1 `covdepGE`: Covariate Dependent Graph Estimation

1.1 Installation

Run the following in `R`:

```
devtools::install_github("JacobHelwig/covdepGE")
```

1.2 Overview

Suppose $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is a data matrix of independent observations $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_p)$, where, for $j \in 1, ..., p$:

$$\boldsymbol{x}_j \sim \mathcal{N}\left(\mu_j, \Sigma_{j,j}\right), \boldsymbol{x}_j \in \mathbb{R}^n \qquad\qquad \boldsymbol{X} \sim \mathcal{N}\left(\mu, \Sigma\right) \tag{1}$$

Let $\boldsymbol{Z}$ be an $n \times p'$ matrix of extraneous covariates. The conditional dependence structure of $\boldsymbol{x_1}, ..., \boldsymbol{x_p}$ can be modeled as an undirected graph $\mathcal{G}$ such that:

$$\mathcal{G}_{i,j} = \begin{cases} 1 & \iff \text{Cov}(\boldsymbol{x_i}, \boldsymbol{x_j}) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

That is, there is an edge between the $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ nodes if, and only if, these variables are dependent on each other given all other variables.

Further suppose that the conditional dependence structure of $\boldsymbol{X}$ is not homogeneous across the individuals, and is instead a continuous function of the extraneous covariates $\boldsymbol{Z}$[1](1). Then, this methodology aims to estimate a graph for each of the individuals, possibly unique to the individual, such that similar estimates are made for those who are similar to one another in terms of the extraneous covariates.

For an example application, see [1](1), wherein the sample was composed of healthy and cancerous individuals, $\boldsymbol{x}_1, ..., \boldsymbol{x}_8$ were protein expression levels of 8 genes, and $\boldsymbol{Z}$ was the copy number variation of a gene $\boldsymbol{z}$ associated with cancer, $\boldsymbol{z} \notin \{\boldsymbol{x}_1, ..., \boldsymbol{x}_8\}$.

1.3 Functionality

The main function, `covdepGE::covdepGE(`$\boldsymbol{X}, \boldsymbol{Z}$`)`, estimates the posterior distribution of the graphical structure $\mathcal{G}_l$ for each of the n individuals using a variational mean-field approximation. The function will output n $p \times p$ symmetric matrices $\mathcal{A}_l$, where $\mathcal{A}_{i,j}^{(l)}$ is the posterior inclusion probability of an edge between the node representing the i-th variable and the node representing the j-th variable.

1.4 To-do

- Implement KDE to obtain individual specific bandwidths. Currently, only one global bandwidth is used, and it requires that the user specify it (or use the default, deterministic value). This modification will estimate a unique bandwidth for each individual dependent on the empirical density of $\boldsymbol{Z}$ resulting from KDE.
- Add `KDE` argument (boolean). Then, the argument `tau` is only used when `KDE = F`
- Allow for user specification of a norm (`"1"`, `"2"`, or `"inf"`) and symmetrization method (`"mean"`, `"min"`, or `"max"`)
- Allow for user specification of hyperparameter `sigmasq`
- Change default `Pi_est` to a scalar
- Implement automatic `sigmabeta_sq` grid generation via arguments `varmax, varmin, n_sigma`
- Add warnings for when the optimal `sigmabeta_sq` is chosen at either of the grid end points
- Change return type of `alpha_matrices` from a `list` of p $n \times p$ matrices to a `list` of n $p \times p$ matrices
- Create a vignette demonstrating usage on a simple simulated dataset.
- Parallelization of the "main loop" over the predictors in `covdepGE_main.R`). This is complicated by the `C++` code, however, two potential solutions are:
  - StackOverflow suggestion
  - RcppParallel

  This is a finishing touch and **likely will not be implemented in the package by the end of the semester.**

References

[1] (1) Dasgupta, Sutanoy, et al. "An approximate Bayesian approach to covariate dependent graphical modeling." 2021