Using clustering to correct for multiple comparisons

June 10

2013

We present here the various functions needed to perform multiple comparisons correction with LIMO EGG

LIMO EEG toolbox: LInear Modeling of EEG data.

LIMO EEG toolbox: LInear Modeling of EEG data.
Using clustering to correct for multiple comparisons

Cyril Pernet & Guillaume Rousselet
cyril.pernet@ed.ac.uk
Guillaume.Rousselet@glasgow.ac.uk

Contents

Multiple Comparison Correction 3
The neighbourhood matrix 3
Clustering Data 4
Cluster mass 4
Clustering 4
TFCE (Threshold Free Cluster Enhancement) 5
References 5

Multiple Comparison Correction

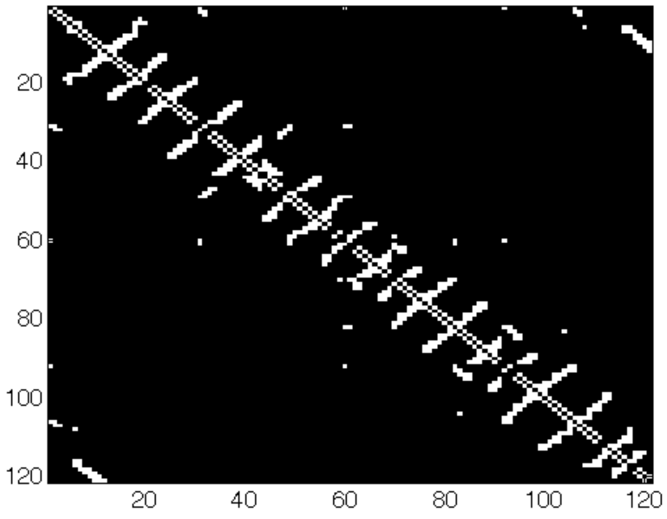
The type I error rate across tests increases with the number n of statistical tests. This family wise error rate, or FWER = 1 - (1 - alpha)^{n}. This formula is for n independent tests, and therefore the actual FWER is even higher for MEEG because data are correlated in space and time.

For a single test, the frequentist framework uses the distribution of t or F values under the null hypothesis to estimate the conditional probability p to observe an effect as large or larger than the one observed in the data, given that the null hypothesis is correct. LIMO EEG deals with multiple testing by estimating the null distribution of maximum statistics across the whole data space. For a t-test or an ANOVA, the null hypothesis is created by centering the distribution for each condition. Then the data are sampled with replacement, and for each bootstrap sample, t and F values are computed. To correct for multiple comparisons, one strategy consists in saving, for each bootstrap, the maximum t or F value across channels and time frames. After N bootstraps, the null distribution of this max statistics can be used to threshold the original results: for instance original F values could be considered significant if they are larger than the 95th percentile of the max bootstrapped F distribution. However, this max statistics approach tends to be conservative because it does not account for the correlation of the data in space and time. As alternatives, we can compute, under H0, the distribution of maximum cluster mass (Pernet et al., in prep) or the distribution of maximum TFCE scores (Smith & Nichols, 2009).

The neighbourhood matrix

To create clusters, we need to describe the relationship among observed data. Across time points, neighbours are simply successive data points. Across channels, we need to consider channels' positions to define a neighbourhood matrix.

Here is an example of a neighbourhood matrix:



This is an Ne x Ne matrix, where Ne is the number of electrodes. White cells indicate pairs of neighbouring electrodes; black cells indicate electrodes that are not neighbours.

A neighbouring matrix can be created in 3 ways:

(i) by calling the limo tools GUI and selecting ‘Create an expected chanloc files’.
(ii) by using limo_get_channeighbstructmat, with this syntax:
[neighbours,channeighbstructmat]=limo_get_channeighbstructmat(EEG,neighbourdist)
(iii) by assigning values directly to a channeighbstructmat matrix, and saving it to disk.

For options (i) and (ii), you need to input neighbourdist, the threshold distance that defines neighbour electrodes. For instance, 0.37 is a good distance for Biosemi standard 128 electrode configuration. However, that value could be very different for your own montage. You can check the accuracy of the neighbourhood matrix for your electrode montage by using the output neighbours from limo_get_channeighbstructmat: neighbours is a structure that defines for each channel a list of its neighbours, and in which a channel is not a neighbour of itself – for instance:

```
neighbours{1}.label = 'Fz';
neighbours{1}.neighblabel = {'Cz', 'F3', 'F3A', 'FzA', 'F4A', 'F4'};
neighbours{2}.label = 'Cz';
neighbours{2}.neighblabel = {'Fz', 'F4', 'RT', 'RTP', 'P4', 'Pz', 'P3', 'LTP', 'LT', 'F3'};
neighbours{3}.label = 'Pz';
neighbours{3}.neighblabel = {'Cz', 'P4', 'P4P', 'Oz', 'P3P', 'P3'};
and so on…
```

Finding an appropriate neighbourdist for your electrode montage will probably require several attempts. Ensure that you check all the electrodes to be sure their neighbours are correct. In some cases, and in particular for 32 or 64 electrode montages, it might be difficult to find an appropriate neighbourdist. You may have to decide subjectively to include or not certain neighbours. In such cases, you might also find useful to get as good a neighbourhood matrix as you can using options (i) or (ii), which you would then tweak manually to adjust the status of certain borderline electrodes (iii).

Clustering Data

Cluster mass

For a given set of statistical results, an uncorrected, univariate, threshold is applied (e.g. p<0.05) to create a thresholded map of clusters. Cluster mass corresponds, for a given cluster, to the sum of statistical values inside that cluster. Cluster mass thus reflects both the height (the strength of the statistical values) and the size (the number of statistical values) of the cluster.

Clustering

To threshold data, we need:
(i) maps of observed t or F values, and their corresponding p values;
(ii) a set of maps of t or F values, and their matching p values obtained under H0;

(iii) a neighbourhood matrix, which describes the relationship among channels.

For channels*time frames maps, the cluster mass is obtained using *limo_getclustersum*. For instance under H0 do something like

```
for boot=1:nboot
     boot_maxclustersum(boot) = limo_getclustersum(...);
end
```

This bootstrapped distribution is then sorted to obtain a threshold for a given alpha

```
U = round((1-alpha)*nboot); % bootstrap threshold
sort_boot_maxclustersum = sort(boot_maxclustersum,2);
maxclustersum_th = sort_boot_maxclustersum(U); % the threshold under H0
```

Finally, we threshold the observed map using *limocluster_test2*

```
[mask,cluster_p]= limo_cluster_test2(...);
```

The same technique can be applied to 1D data using *limo_ecluster_make* and *limo_ecluster_test*.

TFCE (Threshold Free Cluster Enhancement)

As explained above, thresholding based on clustering implies a cluster forming threshold: for instance a p value <= 0.05 is used as a threshold to define clusters in bootstrapped and in original. The TFCE technique instead integrates statistical values through 'all' cluster forming thresholds – in LIMO by discreet steps of 0.1 between some minimum and maximum values). A TFCE score is thus a statistical value (t or F) weighted by the strength of the cluster to which it belongs. belonging to a cluster and tThe weight depends on the extent and the height of this cluster. TFCE scores for observed data (3D) and data under H0 (4D) can be obtained using limo_tfce. Finally, a significance test can be obtained by comparing the observed TFCE scores to a percentile (e.g. 95th) of the bootstrapped maximum TFCE scores (across all electrodes and time frames). This strategy effectively controls for multiple comparisons.

References

Pernet, C. R., Chauveau, N., Gaspar, C., & Rousselet, G. A. (2011). LIMO EEG: A toolbox for hierarchical linear modeling of electroencephalographic data. Computational Intelligence and Neuroscience, 2011, 831409.

Pernet, C., Nichols, T.E., Latinus, M. & Rousselet, G.A. Cluster-based computational methods for mass univariate analysis of event-related brain potentials/fields. - in preparation

Smith, S.M. & Nichols, T.E. (2009) Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. NeuroImage, 44, 83-98.

For details of the clustering procedure applied to real data see also
Rousselet, G. A., Gaspar, C. M., Wieczorek, K. P., & Pernet, C. R. (2011). Modeling single-trial ERP reveals modulation of bottom-up face visual processing by top-down task constraints (in some subjects). Frontiers in Psychology, 2(137). doi:10.3389/fpsyg.2011.00137