FK7061 Statistical Methods in Physics pueh-leng.tan@fysik.su.se
Computer Exercise 3 jens.jasche@fysik.su.se

1 Programming preliminaries

1.1 Performing linear algebra calculations with python

To perform linear algebra computations in python you can use the the numpy linear algebra library. The library contains routines to solve linear systems of the type $\sum_j A_{ij}\, x_j = y_i$, invert matrices A_{ij} (defined as a list of lists or numpy array of numpy arrays) and estimate determinants.

```
>>> import numpy as np
>>> A = [[a_00, a_01, ..., a_0m],
         [a_10, a_11, ..., a_1m],
         .
         .
         .
         [a_n0,a_n1,..., a_nm]]
>>> A_inv = np.linalg.inv(A)
>>> detA = np.linalg.det(A)
```

Applying a matrix to a vector $c_i = \sum_j A_{ij}\, b_j$ can be achieved by:

```
>>> c = A.dot(b)
```

While in order to make a diagonal matrix out of a one-dimensional vector (a δ_{ij}), it is possible to write:

```
>>> c_matrix = np.diag(c)
```

2 Lab task

This task follows the description of Bayesian regression and model testing in Lecture 5.

In this lab task, we are going to address a Bayesian polynomial regression and model comparison task. Suppose, a detector reports time ordered measurements of stellar luminosities recorded at every day of the last year. This data consists of observed luminosities L_i recorded at times t_i and the corresponding Gaussian errors ϵ_i, being the standard deviations. We are now searching for a time-varying signal

$$s(t_i) = \sum_{l=0}^{M} \alpha_l\, t_i^l \tag{1}$$

which is a polynomial of unknown order M and coefficients α_l. You may assume that you have prior knowledge on the coefficients α_l specified by a mean μ_l^α and uncertainty $\left(\sigma_l^{\alpha_l}\right)^2$. The principle of Maximum entropy then states that the least informative prior for α_l, given that information, is a multivariate normal distribution. As demonstrated in lecture 5 the resulting posterior distribution is a multivariate normal distribution for the coefficients α_l. The following tasks will consist in inferring the parameters α_l and perform a Bayesian model test to identify the polynomial order M that explains the data best according to Ockhams' razor.

You will be provided numpy with the data in zip/numpy files labelled `data1.npz` - `data5.npz`. They contain the time series of the same luminosity measurements but with different levels of noise. To load the data files,

```
>>> import numpy as np
>>> data = np.load('data1.npz')
>>> luminosities = data['d']
>>> errors = data['e']
>>> times = data['t']
```

The array `data['d']` contains the time series of luminosity measurements, the array `data['e']` corresponding Gaussian errors of each measurement, The array `data['t']` contains the time at which each luminosity measurement was made.

2.1 Bayesian parameter inference

In lecture 5 we demonstrated that the posterior distribution for the case of polynomial regression is a multivariate normal distribution specified by a mean and a covariance matrix.

- Using at first the data in file `data3.npz`, for the nine different models of polynomial order $M = 0, 1, 2, 3, 4, 5, 6, 7, 8$ find the Maximum A Posteriori values, that is the optimal values for the polynomial coefficients α_l. Note, that for a Gaussian distribution the MAP values are equal to the mean. Solving for the MAP values therefore is a problem of solving a set of linear equations.

$$\mu_l^P = \sum_m^M D_{lm}^P A_m \tag{2}$$

  A_m is a convenient grouping of terms,

$$A_m = \frac{\mu_m^\alpha}{(\sigma_m^\alpha)^2} + \sum_i^N \frac{t_i^m \, d_i}{\sigma_i^2} \,, \tag{3}$$

  where N is the number of measurements, σ_i is the error on each measurement, t_i^m is the time of the i-th measurement raised to the m-th power, d_i is the i-th measured luminosity, μ_m^α is the mean of the prior, $(\sigma_m^\alpha)^2$ is the variance of the prior. The α here is an index, not a power.
  D_{lm}^P is the posterior covariance matrix which can be obtained by inverting the following

$$\left(D_{lm}^P\right)^{-1} = \frac{\delta_{lm}^K}{(\sigma_l^\alpha)^2} + \sum_i^N \frac{t_i^l \, t_i^m}{\sigma_i^2} \tag{4}$$

  δ_{lm}^K is the Kronecker delta, t_i^l, t_i^m are the time of the i-th measurement raised to the l-th and m-th power respectively.

- Once the MAP values for the 9 polynomial models have been found make predictions for the time varying signal $s(t_i)$ and compare the results with the observations.

2.2 Bayesian Model testing

The previous task aimed to infer optimal parameters for models of different polynomial order M. Now we need to search for the model that explains the observations best according to Ockhams razor, or Bayesian parsimony. To achieve this, compare the Bayes factors, BF, of the nine different models of polynomial order $M = 0, 1, 2, 3, 4, 5, 6, 7, 8$. As demonstrated in lecture

5, for the Gaussian regression task it is possible to analytically estimate evidences.

$$\begin{aligned}\pi(\{d_i\}|M) &= \int d\{\alpha_l\}\pi\left(\{\alpha_l\}|M\right)\pi\left(\{d_i\}|\{\alpha_l\},M\right)\\ &= \sqrt{\det(2\pi D^P)}\\ &\times \exp\left(\frac{1}{2}\sum_{lm}\mu_l^P(D_{lm}^P)^{-1}\mu_m^P\right)\\ &\times \prod_l \frac{\exp\left(-\frac{1}{2}\left(\frac{\mu_l^\alpha}{\sigma_l^\alpha}\right)^2\right)}{\sqrt{2\pi(\sigma_l^\alpha)^2}}\\ &\times \prod_i \frac{\exp\left(-\frac{1}{2}\left(\frac{d_i}{\sigma_i}\right)^2\right)}{\sqrt{2\pi(\sigma_i)^2}}\end{aligned} \tag{5}$$

- Estimate the Bayes factors for the 9 models and determine the polynomial order M that explains the data best.

2.3 Frequentist goodness of fit

- Can you perform a goodness of fit test of your choice (χ^2, K-S, maximum likelihood ratio ...) for the models obtained by fixing the polynomial parameters to the MAP values (say for $M = 1, 3, 5, 6, 7$) and say which hypothesis is preferred? Does it agree with the Bayesian result? Try to think of an explanation.

2.4 *[Optional]* Different noise levels

Redo the previously described analyses with different data sets. All data sets contain the same true underlying polynomial signal.

- What do you observe when using data sets with decreasing noise levels? How can you explain this observation?