Intensive course of Bayesian methods: aims, methods and requirements.

by S. Andreon, stefano.andreon@brera.inaf.it

The course consists in one week-long, 3 h a day, laboratory on Bayesian (statistical) methods. It will not assume (almost) any previous knowledge in statistics. It is, by large, a laboratory course (i.e. attendees will do something, not just hear someone), and, for this reason, assumes a familiarity with the use of a plotting environment (at the attendee choice).

Course Syllabus:

- Probability axioms. Computation of the posterior: analytical vs numerical sampling. Upper limits. Initial discussion on the role of the prior. Importance of checking numerical convergence. A glimpse on sensitivity analysis.
- Single parameters models. Combining information coming from multiple data. The prior (and the Malmquist-like effect). Prior sensitivity. Two-parameters models. Joint probability contours. Comparison of the performances of state-of-the-art methods to measure a dispersion.
- Introduction to regression. Comparison of regression fitters. Regressions (of increasing difficulty): non-linear regression with non-gaussian errors of different sizes (but no error on predictor and no intrinsic scatter). Allowing systematics (intrinsic scatter). Allowing errors on x. Regressions with two (or more) predictors. A glimpse on other important issues such as mixture of regressions, non-random data collection, model checking.
- Elements of model selection.

The preliminary program (based on the one given in 2017) is at this URL[1].

Rationale and Methods:

The purpose of the course is not to teach a content. Attendees will hear the instructor for a tiny fraction of the time, and spend most of their time solving by themselves (with the teacher help) problems of increasing (statistical) complexity, often using real data (to be downloaded during the course). This poses constraints on requirements, and demands that the attendees attend *all* lectures.

Requirements for organisers.

Internet connection for all people (including teacher) and electrical power (and sockets) for all computers.

[1] http://www.brera.mi.astro.it/∼andreon/corso_metodi_bayesiani/CorsoMetodiBayesiani1718.html . If you experience difficulties in reaching this and other links, do not cut&paste them. Instead, type them from scratch.

Requirements for attendees.

Since attendees will solve by themselves problems using some software, theyare requested to install a few software packages and to write some reading/plotting routines prior to the beginning of the course. Failing to do so will significantly impair the learning outcomes, because there will be no time during the course to write them from scratch.

Attendees should:

1. have their own computer (with an internet connection), and be acquainted with it;
2. have installed JAGS[2] which in turn may demand the installation of some additional libraries; it may be useful to browse the table of distributions (e.g., page 30 of version 2 of the user manual) to refresh your memory about the mathematical expression of some famous functions.
3. be able to make plots and simple data manipulation using their preferred environment. In particular, attendees should have already written routines for:
   - properly read data when they are in the standard JAGS input file format, illustrated at this URL[3]. Each quantity, whose name is declared just before the string '$<-$', is followed by its values (separated by commas). The script should work independently of the number of variables and the number of values per variable.
   - properly reading files in the CODA format (standard JAGS output file): CODAindex.txt[4] describes the content of the CODAchain1.txt[5] file by listing the variable names, where they start and where they end. For example (inspect CODAindex.txt) the variable s starts at line 1 and ends at 50,000 (and it is on the 2nd column) of CODAchain1.txt. The reading routine should work for any number of variables (e.g. 10) and of samples (e.g. 30,000).
   - compute mean and standard deviation (check that s has mean 28.5 and standard deviation 16.5)
   - compute the shortest interval including x % of the samples (check that the 95% interval of s is [0,56]). To compute it, you may, for example, start from the peak of the pdf and move down until the interval includes x% of the samples.
   - produce a trace plot, i.e., a plot that gives the variable value as a function of its rank (or step in the chain, listed as first column in the CODAchain file), as in Fig. 1. This plotting routine should work also if CODAindex.txt contains, say, 10 variables.
   - produce normalized histograms, as in Fig. 1, right panel (note that the integral must be unity and independently of bin size; test it by changing the bin size).
   - draw contours. The routine should work for non-elliptical contours, for example when one has two separate "islands". The contours should include about 68% and about 95% of the samplings. A small margin of error is allowed (i.e. 70% in place of 68% is fine). It is instead not allowed to draw contours at pre-defined thresholds (e.g, taking the peak value and dividing by a "magic number"). Check your contours against those in Fig. 2 with the sampling in CODAchain1.txt. The latter contours are somewhat approximated (and nothing better than this is required!).

[2] https://sourceforge.net/projects/mcmc-jags/
[3] http://www.brera.mi.astro.it/%7Eandreon/BayesianMethodsForThePhysicalSciences/data9.1.2.dat.R
[4] http://www.brera.mi.astro.it/∼andreon/corso_metodi_bayesiani/CODAindex.txt
[5] http://www.brera.mi.astro.it/∼andreon/corso_metodi_bayesiani/CODAchain1.txt

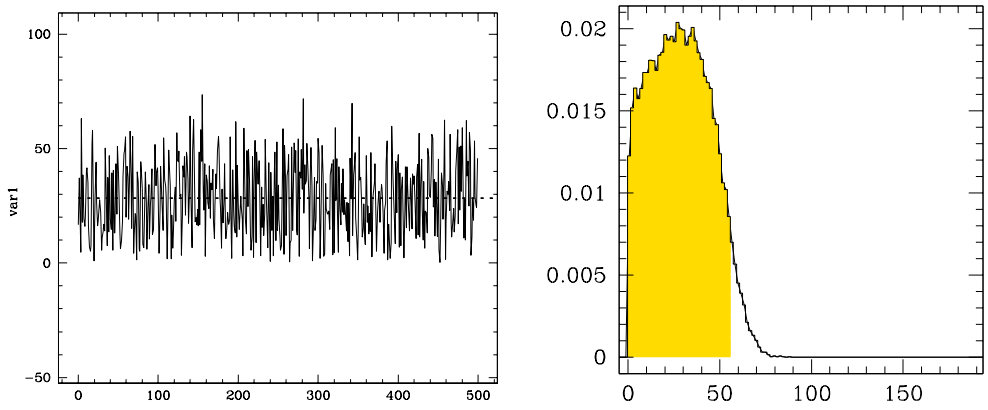

Figure 1: Left panel: Trace plot. Right panel: Marginal distribution (histogram)

- compute the mean y for each (small) bin of x, plot it at the mean x of the bin, using the sample generated by JAGS available at this URL[6]. Its CODAindex is here[7]. The found points should be roughly aligned on the line $y = x/2$. When done, make the reverse: compute instead the mean x per each (small) bin of y and plot mean values. The found points should be roughly aligned on the line $y = 2x$. Figure 3 is a (nicer) illustration of the idea (the red lines connect the computed points). You are not asked to exactly reproduce this figure, plotting the means (and checking them against the lines) suffices.

N.B. If you use Python or R, you got for free much of the above (and much more, e.g., corner.py and pyjags), but this may imply to learn a new plotting language.

Attendees should came with these routines working on their computer and with some ability to slightly modify them when needed, because this course, on statistical methods, has no time to address non-statistical issues. Past experience tell that attendees able to write the above routines, but coming without them, failed to attend the lectures (spent all time in writing the routines). Therefore, attending the lectures without these working routines is, by large, a loss of time.

All non-graduate students willing to attend the course should come with their version of figures 1, 2, and 3 printed on paper and give them to the instructor at the start of the course. Lacking to fullfill this requirement, they will not admitted to the course.

[6] http://www.brera.mi.astro.it/%7Eandreon/corso_metodi_bayesiani/CODAchain_fakesampleregr.txt
[7] http://www.brera.mi.astro.it/%7Eandreon/corso_metodi_bayesiani/CODAindex_fakesampleregr.txt

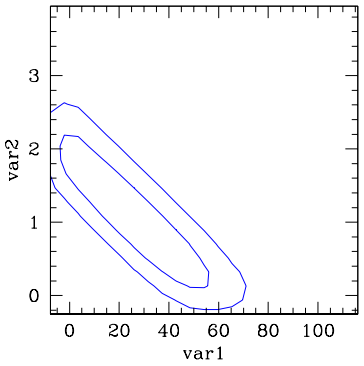

Figure 2: Contours. In this plot, contours are imprecisely determined close to $var1 = 0$ e $var2 = 0$ boundaries, and not corrected for smoothing effects. You are not asked to do better (but nothing precludes you from doing it).

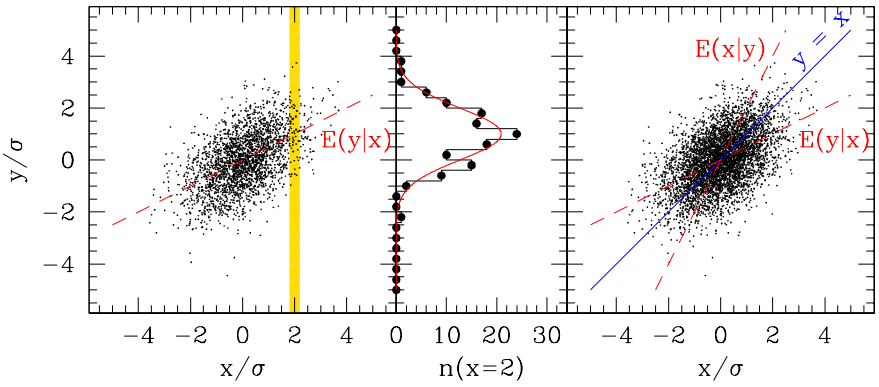

Figure 3: Left panel: 500 points drawn from a bivariate Gaussian, overlaid by the line showing the expected value of y given x. The yellow vertical stripe captures those y for which x is close to 2. Central panel: Distribution of the y values for x values in a narrow band of x centred on 2, as shaded in the left panel. Right panel: as the left panel, but we also add the lines joining the expected x values at a given y, and the $x = y$ line.

Checking JAGS istallation

In order to check having properly installed JAGS, first, save the file below as model.bug and make you sure that it only contain ASCII

```
# Bayesian Methods for the Physical Science. Learning from
# Examples in Astronomy and Physics. By S. Andreon and B. Weaver.
model {
for (i in 1:length(nrec)) {
 nrec[i] ~ dbin(eff[i],ninj[i])
 nrec.rep[i] ~ dbin(eff[i],ninj[i])
 eff[i] <- A + (B-A)*phi((E[i]-mu)/sigma)
}
A~dunif(0,1)
B~dunif(0,1)
mu~dunif(0,100)
sigma~dunif(0,100)
}
```

Second, save the file below as model.cmd and check that it only contain ASCII

```
model in model.bug
data in data.dat.R
compile,nchains(1)
initialize
update 3000
monitor set A, thin(10)
monitor set B, thin(10)
monitor set mu, thin(10)
update 100000
coda *
data to testata
samplers to testsamplers
exit
```

Now, using the data listed at this URL[8] (save the file as data.dat.R), run JAGS redirecting the standard input from the file model.cmd. Under linux the command to execute is

```
jags < model.cmd
```

provided that the jags executable is in the path. If a file CODAchain1.txt is produced, then JAGS has been properly installed.

[8] http://www.brera.mi.astro.it/%7Eandreon/BayesianMethodsForThePhysicalSciences/data8.2.dat.R

Examination (in the case this applies)

Attendees registered for examination will be asked to write a relation describing how they solved a statistical problem of their own choice similar to, but different from, those addressed during the course. Each attendee must solve a problem different from those chosen by other attendees. Attendees have to prepare a short relation (consider a soft limit of about 4 pages) with:

- Title
- Your name
- A few line of background (why the analysis of these data are interesting). A minimal background information about topic to understand what follows. Identify the problem you will address.
- Present data: plot data; list them if just a few, otherwise only plot them.
- Which errors? (here you are describing the likelihood: gaussian? Poisson?, a combination of, etc.).
- Description of the model: describe link between quantities (if any). With extra scatter? Describe priors used. Conclude with "therefore the JAGS model reads" followed by the JAGS model.
- Check convergence. Are results independent of chains? Check trace plots, insert one (and mention the other being similar).
- Plot posterior parameter distributions for the key parameters. If more than one parameter: plot joint probability contours. Overplot the prior. Optional: shade the 68% or 90% interval on individual parameters.
- Quote in the text (or in a table) posterior mean and standard deviation (or other summaries of the posterior distribution).
- Plot the fitted model on the data, with 68% bounds (when this applies).
- If you made a y vs x fit, write down the math expression with inserted values (and errors) for the parameters.
- Comment about prior sensitivity: are your parameters strongly affected by the prior choice? Optional: adopt a sligtly different prior to prove little sensitivity (the mean/stddev should not change).
- Conclusion: so what?

A template relation (only useful for understanding the amount of blurb to be put in there) is given at the URL[9]

Attendees should complete their evaluation within 4 weeks after the end of the course, and can submit their document for evaluation at most twice. The second submission, if any, should account for all comments given to the first submission.

[9] http://www.brera.mi.astro.it/%7Eandreon/corso_metodi_bayesiani/template_relation.pdf