# EigenEdge package for MATLAB

Edgar Dobriban*

April 25, 2015

## Contents



## 1 Introduction

The EigenEdge MATLAB package contains open source implementations methods for working with eigenvalue distributions of large random matrices. In particular, it contains the Atomic method to compute the limit empirical spectrum of sample covariance matrices (proposed in Dobriban, 2015).

- Version: 0.0.1
- Requirements: Tested on MATLAB R2014a and R2014b.
- Author: Edgar Dobriban
- License: `GPL-3`

In addition, this package contains the code to reproduce all simulation results from the paper Dobriban (2015). These are contained in the `\Experiments` folder.

## 2 Installation

Extract the archive in any folder, say to `<path>` . The main functions are in the `Code` directory, which needs to be on the Matlab path, along with all of its subfolders. This can be accomplished in at least three ways. First, you can add the following lines to your Matlab startup:

```
addpath('<path>/pvalue_weighting_matlab/Code')
addpath('<path>/pvalue_weighting_matlab/Code/Basic')
addpath('<path>/pvalue_weighting_matlab/Code/External Helper Code')
```

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

The second option is to add those line to scripts that call functions in this package. The third option is to execute the `setpaths.m` script every time you start a new session with the EigenEdge package. This will include the current folder in the Matlab search path.

An example computation is in the `\Experiments\Examples\example.m` file. This is described in Section 3.

This file is the main documentation for the package. To start, look at the example (Section 3) or at the methods implemented (Sections 4).

# 3 Example

# 4 P-value weighting methods

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 $\Phi$ is the normal cumulative 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 $P_i \leq q w_i$. Usual Bonferroni corresponds to $w_i = 1$.

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

## 4.1 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 $\mu_i$ of the test statistics as:

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

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 **?**.

# References


Dobriban, E. (2015). Precise computation of limit spectra of sample covariance matrices.