- *Bayesian Parameter Estimation for BNs: Bayesian Priors for BNs (17:02)*

Watch the following videos in the Coursera course Probabilistic Graphical Models 3: Learning, Week 2:

- *Parameter Estimation in MNs: MAP Estimation for MRFs and CRFs (9:59)*

Notes on these videos:

- The 3 videos about *Bayesian Parameter Estimation for BNs* follow the full Bayesian approach, whereas the video *Parameter Estimation in MNs: MAP Estimation for MRFs and CRFs (9:59)* discusses MAP estimation.
- The video *Bayesian Parameter Estimation for BNs: Bayesian Estimation (15:27)* introduces the idea of *conjugate prior distributions*, which are specific choices for the parameter prior distribution such that the posterior distribution is in same parametric family (e.g. the Dirichlet distribution is a conjugate prior distribution for the multinomial distribution). Take note of the idea; however, we will not use it in the practical parts of this assignment.

- With continuous features, the required integration required for marginalisation in the full Bayesian approach often makes this impossible to do directly. One of our alternatives is to switch to max-sum inference instead of sum-product inference. The maximisation can then be done by finding the maximal points on the respective distributions – this can be done via differentiation/optimisation, an easier alternative to the integrations originally required. In effect this finds the most likely parameter values for the model given the observed data.
- In Bayesian estimation, equations often take the form where it appears as if the training data is supplemented with a number of extra "ghost" features which correspond to the characteristics of the prior distribution– this is the so-called "equivalent sample size" introduced by the prior distribution, as discussed in *Bayesian Parameter Estimation for BNs: Bayesian Prediction (13:40)*. As the number of data points increase, ML and MAP estimation converge to the same values.
- In *Parameter Estimation in MNs: MAP Estimation for MRFs and CRFs (9:59)*, Koller makes the connection between using a parameter prior and regularisation – regularisation can be viewed as choosing a prior distribution for the paramaters. Being a distribution, the prior distribution over the parameters in turn is governed by its own parameters, now known as *hyperparameters*. Closer inspection should show that these hyperparameters are exactly what we tune when we set regularisation parameters. But now they have physical meaning in terms of the constraints they place on the model parameters.

2 Preparation: Read sections of Barber Chapters 8 and 9 on Bayesian estimation

Read the following sections of Barber Chapter 8 about statistics for machine learning:

- Section 8.3: This section covers several distributions that are useful to know of. By now you should be able to appreciate the relevance of most of these distributions: some of these distributions are used for "normal" RVs (e.g. Bernoulli, categorical, Gaussian), whereas others are used as prior distributions over parameters (e.g. gamma, beta, Dirichlet, Laplace). For this module, we'll stick to familiar discrete and Gaussian distributions though.
- Section 8.5: This section discusses the exponential family, which is a general description of many common distributions. It also introduces the idea of a conjugate prior distribution – every likelihood that is a member of the exponential family has a conjugate prior distribution. It is often mathematically convenient to choose the form of the prior to match that of the posterior - this makes of it simply another term similar to the others in the likelihood product. This is called a conjugate prior. For multinomials (i.e., probability tables) we use the Dirichlet prior. The joint prior for the mean and precision of a 1-dimensional Gaussian is the Gaussian-gamma distribution, and for a multi-dimensional Gaussian it is the Gaussian-Wishart distribution. Take note of these concepts, but we will not use them futher in this module.
- Section 8.6: Reread this section – it is important that you understand it.
- Section 8.8: In session 8, you would have read Subsection 8.8.1; now study Subsection 8.8.2 and read through Subsection 8.8.3.

Read the following sections of Barber Chapter 9 about learning as inference:

- Section 9.1: Here Barber introduces the perspective that learning (parameters) is just a form of inference – this idea is important to understand. You can leave out the parts about making decisions (Subsections 9.1.2 and 9.1.4).
- Section 9.2: Barber introduces ML-II here, which learns the hyperparameters of a model using MLE. Of course, nothing prevents us – in principle – from considering the hyperparameters to be random variables in their turn too, and instead tune their hyper-hyperparameters on a validation set. However, we have to stop somewhere, and at this level of abstraction, we simply use MLE or choose them to be something sensible.
- Section 9.4: This section is about Bayesian parameter learning for BNs, which in essence is just inference. It is important that you understand the ideas (modelling the problem, decomposing the problem given complete data, hyperparameters), but do not get bogged down in the calculations with the beta and Dirichlet prior distributions.

3 Information: Implementation of affine Gaussian factors in `emdw`

First study `lingauss.pdf`, available from the SUNLearn page. Also update the `sqrtmvg` code – for that you will need to download the updates from SUNLearn, and then from the `emdw` root directory untar it with:

```
tar -xvzf untar_from_emdwroot.tgz
```

We do not (currently) have an implementation of an explicit `AffineGaussian Factor` in `emdw`, that is still in the pipeline. However, we have got code that will build the joint Gaussians corresponding to such an affine transformation. To build an affine Gaussian describing the joint distribution of X, Y, you will have to use the

```
static  SqrtMVG* constructAffineGaussian(
    const SqrtMVG& xPdf,
    const prlite::ColMatrix<double>& aMat,
    const prlite::ColVector<double>& cVec,
    const emdw::RVIds& newVars,
    const prlite::ColMatrix<double>& noiseR,
    ... more stuff ...);
```

function available in `sqrtmvg.hpp`. Study its interface, documentation available from line 495. Things to note:

- Note that `static` right at the beginning. In C++ a `static` member function is one that operates independently from the class it is seemingly embedded into – you can think of it as just a separate function saved there for convenience. Indeed, you can call it directly without an associated object:

  ```
  rcptr<Factor> joint = uniqptr<SG>(
          SqrtMVG::constructAffineGaussian(*prior2joint, aMat, cVec, yIds, noiseR) );
  ```

- Careful, the `aMat` mentioned here is the transpose of that in `lingauss.pdf`.
- For our purposes you can simply declare `cVec` as a zero-length vector; i.e., `prlite::ColVector<double> cVec;` (Its typical use is as a control input in Kalman filters.)
- `newvars` contains the RV IDs of Y. However, at this stage the distribution for Y does not exist as yet, it will be created via this function as a part of the joint.
- In typical use `noiseR` is a diagonal matrix containing the *standard deviations* (not variances) of the noise term ν.
- The result of all of this is just another `SqrtMVG`, now describing the joint distribution of X, Y. That implies that all the other factor-operators for `SqrtMVG` that you have used before, are directly available.

4 Practical: MAP estimation for a simple regression example

For this question, we will revisit the line fitting problem of Question 4 of session 8, and use MAP estimation to determine the y-coordinates of the two endpoints of the line, y_I and y_F. Since you still do not need `emdw`'s functionality for this question, you again do not need to use C++ for coding.

(a) (Paper:) Redraw the BN for this model with M measurements using plate notation.

(b) (Paper:) Choose the prior parameter distribution as $p(\boldsymbol{\Theta}) = \mathcal{N}(y_I; \mu_I, \sigma_I^2) \cdot \mathcal{N}(y_F; \mu_F, \sigma_F^2)$. Use the likelihood function $p(\mathcal{D}|\boldsymbol{\Theta})$ from session 8 to write $\log p(\boldsymbol{\Theta}|\mathcal{D})$ as a sum of quadratic terms in terms of $\hat{y}^{[m]}$, y_I and y_F (except for a few extra terms, your answer should be the same as for Question 4(b) of session 8).

(c) (Paper:) Now derive $\boldsymbol{\Theta}^{\text{MAP}} = \overset{\text{argmax}}{\boldsymbol{\Theta}} \log p(\boldsymbol{\Theta}|\mathcal{D})$ using a similar procedure to that of Question 4(c) of session 8 (you should be able to reuse much of that answer).

(d) (Code:) Use the same procedure as Question 4(d) of session 8 to generate data (or reuse the same data). Choose appropriate parameters for the prior distribution $p(\boldsymbol{\Theta})$ and calculate $\boldsymbol{\Theta}^{\text{MAP}}$ for the generated data. Plot the actual line, data points, MAP estimate as well as ML estimate on the same graph. How similar are the MAP and ML estimates?

(e) (Code:) Repeat the sampling of measurements and subsequent paramater estimation, but vary the number of measurements. What is the smallest number of measurements that you need to calculate a MAP estimate? What is the effect of the number of measurements on the difference between the MAP and ML estimates?

(f) (Code:) Repeat the sampling of measurements and subsequent parameter estimation, but vary the parameters of the prior parameters distribution $p(\boldsymbol{\Theta})$ while keeping the number of parameters fixed. How do these parameters influence the difference between the MAP and ML estimates?

5 Practical: Full Bayesian estimation for a simple regression example

Finally, a teeny weeny bit of Bayesian work. We use exactly the same line fitting example as we did for the ML and MAP estimation. However, we are now interested in calculating the posterior *distribution* over the parameters y_I and y_F, instead of calculating *values* for them

(a) (Paper:) Redraw the BN for this model with M measurements using plate notation.

(b) (Paper:) Write the transformation from the parameters $\boldsymbol{\Theta}$ to the measurement $\hat{y}^{[m]}$ as an affine transformation.

(c) (Code:) Using `emdw`, specify a prior distribution over the parameters (use the same one as for the MAP estimation in Question 4), construct the factors with the measurements $\hat{y}^{[m]}$ using the `constructAffineGaussian` function, and then infer posterior distribution over the parameters given all the observed values $\mathcal{D} = \left(\hat{y}^{[1]}, \ldots, \hat{y}^{[M]}\right)$, using `emdw`'s built-in functionality.

Tip: Group your observed $\hat{y}^{[m]}$ values in nodes containing pairs of them. The message passing will not work if you keep the $\hat{y}^{[m]}$ nodes single, nor if you use groups larger than two. Can you figure out why?

(d) (Code:) Draw a number of samples from the posterior distribution $p(\boldsymbol{\Theta}|\mathcal{D})$ (e.g. using the `Factor` member function `sample`). Draw the sampled lines, the actual line and the data points on the same graph. Also draw the MAP and ML estimates, and compare them with the sampled lines of the full Bayesian approach.

(e) (Code:) Repeat the sampling of measurements and subsequent parameter estimation, but vary the number of measurements as well as the variance of the measurement noise. For which scenarios does the full Bayesian approach provide significantly more information about the parameters than ML and MAP estimation, and for which scenarios are there no significant differences?

Something to think of: In terms of Bayesian work, we (for simplicity) committed a small cheat. Can you spot where?