Computational Biology I: Quantitative Data Analysis
Frühjahrsemester 2019

Exercise 5

Athos Fiori and Pascal Grobecker

Student t-distribution

The goal of today exercise is to infer, using micro array technology, which genes are up/down-regulated if the cell experiences some new condition. Imagine we measure gene expression in a certain condition, called reference, which we consider as our "zero" measure. When the cell experiences some new condition, called perturbed, gene expression can change and we can measure it using microarray technology. The change in expression for any particular gene is usually summarised by its log fold change:

$$x = \log_2(I_{\text{perturbation}}) - \log_2(I_{\text{reference}}),$$

where I is the measured intensity. Let's focus on a specific gene and measure it's fold change x for any of the n replicates. Thus, our dataset consists in the fold change $(x_1, ..., x_n)$ of some specific gene in any of the n replicates. We can assume that the fold change μ in expression of a gene is the same in all n replicates and that the measurement error is Gaussian distributed with the same standard deviation σ i.e.

$$x_i = \mu + \epsilon \tag{1}$$

$$\epsilon \sim \mathcal{N}\left(0, \sigma^2\right) \tag{2}$$

where the first equation means that the measured fold change for the i^{th} replicate (x_i) equals to μ plus some error ϵ which is Gaussian distributed with mean 0 and variance σ^2, equation (2). Thus the distribution of x_i is simply $\mathcal{N}\left(x_i - \mu, \sigma^2\right)$ and the total likelihood reads, since the x_i are independent,

$$P(x_1, ..., x_n|\mu, \sigma) = \prod_{i=1}^{n} \underbrace{P(x_i|\mu, \sigma)}_{\mathcal{N}(x_i - \mu, \sigma^2)} = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{n}{2\sigma^2}\left((\bar{x} - \mu)^2 + \text{var}(x)\right)\right] \tag{3}$$

where $\bar{x} = \frac{1}{n}\sum_i^n x_i$ and $\text{var}(x) = \frac{1}{n}\sum_i x_i^2 - (\bar{x})^2$. This likelihood contains the two unknown parameters μ and σ but we are only interested in the mean fold change μ. We now have 2 ways to proceed to get rid of the σ parameter. The first, which is the more proper and formal, is to marginalise over σ, i.e. integrate over σ

$$P(x_1, ..., x_n|\mu) = \int P(x_1, ..., x_n|\mu, \sigma) P(\sigma|I) d\sigma \tag{4}$$

$$\propto \left(1 + \frac{(\bar{x} - \mu)^2}{\text{var}(x)}\right)^{-(n-1)/2}. \tag{5}$$

which is called Student t-distribution. The second way, which works only if the dataset is large enough, consist in estimating the unknown variance σ^2 from the observed variance $\text{var}(x)$ of our sample. In the following we'll use both ways to proceed. The goal is to find which of the genes in the given data set is significantly up-regulated in the perturbed environment. To this end we have to determine the posterior probability $P(\mu|x_1, ..., x_n)$ and then check how likely is that $\mu > 0$ i.e. that the gene is up-regulated.

1. Complete the functions `gaussian_likelihood` and `student_likelihood` which implement the likelihoods respectively given in Eqn.(3) and Eqn.(5).
   Tip: Remember that the numpy module has many build-in functions.

2. Next, we are going to calculate the posterior probability $P(\mu|x_1, ...x_n)$ assuming **(a)** the gaussian-likelihood with $\sigma^2 = \text{var}(x)$ and **(b)** the Student-t likelihood . In order to derive the posterior probabilty we assume a uniform prior over μ and, using Bayes theorem, we find

$$P(\mu|x_1, ..., x_n) = \frac{P(x_1, ..., x_n|\mu)}{\int_{-\infty}^{\infty} P(x_1, ..., x_n|\mu) d\mu}$$

(a) For the Gaussian case the integral in the denominator

$$\int_{-\infty}^{\infty} \frac{\exp\left(-\frac{n}{2\sigma^2}\left((\bar{x}-\mu)^2+\mathrm{var}(x)\right)\right)}{\sigma^n(2\pi)^{n/2}}d\mu$$

can be easily work out analytically using an appropriate change of variable and knowing $\int e^{-ax^2}dx = \sqrt{\frac{\pi}{a}}$

(b) For the Student-t case the analytical form is much more complicated to work out. Therefore we'll solve it numerically using the `quad` function of the `scipy.integrate` module. Have a look at its documentation! The `quad` function takes as its first argument the name of the integrant function. By convention, the first argument of the integrant function must be the integration variable (in our case it's μ). The other necessary arguments are, the lower and the upper integration boundaries (i.e. plus and minus infinity) and a tuple containing all additional arguments passed to the integrant function. The output is a tuple containing the result of the integral and an estimate of the absolute error of the integration result which we will neglect for now.

Implement your results in the functions `gaussian_posterior` and `student_posterior`.

3. Complete the function `expectedFoldChange` which returns the mean and the 95% probability interval of the posterior probability distribution of μ using the Student-t posterior (when the parameter `dist=0`) and the Gaussian approximation (`dist=1`). In order to calculate the integral over the continuous posterior distribution and the mean, an easy way is to use the following approximation

$$\int_{\mu_l}^{\mu_u} P(\mu|x)d\mu \approx \sum_{i=1}^{K} P(\mu_k|x)\Delta\mu \tag{6}$$

$$\int_{\mu_l}^{\mu_u} \mu P(\mu|x)d\mu \approx \sum_{i=1}^{K} \mu_k P(\mu_k|x)\Delta\mu \tag{7}$$

where $\mu_1 = \mu_l,\ \mu_K = \mu_u$ and the number of steps K have to be chosen such that $\Delta\mu = \mu_{i+1} - \mu_i$ is reasonably small. This is called Riemann sum and it's often used for numerical integration.

4. Imagine you have measured five genes of interest with different number of replicates each and you obtained the following fold changes:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Gene 1	-0.5989	0.9163	-1.1192	-	-	-	-	-	-	-
Gene 2	2.6043	2.4013	2.8432	1.9412	0.298	-	-	-	-	-
Gene 3	2.9973	4.5676	1.8934	-1.1978	1.7192	4.0529	0.325	-1.9837	4.9612	-1.2523
Gene 4	-2.729	5.9134	-2.9845	-	-	-	-	-	-	-
Gene 5	1.9134	0.015	-	-	-	-	-	-	-	-

Plot the posterior probability for each gene, using the Gaussian and Student-t posteriors. Try to understand how the number of available replicates affect the Student-t distribution and its Gaussian approximation. Specifically, do they differ in the mean? Which of the two gives the tighter 95% probability interval? When do they agree? For this part, you don't have to submit anything but questions would be asked during the exercise session.
NB: This data set is already implemented in your python script.

5. Complete the function `positiveFoldChange` which returns the posterior probability of $\mu >= 0$ using either the Student-t posterior distribution (`dist=0`) or the Gaussian approximation (`dist=1`). For which of the five genes you are at least 90% sure that it is upregulated? Compare the two measures.

Theoretical questions

1. Let t the life-time of a protein and assume that we are given the information that t follows a simple exponential distribution $P(t|\tau) = \frac{1}{\tau}e^{-t/\tau}$, with $t \geq 0$. What's the median t_m of this distribution?

2. For the same distribution $P(t|\tau)$, as in the previous exercises, find the upper bound t_{up} such that the probability of t being larger than t_{up} is only ϵ, i.e. $P(t > t_{up}) = \epsilon$. What is the solution of t_{up} in terms of τ and ϵ?

3. Assume the quantity x is uniformly distributed on the range [0,L], i.e. $P(x|L) = \frac{1}{L}$ for $0 \leq x \leq L$, and zero everywhere else. We now obtain a data-set D of n samples from this probability distribution $D = (x_1, x_2, ..., x_n)$, where we have sorted the values by size such that $x_n > x_{n-1} > ... > x_2 > x_1$. We now want to derive the posterior distribution $P(L|D)$ for the parameter L given our data. Assuming a uniform prior $P(L)$=constant, show that

$$P(L|D) = (n-1)\frac{x_n^{n-1}\Theta(L - x_n)}{L^n}$$

4. We toss a coin 8 times and count the number of heads. Our first seven tosses give tails and the last toss a heads, i.e. we get 1 heads in our 8 tosses. We now want to calculate the p-value for the hypothesis that the coin is *unbiased*, i.e. equally likely to be heads or tails in each toss. What's the p-value to get a number of heads that is so low under this hypothesis?

5. We now do a different experiment. Instead of deciding to throw the coin 8 times and count the number of heads we got, we decided to throw the coin until we get heads for the first time. Interestingly, we got exactly the same data, i.e. the first 7 tosses were tails and the $8th$ toss was heads. We now again want to calculate a p-value for the hypothesis that the coin is *unbiased*. We now need to calculate: If I start throwing an unbiased coin, what's the probability that I throw 7 or more tails before I throw the first heads?
   Extra Both the data and the question, i.e. "is the coin biased?", were the same for this and the previous question. Can you compare the results we got an give comments? Is there something that surprises you? Can you explain the result?

How to submit the exercise

1. Check that your functions are working properly before submitting them!
2. Don't change the name of the file or the name of any function within the file!
3. Submit your exercise by email to `biocomp1-bioz@unibas.ch`
4. As the email subject use: first name, last name
5. Don't forget to attach your completed `exercise5.py`!
6. The submission dealine is Thursday, April 4th, midnight.