GGLasso - a Python package for General Graphical Lasso computation

Fabian Schaipp[1], **Oleg Vlasovets**[2,3], **and Christian L. Müller**[2,3,4]

1 Technische Universität München **2** Institute of Computational Biology, Helmholtz Zentrum München **3** Department of Statistics, Ludwig-Maximilians-Universität München **4** Center for Computational Mathematics, Flatiron Institute, New York

DOI: 10.21105/joss.03865

Software

- Review
- Repository
- Archive

Editor: George K. Thiruvathukal

Reviewers:

- @papachristoumarios
- @jameschapman19

Submitted: 18 October 2021
Published: 03 December 2021

Summary

We introduce `GGLasso`, a Python package for solving General Graphical Lasso problems. The Graphical Lasso scheme, introduced by (Friedman et al., 2007) (see also (Banerjee et al., 2008; Yuan & Lin, 2007)), estimates a sparse inverse covariance matrix Θ from multivariate Gaussian data $\mathcal{X} \sim \mathcal{N}(\mu, \Sigma) \in \mathbb{R}^p$. Originally proposed by (Dempster, 1972) under the name Covariance Selection, this estimation framework has been extended to include latent variables in (Chandrasekaran et al., 2012). Recent extensions also include the joint estimation of multiple inverse covariance matrices, see, e.g., in (Danaher et al., 2013; Tomasi et al., 2018). The `GGLasso` package contains methods for solving the general problem formulation:

$$\min_{\Theta, L \in \mathbb{S}^K_{++}} \quad \sum_{k=1}^{K} \left(-\log\det(\Theta^{(k)} - L^{(k)}) + \langle S^{(k)}, \Theta^{(k)} - L^{(k)} \rangle \right) + \mathcal{P}(\Theta) + \sum_{k=1}^{K} \mu_{1,k} \| L^{(k)} \|_{\star}. \tag{1}$$

Here, we denote with $\mathbb{S}^K_{++}$ the K-product of the space of symmetric, positive definite matrices. Moreover, we write $\Theta = (\Theta^{(1)}, \dots, \Theta^{(K)})$ for the sparse component of the inverse covariances and $L = (L^{(1)}, \dots, L^{(K)})$ for the low rank components, formed by potential latent variables. Here, $\mathcal{P}$ is a regularization function that induces a desired sparsity structure. The above problem formulation subsumes important special cases, including the single (latent variable) Graphical Lasso, the Group, and the Fused Graphical Lasso.

Statement of need

Currently, there is no Python package available for solving general Graphical Lasso instances. The standard single Graphical Lasso problem (SGL) can be solved in `scikit-learn` (Pedregosa et al., 2011). The `skggm` package provides several algorithmic and model selection extensions for the single Graphical Lasso problem (Laska & Narayan, 2017). The package `regain` (Tomasi et al., 2018) comprises solvers for single and Fused Graphical Lasso problems, with and without latent variables. With `GGLasso`, we make the following contributions:

- Proposing a uniform framework for solving Graphical Lasso problems.
- Providing solvers for Group Graphical Lasso problems (with and without latent variables).

- Providing a solver for – what we call – *nonconforming GGL* problems where not all variables need to be present in every instance. We detail a use case of this novel extension on synthetic data.
- Implementing a block-wise ADMM solver for SGL problems following (Witten et al., 2011) as well as proximal point solvers for FGL and GGL problems (N. Zhang et al., 2021; Y. Zhang et al., 2020).

In the table below we give an overview of existing functionalities and the GGLasso package.

	scikit-learn	regain	GGLasso	comment
SGL	**yes**	**yes**	**yes**	new: block-wise solver
SGL + latent	**no**	**yes**	**yes**	
GGL	**no**	**no**	**yes**	
GGL + latent	**no**	**no**	**yes**	
FGL	**no**	**yes**	**yes**	new: proximal point solver
FGL + latent	**no**	**yes**	**yes**	
GGL nonconforming (+latent)	**no**	**no**	**yes**	

Functionalities

Installation and problem instantiation

GGLasso can be installed via pip.

```
pip install gglasso
```

The central object of GGLasso is the class glasso_problem which streamlines the solving or model selection procedure for SGL, GGL, and FGL problems with or without latent variables.

As an example, we instantiate a single Graphical Lasso problem (see the problem formulation below). We input the empirical covariance matrix S and the number of samples N. We can choose to model latent variables and set the regularization parameters via the other input arguments.

```
# Import the main class of the package
from gglasso.problem import glasso_problem

# Define a SGL problem instance with given data S
problem  = glasso_problem(S, N, reg = None,
                          reg_params = {'lambda1': 0.01}, latent = False)
```

As a second example, we instantiate a Group Graphical Lasso problem with latent variables. Typically, the optimal choice of the regularization parameters are not known and are determined via model selection.

```
# Define a GGL problem instance with given data S
problem  = glasso_problem(S, N, reg = "GGL", reg_params = None, latent = True)
```

Depending on the input arguments, glasso_problem comprises two main modes:

- if regularization parameters are specified, the problem-dependent default solver is called.

- if regularization parameters are *not* specified, `GGLasso` performs model selection via grid search and the extended BIC criterion (Foygel & Drton, 2010)).

```
problem.solve()
problem.model_selection()
```

For further information on the input arguments and methods, we refer to the detailled documentation.

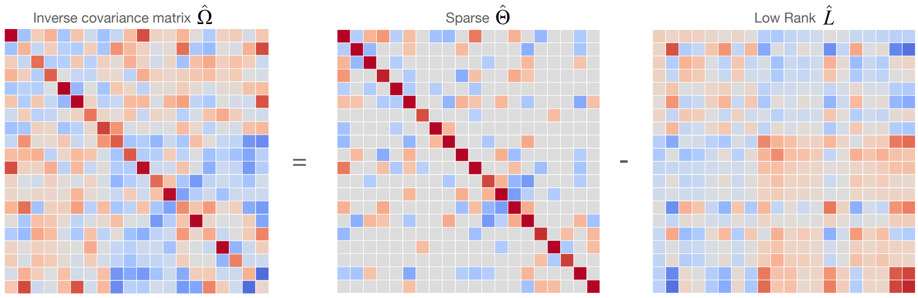

Figure 1: Illustration of the latent SGL: The estimated inverse covariance matrix $\hat{\Omega}$ decomposes into a sparse component $\hat{\Theta}$ (central) and a low-rank component $\hat{L}$ (right).

Problem formulation

We list important special cases of the problem formulation given in Equation 1. For a mathematical formulation of each special case, we refer to the documentation.

Single Graphical Lasso (*SGL*):

For $K = 1$, the problem reduces to the single (latent variable) Graphical Lasso where

$$\mathcal{P}(\Theta) = \lambda_1 \sum_{i \neq j} |\Theta_{ij}|.$$

An illustration of the single latent variable Graphical Lasso model output is shown in Figure 1.

Group Graphical Lasso (*GGL*):

For

$$\mathcal{P}(\Theta) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\Theta_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \left(\sum_{k=1}^{K} |\Theta_{ij}^{(k)}|^2 \right)^{\frac{1}{2}}$$

we obtain the Group Graphical Lasso as formulated in (Danaher et al., 2013).

Fused Graphical Lasso (*FGL*):

For

$$\mathcal{P}(\Theta) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\Theta_{ij}^{(k)}| + \lambda_2 \sum_{k=2}^{K} \sum_{i \neq j} |\Theta_{ij}^{(k)} - \Theta_{ij}^{(k-1)}|$$

we obtain Fused (also called Time-Varying) Graphical Lasso (Danaher et al., 2013; Hallac et al., 2017; Tomasi et al., 2018).

Nonconforming GGL:

Consider the GGL case in a situation where not all variables are observed in every instance $k = 1, \dots, K$. `GGLasso` is able to solve these problems and include latent variables. We provide the mathematical details in the documentation and give an example.

Optimization algorithms

The `GGLasso` package implements several methods with provable convergence guarantees for solving the optimization problems formulated above.

- *ADMM*: for all problem formulations we implemented the ADMM algorithm (Boyd et al., 2011). ADMM is a flexible and efficient optimization scheme which is specifically suited for Graphical Lasso problems as it only relies on efficient computation of the proximal operators of the involved functions (Danaher et al., 2013; Ma et al., 2013; Tomasi et al., 2018).
- *PPDNA*: for GGL and FGL problems without latent variables, we included the proximal point solver proposed in (N. Zhang et al., 2021; Y. Zhang et al., 2020). According to the numerical experiments in (Y. Zhang et al., 2020), PPDNA can be an efficient alternative to ADMM especially for fast local convergence.
- *block-ADMM*: for SGL problems without latent variables, we implemented a method which solves the problem block-wise, following the proposal in (Witten et al., 2011). This wrapper simply applies the ADMM solver to all connected components of the empirical covariance matrix after thresholding.

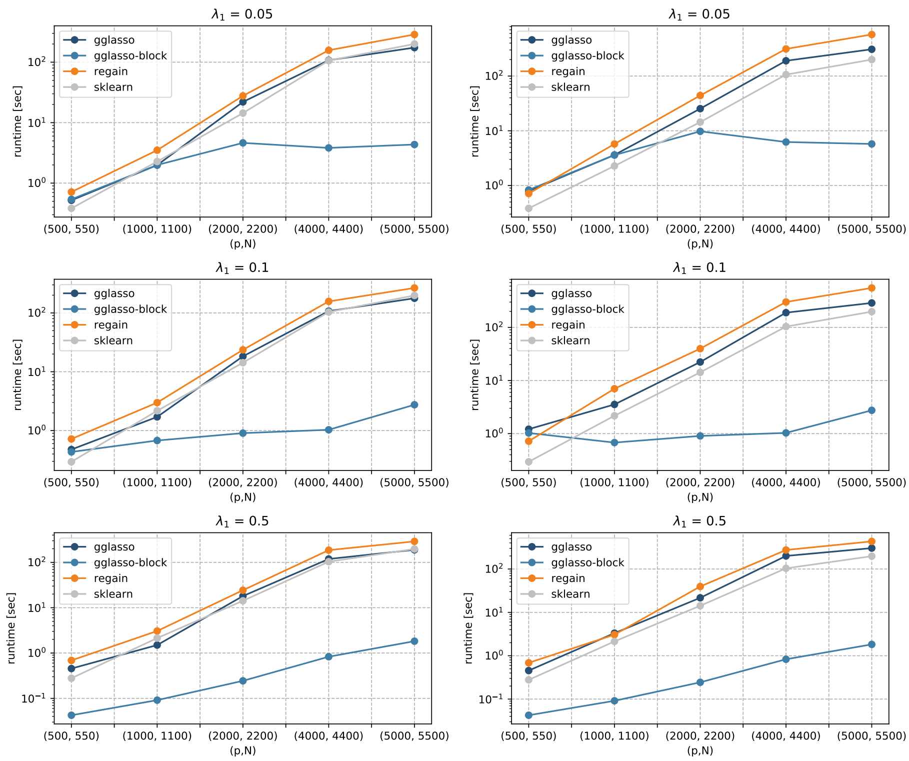

Figure 2: Runtime comparison for SGL problems of varying dimension and sample size at three different λ_1 values. The left column shows the runtime at low accuracy, the right column at high accuracy.

Benchmarks and applications

In our example gallery, we included benchmarks comparing the solvers in `GGLasso` to state-of-the-art software as well as illustrative examples explaining the usage and functionalities of the package. We want to emphasize the following examples:

- Benchmarks for SGL problems: our solver is competitive with `scikit-learn` and `regain`. The newly implemented block-wise solver is highly efficient for large sparse networks (see Figure 2 for runtime comparison at low and high accuracy, respectively).
- Soil microbiome application: following (Kurtz et al., 2019), we demonstrate how latent variables can be used to identify hidden confounders in microbial network inference.
- Nonconforming GGL: we illustrate how to use `GGLasso` for GGL problems with missing variables.

Acknowledgements

We thank Prof. Dr. Michael Ulbrich, TU Munich, for supervising the Master's thesis of FS that led to the development of the software. We also thank Dr. Zachary D. Kurtz for helping with testing of the latent graphical model implementation.

References

Banerjee, O., El Ghaoui, L., & D'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res.*, *9*, 485–516. http://dl.acm.org/citation.cfm?id=1390696

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, *3*(1), 1–122. https://doi.org/10.1561/2200000016

Chandrasekaran, V., Parrilo, P. A., & Willsky, A. S. (2012). Latent variable graphical model selection via convex optimization. *Ann. Statist.*, *40*(4), 1935–1967. https://doi.org/10.1214/11-aos949

Danaher, P., Wang, P., & Witten, D. M. (2013). The joint graphical lasso for inverse covariance estimation across multiple classes. *J. R. Stat. Soc. B*, *76*(2), 373–397. https://doi.org/10.1111/rssb.12033

Dempster, A. P. (1972). Covariance selection. *Biometrics*, *28*(1), 157–175. https://doi.org/10.2307/2528966

Foygel, R., & Drton, M. (2010). Extended Bayesian Information Criteria for Gaussian Graphical Models. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems* (Vol. 23). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2010/file/072b030ba126b2f4b2374f342be9ed44-Paper.pdf

Friedman, J., Hastie, T., & Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441. https://doi.org/10.1093/biostatistics/kxm045

Hallac, D., Park, Y., Boyd, S., & Leskovec, J. (2017, August). Network Inference via the Time-Varying Graphical Lasso. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/3097983.3098037

Kurtz, Z. D., Bonneau, R., & Müller, C. L. (2019). Disentangling microbial associations from hidden environmental and technical factors via latent graphical models. *bioRxiv*, 2019.12.21.885889. https://doi.org/10.1101/2019.12.21.885889

Laska, J., & Narayan, M. (2017). *skggm 0.2.7: A scikit-learn compatible package for Gaussian and related Graphical Models*. https://doi.org/10.5281/zenodo.830033

Ma, S., Xue, L., & Zou, H. (2013). Alternating Direction Methods for Latent Variable Gaussian Graphical Model Selection. *Neural Comput.*, *25*(8), 2172–2198. https://doi.org/10.1162/neco_a_00379

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *J Mach Learn Res*, *12*, 2825–2830.

Tomasi, F., Tozzo, V., Salzo, S., & Verri, A. (2018, July). Latent Variable Time-varying Network Inference. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3219819.3220121

Witten, D. M., Friedman, J. H., & Simon, N. (2011). New Insights and Faster Computations for the Graphical Lasso. *J. Comput. Graph. Statist.*, *20*(4), 892–900. https://doi.org/10.1198/jcgs.2011.11051a

Yuan, M., & Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, *94*(1), 19–35. https://doi.org/10.1093/biomet/asm018

Zhang, N., Zhang, Y., Sun, D., & Toh, K.-C. (2021). An efficient linearly convergent regularized proximal point algorithm for fused multiple graphical Lasso problems. *SIAM J. Math. Data Sci.*, *3*(2), 524–543. https://doi.org/10.1137/20M1344160

Zhang, Y., Zhang, N., Sun, D., & Toh, K.-C. (2020). A proximal point dual Newton algorithm for solving group graphical Lasso problems. *SIAM J. Optim.*, *30*(3), 2197–2220. https://doi.org/10.1137/19M1267830