Sparse Supervised Gaussians (v0.6)

Summary

The **Sparse Supervised Gaussian** (SSG) library contains methods for learning dependencies among multiple high-dimensional variables in the presence of confounders, and using the learned dependencies to predict class memberships. The approach extends sparse high-dimensional (*large-p, small-n*) Gaussian Graphical Models to the setting where the model structures are conditioned on class labels and on potentially high-dimensional confounding variables. The learned structures can be used for improving predictions of class labels on test data.

Background

The *graphical lasso* (Friedman et al, 2008) learns a sparse inverse covariance (precision) matrix. Zeros in the precision matrix correspond to pairs of variables that are conditionally independent in the learned model. *MRCE* (Rothman et al, 2010) combines graphical lasso with sparse linear regression to allow the simultaneous adjustment for covariates while learning the structure. The joint fused graphical lasso (Danaher et al, 2014) fits multiple graphical lasso models to each class while imposing a penalty on the differences between the structures; however, the method does not adjust for confounding and does not model class memberships. The methods in our library *SSG* extend these approaches to the setting where multiple classes are present – cases and controls of some condition, for example. The methods learn one model for each class and can be used both for elucidating the between-class differences in the learned structures and for diagnosing the condition. This enables, for example, identification of significant changes in molecular interactions in cases and controls, where the interactions are confounded by clinical factors or by -omics variables. This also enables diagnosis of the disease status using both clinical predictors and the learned dependency networks.

Selected methods

1. *ccEmpGaussian*: *Class-conditional empirical Gaussians.* Fit an empirical Gaussian to each class, ie use the maximum likelihood precision matrix. This results in dense structures. It is a baseline against which to compare the other models. The predictions are via linear discriminant classifiers.
2. *ccGlasso*: *Class-conditional Graphical Lasso*. Train a graphical lasso model for each class; the predictions are via linear discriminant classifiers with the sparse class-conditional structures.
3. *ccGlassoPre: Class-conditional Graphical Lasso with Stability Pre-training.* Pre-learn the structure of each class using stability selection (Meinshausen and Bühlmann, 2010) with thresholding, then run graphical lasso. Using simulated data, it was observed empirically that this may result in more accurate structures, with little sacrifice in predictive power.
4. *ccGGM: Class-conditional Gaussian Graphical Models*. Uses the approach of Opgen-Rhein and Strimmer (2006), followed by thresholding of the precision entries, to fit a sparse Gaussian model to each class; the predictions are via linear discriminant classifiers with the sparse structures learned by the thresholding method above.
5. *ccMRCE: Class-conditional Sparse Multivariate Regression with Covariate Estimation.* When there are multiple classes (such as cases and controls for some condition), fit an MRCE model for each class. The predictions are by an application of Bayes rule combining priors over the class label with the conditionally-trained MRCE.
6. *cFGL*: *Conditional Fused Graphical LASSO.* The fused graphical lasso (Danaher et al, 2014) fits multiple graphical lasso models while imposing a penalty on differences between the structures. Our method *cFGL* generalises this by introducing covariates and by imposing penalties on the differences between the regression parameters for each class. This approach "marries" the joint fused graphical lasso method with the MRCE and uses a new iterative convex optimization procedure to learn the parameters. The predictions are by an application

of Bayes rule combining lasso regression of the class label on the side information with the class-conditional MRCE, where the training penalized large differences between the model parameters corresponding to each class.

The library contains additional functions to evaluate multiple performance metrics, visualize differences in the network structures, examine convergence of the optimization algorithms, etc. The released functions are supported for binary classification; please contact us for extensions to multiple classes. Future releases may include high-performance implementations.

Illustration

Figure 1 illustrates an application of SSG to MIMOmics datasets. Circle-, square-, and diamond-shaped nodes correspond to different groups of -omics variables; black and red links correspond to varying dependencies between cases and controls for applications to the colorectal cancer (left) and metabolic health (right) data. Line thickness indicates the significance of the disparities.

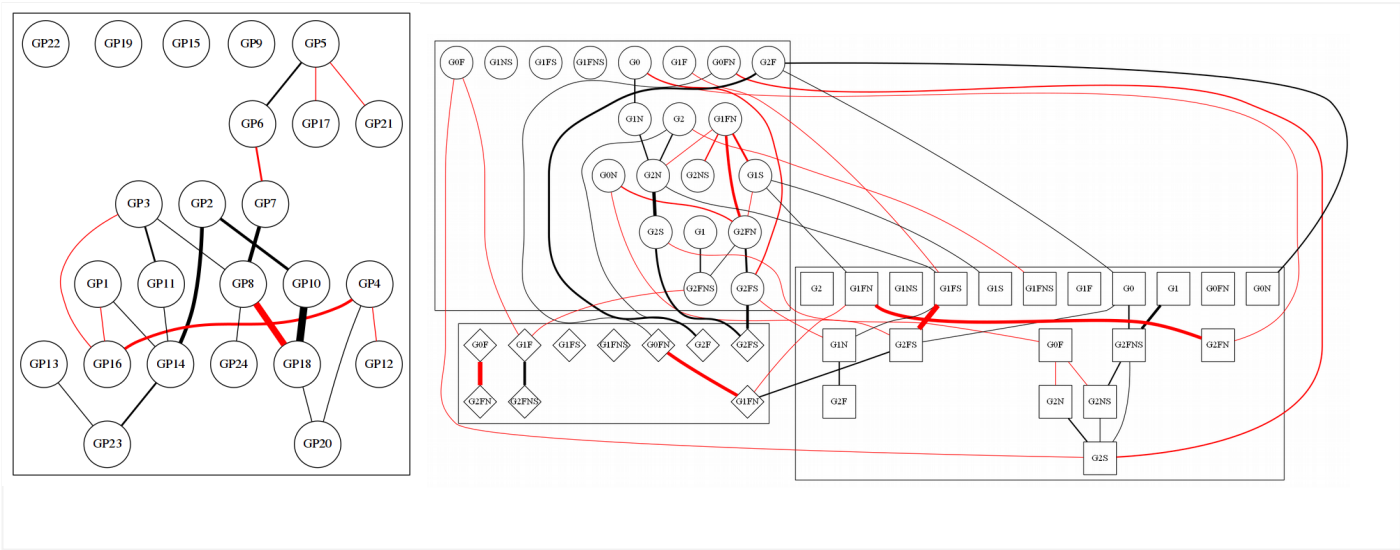

Figure 1 Illustration of an application of SSG to two MIMOmics datasets. *Left:* Difference between glycan network structures for cases and controls in colorectal cancer data. *Right:* Difference between glycan network structures for obese and non-obese patients, learned using a mixture of conditionally trained graphical lasso models, with sex, age, and age^2 as covariates.

References

1. P. Danaher, P. Wang, and D.M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76.2: 373-397, 2014.
2. J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432–441, 2008.
3. N. Meinshausen and P. Bühlmann. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72.4: 417-473, 2010.
4. A.J. Rothman, E. Levina, J. Zhu. Sparse multivariate regression with covariance estimation. Journal of Computational and Graphical Statistics 19.4: 947-962, 2010.
5. R. Opgen-Rhein and K. Strimmer. Using regularized dynamic correlation to infer gene dependency networks from time-series microarray data. Proceedings of the 4th International Workshop on Computational Systems Biology, WCSB: 73-76, 2006.

Software Availability: Python and R code are available at https://github.com/mimomics

Contact: peter.orchard@pharmaticsltd.com