





sparse-lm: Sparse linear regression models in Python

Luis Barroso-Luque ^{1,2} and Fengyu Xie ^{1,2}

¹ Materials Sciences Division, Lawrence Berkeley National Laboratory, Berkeley CA, 94720, USA ² Department of Materials Science and Engineering, University of California Berkeley, Berkeley CA, 94720, USA  Corresponding author

DOI: [10.21105/joss.05867](https://doi.org/10.21105/joss.05867)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Summary

Sparse linear regression models are a powerful tool for capturing linear relationships in high dimensional spaces. Sparse models have only a small number of nonzero parameters—even if the number of covariates used in estimation is large—as a result they can be easier to fit and interpret compared to dense models (Hastie et al., 2015). Regression objectives resulting in sparse linear models such as the Lasso (Tibshirani, 2018; Zou, 2006) and Best Subset Selection (Hocking & Leslie, 1967) have been widely used in a variety of fields. However, many regression problems involve covariates that have a natural underlying structure, such as group or hierarchical relationships, that can be further leveraged to obtain improved model performance and interpretability. Such structured regression problems occur in a wide range of fields including genomics (Chen & Wang, 2021), bioinformatics (Ma et al., 2007), medicine (Kim et al., 2012), econometrics (Athey & Imbens, 2017), chemistry (Gu et al., 2018), and materials science (Leong & Tan, 2019). Several generalizations of the Lasso (Friedman et al., 2010; Simon et al., 2013; Wang & Tian, 2019; Yuan & Lin, 2006) and Best Subset Selection (Bertsimas et al., 2016; Bertsimas & King, 2016) have been developed to effectively exploit additional structure in linear regression. The sparse-lm Python package provides a flexible, comprehensive, and user-friendly implementation of (structured) sparse linear regression models, which allows researchers to easily experiment and choose the best regression model for their specific problem.

Editor: [Mehmet Hakan Satman](#) 



Reviewers:

- [@htjb](#)
- [@mhu48](#)

Submitted: 09 August 2023

Published: 21 December 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Statement of need

The sparse-lm Python package implements a variety of sparse linear regression models based on convex objectives (generalizations of the Lasso) and mixed integer quadratic programming objectives (generalizations of Best Subset Selection) that support a flexible range of ways to introduce structured sparsity. The linear models in sparse-lm are implemented to be compatible with scikit-learn (Buitinck et al., 2013; Pedregosa et al., 2011), in order to enable interoperability with the wide range of tools and workflows available. The regression optimization problems in sparse-lm are implemented and solved using cvxpy (Diamond & Boyd, 2016), which allows users to choose from a variety of well-established open-source and proprietary solvers. In particular, for regression problems with mixed integer programming objectives, access to state-of-the-art proprietary solvers enables solving larger problems that would otherwise be unsolvable within reasonable time limits.

A handful of pre-existing Python libraries implement a subset of sparse linear regression models that are also scikit-learn compatible. celer (Massias et al., 2018) and groupyr (Richie-Halford et al., 2021) include efficient implementations of the Lasso and Group Lasso. group-lasso (Moe, 2020) is another scikit-learn compatible implementation of the Group Lasso. skglm (Bertrand et al., 2022) includes several implementations of sparse linear models based on regularization using combinations of ℓ_p ($p \in \{1/2, 2/3, 1, 2\}$) norms and pseudo-

norms. And `abess` (Zhu et al., 2022) includes an implementation of Best Subset Selection and ℓ_0 pseudo-norm regularization.

The aforementioned packages include highly performant versions of the specific models they implement. However, none of these packages implement the full range of sparse linear models available in `sparse-lm`, nor do they support the flexibility to modify the optimization objective and choose among many open-source and commercially available solvers. `sparse-lm` satisfies the need for a flexible and comprehensive library that enables easy experimentation and comparisons of different sparse linear regression algorithms within a single package.

Background

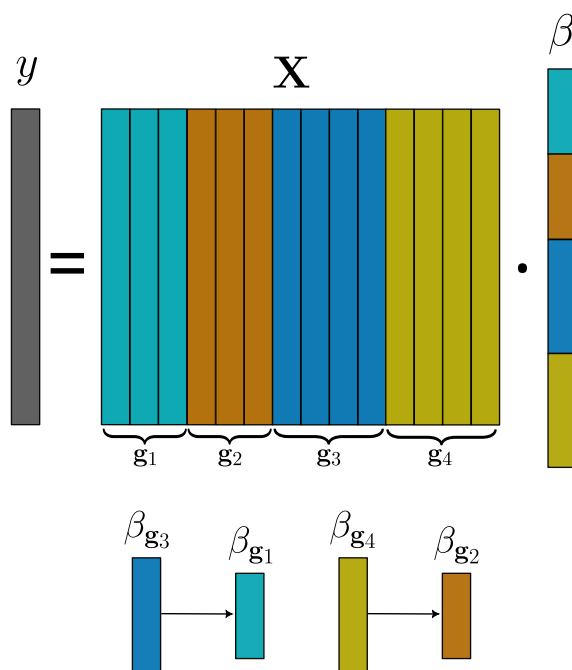


Figure 1: Schematic of a linear model with grouped covariates with hierarchical relations. Groups of covariates are represented with different colors and hierarchical relationships are represented with arrows (i.e. group 3 depends on group 1). The figure was inspired by Ref. (Richie-Halford et al., 2021).

Structured sparsity can be introduced into regression problems in one of two ways: convex group regularization or mixed integer quadratic programming with linear constraints. The first way to obtain structured sparsity is by using regularization based on generalizations of the Lasso, such as the Group Lasso and the Sparse Group Lasso (Friedman et al., 2010; Simon et al., 2013; Wang & Tian, 2019; Yuan & Lin, 2006). The Sparse Group Lasso regression problem can be expressed as follows,

$$\beta^* = \underset{\beta}{\operatorname{argmin}} \|\mathbf{X}\beta - \mathbf{y}\|_2^2 + (1 - \alpha)\lambda \sum_{\mathbf{g} \in G} \sqrt{|\mathbf{g}|} \|\beta_{\mathbf{g}}\|_2 + \alpha\lambda \|\beta\|_1 \quad (1)$$

where \mathbf{X} is the design matrix, \mathbf{y} is the response vector, and β are the regression coefficients. \mathbf{g} are groups of covariate indices, G is the set of all such groups being considered, and $\beta_{\mathbf{g}} \in \mathbb{R}^{|\mathbf{g}|}$ are the covariate coefficients in group \mathbf{g} . $\lambda \in \mathbb{R}_+$ and $\alpha \in [0, 1]$ are regularization hyperparameters. The parameter $\alpha \in [0, 1]$ controls the relative weight between the single covariate ℓ_1 regularization and the group regularization term. When $\alpha = 0$, the regression

problem reduces to the Group Lasso objective, and when $\alpha = 1$, the problem reduces to the Lasso objective.

The (Sparse) Group Lasso can be directly used to obtain a grouped sparsity pattern. Hierarchical sparsity patterns can be obtained by extending the Group Lasso to allow overlapping groups, which is referred to as the Overlap Group Lasso (Hastie et al., 2015).

The second method to obtain structured sparsity is by introducing linear constraints into the regression objective. Introducing linear constraints is straight-forward in mixed integer quadratic programming (MIQP) formulations of the Best Subset Selection (Bertsimas et al., 2016; Bertsimas & King, 2016). The general MIQP formulation of Best Subset Selection with grouped covariates and hierarchical constraints can be expressed as follows,

$$\beta^* = \underset{\beta}{\operatorname{argmin}} \beta^\top (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}) \beta - 2\mathbf{y}^\top \mathbf{X} \beta \quad (2)$$

subject to $z_g \in \{0, 1\}$
 $-Mz_g \mathbf{1} \leq \beta_g \leq Mz_g \mathbf{1}$
 $\sum_{g \in G} z_g \leq k$
 $z_g \leq z_h$

where z_g are binary slack variables that indicate whether the covariates in each group g are included in the model. The first set of inequality constraints ensure that coefficients β_g are nonzero if and only if their corresponding slack variable $z_g = 1$. M is a fixed parameter that can be estimated from the data (Bertsimas et al., 2016). The second inequality constraint introduces general sparsity by ensuring that at most k coefficients are nonzero. If G includes only singleton groups of covariates then the MIQP formulation is equivalent to the Best Subset Selection problem, otherwise it is a generalization that enables group-level sparsity structure. The last inequality constraint can be used to introduce hierarchical structure into the model. Finally, we have also included an ℓ_2 regularization term controlled by the hyperparameter λ , which is useful when dealing with poorly conditioned design matrices.

The user-friendly implementation of statistical regression models with structured sparsity parametrized via Group Lasso or Best Subset Selection based objectives in `sparse-lm`, along with the flexibility to choose from a variety of established solvers, enables researchers to quickly iterate, experiment and benchmark performance when choosing the best regression model for their specific problem. `sparse-lm` has already been used to build linear models with structured sparsity in a handful of material science studies (Barroso-Luque et al., 2022; Xie et al., 2023; Zhong et al., 2022, 2023).

Usage

Since the linear regression models in `sparse-lm` are implemented to be compatible with `scikit-learn` (Buitinck et al., 2013; Pedregosa et al., 2011), they can be used independently or as part of a workflow—such as in a hyperparameter selection class or a pipeline—in similar fashion to any of the available models in the `sklearn.linear_model` module.

Implemented regression models

The table below shows the regression models that are implemented in `sparse-lm` as well as available implementations in other Python packages. A checkmark (✓) indicates that the model selected is implemented in the package located in the corresponding column.

Model	sparse- lm	celer	groupyr	group- lasso	skglm	abess
(Adaptive) Lasso	✓	✓			✓	
(Adaptive) Group Lasso	✓	✓	✓	✓	✓	
(Adaptive) Sparse Group Lasso	✓		✓	✓	✓	
(Adaptive) Ridged Group Lasso	✓				✓	
Best Subset Selection	✓					✓
Ridged Best Subset Selection	✓					
ℓ_0 pseudo-norm	✓					
$\ell_0\ell_2$ mixed-norm	✓					
$\ell_{1/2}$ psuedo-norm					✓	
$\ell_{2/3}$ psuedo-norm					✓	

Note that only `sparse-lm` includes adaptive versions of Lasso based estimators. However, some of the third party packages, notably `skglm` and `abess`, include additional penalties and regression objectives that are not implemented in `sparse-lm`.

Implemented model selection and composition tools

In addition to the regression models in the table above, a few model selection and composition models are also implemented. These models are listed below:

- One standard deviation rule grid search cross-validation
- Line search cross-validation
- Stepwise composite estimator

The package can be downloaded through the [Python Package Index](#). Documentation, including an API reference and examples, can be found in the [online documentation](#).

Acknowledgements

The first author (L.B.L.) is the lead developer of `sparse-lm`, and the lead and corresponding author. The second author (F.X.) is a main contributor to the package. Both authors drafted, reviewed and edited the manuscript.

L.B.L. and F.X. would like acknowledge the contributions from Peichen Zhong, Ronald L. Kam, and Tina Chen to the development of `sparse-lm`. L.B.L gratefully acknowledges support from the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1752814.

References

- Athey, S., & Imbens, G. W. (2017). The State of Applied Econometrics: Causality and Policy Evaluation. *Journal of Economic Perspectives*, 31(2), 3–32. <https://doi.org/10.1257/jep.31.2.3>
- Barroso-Luque, L., Zhong, P., Yang, J. H., Xie, F., Chen, T., Ouyang, B., & Ceder, G. (2022). Cluster expansions of multicomponent ionic materials: Formalism and methodology. *Physical Review B*, 106(14), 144202. <https://doi.org/10.1103/PhysRevB.106.144202>

- Bertrand, Q., Klopfenstein, Q., Bannier, P.-A., Gidel, G., & Massias, M. (2022). Beyond L1: Faster and Better Sparse Models with skglm. *Advances in Neural Information Processing Systems*, 35, 38950–38965. https://proceedings.neurips.cc/paper_files/paper/2022/hash/fe5c31e525e9a26a1426ab0b589f42fe-Abstract-Conference.html
- Bertsimas, D., & King, A. (2016). OR Forum—An Algorithmic Approach to Linear Regression. *Operations Research*, 64(1), 2–16. <https://doi.org/10.1287/opre.2015.1436>
- Bertsimas, D., King, A., & Mazumder, R. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2), 813–852. <https://doi.org/10.1214/15-AOS1388>
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.
- Chen, S., & Wang, P. (2021). *Gene Selection from Biological Data via Group Lasso for Logistic Regression Model: Effects of Different Clustering Algorithms*. 6374–6379. <https://doi.org/10.23919/CCC52363.2021.9549471>
- Diamond, S., & Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83), 1–5.
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. *arXiv:1001.0736 [Math, Stat]*. <http://arxiv.org/abs/1001.0736>
- Gu, G. H., Plechac, P., & Vlachos, D. G. (2018). Thermochemistry of gas-phase and surface species via LASSO-assisted subgraph selection. *Reaction Chemistry & Engineering*, 3(4), 454–466. <https://doi.org/10.1039/C7RE00210F>
- Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: The lasso and generalizations*. Chapman & Hall/CRC. ISBN: 1498712169
- Hocking, R. R., & Leslie, R. N. (1967). Selection of the best subset in regression analysis. *Technometrics*, 9(4), 531–540. <https://doi.org/10.1080/00401706.1967.10490502>
- Kim, J., Sohn, I., Jung, S.-H., Kim, S., & Park, C. (2012). Analysis of Survival Data with Group Lasso. *Communications in Statistics - Simulation and Computation*, 41(9), 1593–1605. <https://doi.org/10.1080/03610918.2011.611311>
- Leong, Z., & Tan, T. L. (2019). Robust cluster expansion of multicomponent systems using structured sparsity. *Physical Review B*, 100(13), 134108. <https://doi.org/10.1103/PhysRevB.100.134108>
- Ma, S., Song, X., & Huang, J. (2007). Supervised group Lasso with applications to microarray data analysis. *BMC Bioinformatics*, 8(1), 60. <https://doi.org/10.1186/1471-2105-8-60>
- Massias, M., Gramfort, A., & Salmon, J. (2018). Celer: A fast solver for the lasso with dual extrapolation. *Proceedings of the 35th International Conference on Machine Learning*, 80, 3321–3330.
- Moe, Y. M. (2020). Group lasso. In *GitHub repository*. <https://github.com/yngvem/group-lasso>; GitHub.
- 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. *Journal of Machine Learning Research*, 12, 2825–2830.

- Richie-Halford, A., Narayan, M., Simon, N., Yeatman, J., & Rokem, A. (2021). Groupyr: Sparse group lasso in Python. *Journal of Open Source Software*, 6(58), 3024. <https://doi.org/10.21105/joss.03024>
- Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A Sparse-Group Lasso. *Journal of Computational and Graphical Statistics*, 22(2), 231–245. <https://doi.org/10.1080/10618600.2012.681250>
- Tibshirani, R. (2018). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288. <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- Wang, M., & Tian, G.-L. (2019). Adaptive group Lasso for high-dimensional generalized linear models. *Statistical Papers*, 60(5), 1469–1486. <https://doi.org/10.1007/s00362-017-0882-z>
- Xie, F., Zhong, P., Barroso-Luque, L., Ouyang, B., & Ceder, G. (2023). Semigrand-canonical Monte-Carlo simulation methods for charge-decorated cluster expansions. *Computational Materials Science*, 218, 112000. <https://doi.org/10.1016/j.commatsci.2022.112000>
- Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 49–67. <https://doi.org/10.1111/j.1467-9868.2005.00532.x>
- Zhong, P., Chen, T., Barroso-Luque, L., Xie, F., & Ceder, G. (2022). An L0 L2-norm regularized regression model for construction of robust cluster expansions in multicomponent systems. *Physical Review B*, 106(2), 024203. <https://doi.org/10.1103/PhysRevB.106.024203>
- Zhong, P., Xie, F., Barroso-Luque, L., Huang, L., & Ceder, G. (2023). Modeling Intercalation Chemistry with Multiredox Reactions by Sparse Lattice Models in Disordered Rocksalt Cathodes. *PRX Energy*, 2(4), 043005. <https://doi.org/10.1103/PRXEnergy.2.043005>
- Zhu, J., Wang, X., Hu, L., Huang, J., Jiang, K., Zhang, Y., Lin, S., & Zhu, J. (2022). Abess: A fast best-subset selection library in Python and R. *Journal of Machine Learning Research*, 23(202), 1–7. <http://jmlr.org/papers/v23/21-1060.html>
- Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476), 1418–1429. <https://doi.org/10.1198/016214506000000735>