HW1 Solution sketch

Jonatan Kilhamn

February 2017

1 Problem 1.1

The multivariate spherical Gaussian:

$$P(X = x|\sigma^2 = s) = \frac{1}{(2\pi)^{\frac{p}{2}}\sigma^p} \exp\left(-\frac{(x-\mu)^\top(x-\mu)}{2\sigma^2}\right)$$

The likelihood of several independent observations:

$$\prod_{x \in D} P(X = x|\sigma^2 = s)$$

The maximum likelihood estimator:

$$\hat{\sigma}_{\text{MLE}} = \arg\max_s \prod_{x \in D} P(X = x|\sigma^2 = s)$$

which, through steps not shown here, becomes

$$\hat{\sigma}^2_{\text{MLE}} = \frac{\sum_{x \in D}(x-\mu)^\top(x-\mu)}{pn}$$

where $n = |D|$, the number of observations.

2 Problem 1.2

2.1 1.2 a)

The posterior distribution $P(\sigma^2 = s|D)$ is proportional to the product of the likelihood and the prior distribution:

$$P(\sigma^2 = s|D) \propto P(D|\sigma^2 = s)P(\sigma^2 = s)$$

The likelihood is again the product of the likelihood of $X = x$ for n data points. The prior distribution is Inv-Gamma(α, β). We know that the inverse gamma distribution is a conjugate prior to the spherical Gaussian, with posterior hyperparameters

$$\alpha_{\text{post}} = \alpha + n$$
$$\beta_{\text{post}} = \beta + \frac{1}{2} \sum_{x \in D} (x - \mu)^\top (x - \mu)$$

This can be derived by performing the multiplication above and discounting all constant terms (we only care about proportionality, not equality).

2.2 1.2 b)

The Bayes factor is the ratio between the likelihoods for the two models:

$$BF_{A,B} = \frac{P(D|M_A)}{P(D|M_B)}$$

We know $P(D|\sigma^2 = s)$, i.e. the likelihood given a particular value for $sigma^2$. However, each model only gives a distribution over σ^2. In order to find the likelihood given the entire model, we must integrate out s:

$$P(D|M_A) = \int P(D|\sigma^2 = s)P(\sigma^2 = s|M_A)\, ds$$

The quantities inside this integral is known, since $P(\sigma^2 = s|M_A)$ is Model A's prior inverse Gamma distribution (analogously for M_B).

2.3 1.2 c)

Before, we said that each model only gave a distribution over σ^2 rather than a single value. Now we are allowed to assume that each model consists of one value, specifically the MAP value, using the posterior distribution from 1.2 a).

The MAP value is

$$\hat{\sigma}^2_{\text{MAP}} = \frac{\beta_{\text{post}}}{\alpha_{\text{post}} + 1}$$

We then get the Bayes Factor through the likelihoods:

$$BF_{A,B} = \frac{P(D|M_A)}{P(D|M_B)} = \frac{P(D|\sigma^2 = \hat{\sigma}^2_{\text{MAP},A})}{P(D|\sigma^2 = \hat{\sigma}^2_{\text{MAP},B})}$$

Where each likelihood is a product of the likelihood of each data point, re-written into a sum in the exponential:

$$P(D|\sigma^2 = s) = \frac{1}{(2\pi)^{\frac{np}{2}}\sigma^{np}} \exp \sum_{x \in D} \left(-\frac{(x-\mu)^\top (x-\mu)}{2\sigma^2} \right)$$

3 2.1

Finding the MLE of μ once again consists of maximising the likelihood function:

$$\hat{\mu}_{\text{MLE}} = \arg\max_m \prod_{x \in D} P(X = x|\mu = m, \sigma^2 = s)$$

which, through steps not shown here, becomes

$$\hat{\mu}_{\text{MLE}} = \frac{\sum_{x \in D} x}{n}$$

i.e. the empirical mean.

4 Coding

A correct solution to the coding exercise is not provided at this time.

To clarify the hint about the "log-sum-exp trick", it refers to the fact that implementing the inverse Gamma distribution according to the definition will lead to problems with MATLAB's `gamma` function. When given too large arguments, it returns `inf`, which yields errors down the line. However, the `lngamma`, which implements the function $\log(\Gamma(x))$, can handle much larger arguments. So the problem can be circumvented by encoding the logarithm of the inverse Gamma distribution, which will be a sum of terms rather than a product, and then taking the exponential function to retrieve the distribution itself.

Comments

This is not a complete solution; only a sketch. If students were to hand in a report with as many steps omitted as this one, they would not get full marks.

The purpose of this document is to show students whether their solution was along the right lines. Firstly, this allows them to improve their understanding, if they had not solved the problem correctly; secondly, it gives a hint as to whether they can expect to score points for each task. (Of course, eventually each submission will be graded, and the actual number of points reported to the student.)

4.1 Jonatan's grading policy

If a student submission (a) explains its reasoning and (b) is correct, it will receive full marks.

A correct answer presented poorly can be grounds for deductions, but never for more than half of the point total on the task.

If the answer is incorrect, at least 1 point will be deducted. I will try to follow the reasoning and find where the mistake occurred. Depending on how severe the mistake was, and how good the reasoning was before and after the mistake(s), the deduction might be only the 1 point, or anything up to all the points for that task.

For tasks with more than one sub-problem (a, b, c...) my policy is that a correct answer to one sub-problem guarantees at least 1 point, but an incorrect answer to one of them rules out getting the maximum points for the task. In between those two extremes, each sub-problem does not correspond to an exact number of points; rather, the solution is considered as a whole.

I do not concern myself very much with coding style. If (1) the answers in the report are correct, (2) the code works when I run it, and (3) I don't have reason to suspect that code has been copied, then the quality of the code won't further affect the grade.

If there is a mistake or incorrect answer, clear code might make it easy for me to realise that it was only a typo, and deduct fewer points.

If the answers are correct but I can't see how the code provides them (e.g. it won't run), I will deduct up to half of the task's points.