Mixed-effects inference for classification studies

R package v1.02 (r19176), March 2013

Introduction

Classification algorithms are often used in a hierarchical setting, where a classifier is trained and tested on individual datasets which are themselves sampled from a group. Examples of this sort of analysis are ubiquitous and are common in domains as varied as spam detection, brain-machine interfaces, and neuroimaging.

This R package provides answers to the questions of statistical inference that arise in all of these settings. It implements models that account for both within-subjects (fixed-effects) and between-subjects (random-effects) variance components and thus provide mixed-effects inference.

The package is extremely easy to use and requires no prerequisites other than R.

Literature

For details on the theoretical foundation, practical applications, and advantages over alternative methods, see:

- K.H. Brodersen, J. Daunizeau, C. Mathys, J.R. Chumbley, J.M. Buhmann, & K.E. Stephan (2013). Variational Bayesian mixed-effects inference for classification studies. *NeuroImage* (in press). doi:10.1016/j.neuroimage.2013.03.008.
- K.H. Brodersen, C. Mathys, J.R. Chumbley, J. Daunizeau, C.S. Ong, J.M. Buhmann, & K.E. Stephan (2012). Mixed-effects inference on classification performance in hierarchical data sets. *Journal of Machine Learning Research*, 13, 3133-3176.
- K.H. Brodersen, C.S. Ong, J.M. Buhmann, & K.E. Stephan (2010). The balanced accuracy and its posterior distribution. *ICPR*, 3121-3124.

Installation

To install and load the package, type the following commands into an R session:

```
> install.packages("/path/to/package/micp_1.0.tar.gz",
  repos=NULL, type="source")
> library(micp)
```

Example 1 – inference on the accuracy

Consider a situation in which a classification algorithm (e.g., a support vector machine or a logistic regression model) has been trained and tested to predict the binary label (+1 or -1) of a set of trials. Further, assume the analysis has been carried out independently for each subject within a group. The results can then be summarized in terms of two vectors: The first one, k, encodes the number of correctly classified trials in each subject; the second, n, encodes the total number of trials in each subject. The following steps outline how to apply the R package to this setting.

Step 1: note down observed classification outcomes

We begin by specifying two vectors that fully describe the observed outcomes of our classification analysis:

```
> ks <- c(82,  75,  92,  85,  88)
> ns <- c(100, 100, 100, 100, 100)
```

This says, for example, that 82 out of 100 trials were classified correctly in the first subject. There are 5 subjects in total in this example.

Step 2: inference

We perform inference by typing:

```
> stats <- micp.stats(ks, ns)
```

The above code performs full Bayesian inference using an efficient variational Bayes algorithm. The acronym in micp.stats() is short for **m**ixed-effects **i**nference on **c**lassification **p**erformance. We can obtain a summary of the results using:

```
> micp.stats(ks, ns)
Variational Bayesian mixed-effects inference on classification
accuracy

Population inference
  posterior mean accuracy:    0.82 (p = 0)
  posterior 95% interval:     [0.72, 0.9]

Subject-specific inference
  posterior logit means:      1.52, 1.14, 2.27, 1.71, 1.93
  posterior logit precisions: 16.62, 20.25, 10.39, 14.86, 13
```

```
Bayesian model comparison
  free energy F: -20.28
```

This tells us, for example, that the population mean accuracy was 82%, with a 95% central credible interval of 72% … 90%. This is the interval in which we place 95% of our posterior belief, and we could use it for plotting error bars on the classification performance. The output also contains an approximation to the free energy F, which is a lower bound to the marginal likelihood (or model evidence) and can be used to compare the current model to future alternatives.

We can inspect the function output in more detail using:

```
> stats <- micp.stats(ks, ns)
> names(stats)
[1] "mu"    "p"     "ci"    "q"     "model"
> stats$p
[1] 2.755991e-07
```

With an infraliminal probability of $p \approx 2.8 \times 10^{-7}$, we are supremely confident that the classifier operated above chance at the group level. Put differently, the fact that p is approximately 0 means that we are approximately 100% sure that the population mean accuracy is above chance.

To display all details about the function micp.stats(), type:

```
> ?micp
```

Example 2 – inference on the balanced accuracy

In many real-world problems, the data used for classification are not perfectly balanced. This means that there are more examples from one class then from the other. Denoting the two classes as the *positive* and the *negative* class, respectively, there might for instance be more positive than negative examples in the data. When the data are imbalanced, the accuracy is a misleading performance measure and should be replaced by the *balanced accuracy*.

To infer on the balanced accuracy, the software needs to know how many positive and negative trials were classified correctly (rather than just an overall number of correctly classified trials, as was sufficient in Example 1).

Step 1: note down observed class-specific classification outcomes

We begin by noting down how many trials were classified correctly in each subject. In contrast to Example 1, we are now providing this information separately for positive and negative examples. Thus, k and n are now matrices. The first row refers to positive examples, the second row to negative examples.

```
> ks <- rbind(c(40, 44, 18, 42, 44),
              c(48, 41, 65, 49, 32))
> ns <- rbind(c(45, 51, 20, 46, 48),
              c(55, 49, 80, 54, 32))
```

Here, we recorded that in the first subject, there were 45 examples with true label '+1', out of which 40 were classified correctly. 55 examples had a '-1' label, and 48 of these were classified correctly. Note that in the above example the last subject has fewer trials than the rest; mixed-effects inference will correctly account for this.

Step 2: inference

Inference is as straightforward as before. Since k and n are now matrices (as opposed to row vectors as in Example 1), the code automatically switches to an algorithm for inference on the *balanced* accuracy.

```
> micp.stats(ks, ns)
Variational Bayesian mixed-effects inference on the balanced
classification accuracy

Population inference
  posterior mean balanced accuracy:    0.86 (p = 0)
  posterior 95% interval:              [0.79, 0.91]

Subject-specific inference
  posterior balanced accuracy means:   0.87, 0.85, 0.84, 0.89, 0.92

Bayesian model comparison
  free energy F: -32.82
```

This tells us that the posterior mean of the population mean balanced accuracy is 86%. Is this better than chance? Yes, with a conviction of 1 – 0.000 = 100%. If we wanted to plot error bars, we would use the limits of the central 95% credible interval, which is [79%, 91%].

Software note

Kay H. Brodersen
ETH Zurich
Switzerland
brodersen@biomed.ee.ethz.ch