P-value Weighting package for MATLAB

Edgar Dobriban*

April 3, 2015

Contents

1 Introduction

This MATLAB package contains open source implementations of several p-value weighting methods, including Spjotvoll, exponential and Bayes weights (proposed in Dobriban et al., 2015). These are methods for improving power in multiple testing via the use of prior informaton.

The code was tested on MATLAB R2014a and R2014b. Feel free to contact the author with suggestions and comments.

2 Quick usage example

To compute Bayes weights, call `w = bayes_weights(mu,sigma,q)`, where μ and σ are the prior estimates of the effect sizes and prior standard errors. Also `q` is the significance level at which we want to test each hypothesis.

3 P-value weighting methods implemented

For each p-value weighting method, we assume we observe data $T_i \sim \mathcal{N}(\mu_i, 1)$ and test each null hypothesis $H_i : \mu_i \geq 0$ against $\mu_i < 0$. The p-value for testing H_i is $P_i = \Phi(T_i)$, where Φ is the normal cumulative

*Department of Statistics, Stanford University, `dobriban@stanford.edu`

distribution function. For a weight vector $w \in [0, \infty)^J$ and significance level $q \in [0, 1]$, the weighted Bonferroni procedure rejects H_i if $P_i \leq qw_i$. Usual Bonferroni corresponds to $w_i = 1$.

Each p-value weighting method assumes some additional independent information about μ_i, and returns a weight vector w. These can then be used for weighted Bonferroni or other multiple testing procedures.

3.1 Bayes

Bayes p-value weights can be computed using: `[w, q_star, q_thresh, c] = bayes_weights(eta, sigma, q)`. The inputs specify the prior distribution of the means μ_i of the test statistics as:

$$\mu_i \sim \mathcal{N}(\eta_i, \sigma_i^2),\ i \in \{1, \dots, J\}$$

where:

- `eta`: a vector of length J, the estimated means of test statistics, derived from the prior data
- `sigma`: a strictly positive vector of length J, the estimated standard errors of test statistics, derived from the prior data
- `q`: The weights are optimal if each hypothesis is tested at level q. For instance, if we want to control the FWER globally at 0.05, then we should use $q = 0.05/J$.

The outputs are:

- `w`: the optimal weights. A non-negative vector of length J.
- `q_star`: the value q^* for which the weights are optimal. This may differ slightly from the original q if q is large.
- `q_thresh`: the largest value of q for which the weights can be computed exactly
- `c`: the normalizing constant produced by solving the optimization problem.

This method was proposed in Dobriban et al. (2015).

3.2 Spjotvoll

Spjotvoll p-value weights can be computed using: `[w, c] = spjotvoll_weights(mu, q)`. The inputs our best guess at μ_i from the prior data:

- `mu`: a vector of length J, the estimated means of test statistics, derived from the prior data
- `q`: The weights are optimal if each hypothesis is tested at level q. For instance, if we want to control the FWER globally at 0.05, then we should use $q = 0.05/J$.

The outputs are:

- `w`: the optimal weights. A non-negative vector of length J.
- `c`: the normalizing constant produced by solving the optimization problem.

This method was proposed by Wasserman and Roeder (2006); Roeder and Wasserman (2009) and independently by Rubin et al. (2006). It was called Spjotvoll weights in Dobriban et al. (2015), in honor of (Spjotvoll, 1972).

3.3 Exponential

Exponential weights with can be computed as `w = exp_weights(mu, beta, q)`. Here the inputs are

- `mu`: a vector of length J, the estimated means of test statistics, derived from the prior data
- `beta`: the tilt β, which defines the exponent of the weight. The weights are defined as: $w_i = \exp(\beta|\eta_i|)/c$, where $c = \sum_{i=1}^{J} \exp(\beta|\eta_i|)$.
- `q`: The weights are optimal if each hypothesis is tested at level q. For instance, if we want to control the FWER globally at 0.05, then we should use $q = 0.05/J$.

The outputs are:

- `w`: the exponential weights. A non-negative vector of length J.

Exponential weights are sensitive to large means. To guard against this sensitivity, we truncate the weights larger than $1/q$ and re-distribute their excess weight among the next largest weights.

This weighting scheme was proposed in (Roeder et al., 2006), who recommend $\beta = 2$ as a default.

4 An Example with Synthetic Data

We perform an experiment with synthetic data, showing how using prior data can improve power in the current study. The code is in a script `example.m` in the main folder. To run it, be sure to set your working directory to the main folder where the script resides. In this document we will walk through the code in `example.m` step by step.

The first step consists of getting the synthetic data. You have two options: you can walk through the generation of the data, or you can load the data directly.

4.1 Option 1: Load the readily generated data

The first option is to load the readily generated data. Run the second cell in the matlab script `example.m`

```
load('./Data/example_data.mat', 'J','P_current','t1','t2');
```

This will lead load the same data as Option 2 generates.

4.2 Option 2: Generate data

The second option is to walk through the data generating process. We generate two sets of test statistics, the prior and the current data. A small fraction of the prior data holds some information about the current data. However, most prior data is noise. In our experience working with GWAS data, this is a reasonable model for association studies done on two independent samples and two distinct traits (such as cardiovascular disease and aging).

We do this by drawing from a mixture distribution. We generate a large number J of tests. For each test i we flip a coin X_i: If $X_i = 1$, then the prior is meaningful, else it is noise. If the prior is meaningful, we draw a random negative μ_i and both test statistics (T_i^1, T_i^2) are Gaussians centered at μ_i. Else we draw two independent normal test statistics (T_i^1, T_i^2). This ensures that this small fraction of the data is correlated. The code is:

```
rng(0); %set seed
J = 1e3;
mu = - 2*abs(randn(J,1));
frac_sig = 0.1;
```

```
X = binornd(1,frac_sig,J,1);
t1 = normrnd(X.*mu,1);
t2 = normrnd(X.*mu,1);
```

The data that this generates shows the desired patter, as can be seen on a scatterplot. Most pairs have no correlation, but there is a small fraction that does.

The p-values for the one-sided tests $\mu_i = 0$ vs $\mu_i < 0$ utilizing only the current data are $P_i = \Phi(T_i^2)$.

4.3 Visualize Data

Either way, we should now have the variables `t1,t2,P_current,J` loaded in memory. Next we take a look at the data by plotting a scatter of the prior and current test statistics. This leads to the following figure:

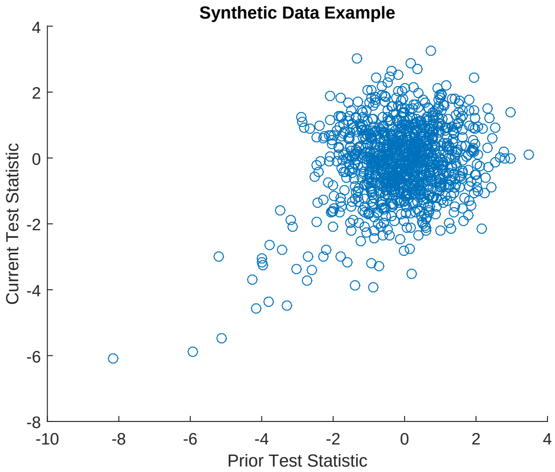

Figure 1: Scatter Plot of the Prior and Current Effects

There is only a weak global correlation between the two effects. However, there is a significant correlation in the tails.

4.4 Data Analysis

We want to test the hypotheses $\mu_i = 0$ against $\mu_i < 0$ for each i utilizing the current data T_i^2. The simplest way to do this is to do Bonferroni-corrected multiple testing. We choose an uncorrected significance level $\alpha = 0.05$ and call the `bonferroni(...)` function on the p-values. Note that this function was also written by the author.

```
alpha = 0.05; report = 'yes';
h_u=bonferroni(P_current,alpha,report);
```

The output should be:

```
Out of 1000 tests, 7 are significant using a  family-wise error rate  of 0.050000.
```

Thus in our case Bonferroni leads to 7 significant test statistics.

Alternatively, one can do p-value weighting. For this, we use T^1 as prior information. For simplicity we set the prior standard errors $\sigma = 1$ in this example. More detailed discussion on the choice of σ can be found in (Dobriban et al., 2015). As explained earlier, we set $q = 0.05/J$. Then we compute the weights. Finally, we run weighted Bonferroni on the weighted p-values $P_i' = P_i/w_i$. This is the code that accomplishes it:

```
q =alpha/J; %expected fraction of false rejections under 'null'
sigma = ones(J,1);
w = informed_weights(t1,sigma,q);
P_wr = P_current./w;
h_r=bonferroni(P_wr,alpha,report);
```

We should get the following output:

```
Out of 1000 tests, 12 are significant using a  family-wise error rate  of 0.050000.
```

Hence, in this example weighting increases the number of significant hits from 7 to 12.

4.5 Post-Analysis

One can get some insight into the procedure by examining which hypotheses were declared significant by the two methods. Typing

```
find(h_u==1)
find(h_r==1)
```

reveals that the significant hypotheses were:

```
ans =

    35
    84
   121
   563
   596
   734
   740


ans =

    35
    84
   121
   188
   221
   429
   563
   596
   645
   656
   734
   740
```

In this particular case weighting leads to a strict increase in power, selecting an addional 5 hypotheses. Taking 429 as an example we see that its P-value in the current data equals `P_current(429)=  1.0778e-04`, corresponding to a z-score `t2(429)=  -3.7001`, and this is not enough for it to be significant since the threshold is $0.05/1000 = 5e - 5$. However, it gets assigned a large weight `w(429)=  8.9348`, because its prior effect is large, `t1(429)= -4.2573`; so it's selected after weighting.

Another insight can be gained from plotting the weights as a function of the prior mean. We see that the weights are non-monotonic as a function of the prior mean. Indeed they place a large mass on the middle means between (-6 and -2).

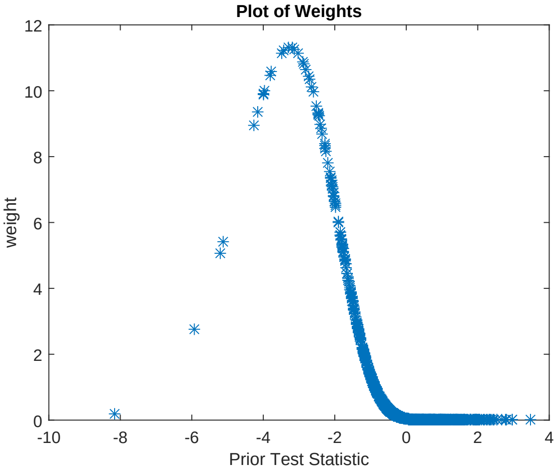

Figure 2: Plot of weights

References

Dobriban, E., Fortney, K., Kim, S., and Owen, A. B. (2015). Optimal p-value weighting for improved power in multiple testing. *in preparation*.

Roeder, K., Bacanu, S.-A., Wasserman, L., and Devlin, B. (2006). Using linkage genome scans to improve power of association in genome scans. *The American Journal of Human Genetics*, 78(2):243–252.

Roeder, K. and Wasserman, L. (2009). Genome-wide significance levels and weighted hypothesis testing. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 24(4):398–413.

Rubin, D., Dudoit, S., and Van der Laan, M. (2006). A method to increase the power of multiple testing procedures through sample splitting. *Statistical Applications in Genetics and Molecular Biology*, 5(1):1–19.

Spjotvoll, E. (1972). On the optimality of some multiple comparison procedures. *The Annals of Mathematical Statistics*, 43(2):398–411.

Wasserman, L. and Roeder, K. (2006). Weighted hypothesis testing. *arXiv preprint math/0604172*.