Adaptive generalized logistic lasso and its application to rankings in sports

Robert Bajons, Kurt Hornik

Institute of Statistics and Mathematics, Vienna University of Economics and Business, Austria
robert.bajons@wu.ac.at

Abstract. The generalized lasso is a popular model for ranking competitors, as it allows for implicit grouping of estimated abilities. In this work, we present an implementation of an adaptive variant of the generalized lasso penalty for logistic regression using conic programming principles. This approach is flexible, robust, and fast, especially in a high-dimensional setting. The methodology is applied to sports data, with the aim of ranking soccer players based on their contribution to possession sequences.

Keywords: Generalized lasso; logistic regression; rankings

1 Introduction

In many fields, e.g. in economics, scientometrics, or sports, there is an innate interest in ranking competitors. However, given the multi-dimensionality and the difficulty of the problem, usually partial rankings, i.e. rankings that allow for groups, are preferred over total rankings. To serve such ranking mentalities and account for potential over-interpretation of insignificant differences between abilities of competitors, Masarotto and Varin (2012) propose the ranking lasso given as the solution of the problem

$$\min_{\boldsymbol{\beta}} \ell\left(Y; X, \boldsymbol{\beta}\right) + \lambda \sum_{i<j}^{N} w_{ij} |\beta_i - \beta_j|, \tag{1}$$

where w_{ij} are pair-specific weights. The penalty term in Eq. (1) is a variation of the lasso penalty that allows for grouping of the estimated abilities of competitors into several equivalence classes. This problem can be more generally written as

$$\min_{\boldsymbol{\beta}} \ell\left(Y; X, \boldsymbol{\beta}\right) + \lambda ||D\boldsymbol{\beta}||_1. \tag{2}$$

Here the matrix $D \in \mathbb{R}^{m \times p}$ represents some structural behavior of the coefficients $\boldsymbol{\beta}$. In principle, $\ell(Y; X, \boldsymbol{\beta})$ may represent any convex loss function, however, for our purpose, we consider the (negative) log-likelihood function for the binomial response variable Y in the usual logistic regression framework. For squared loss, Eq. (2) is often termed generalized lasso and encompasses various special cases (see Tibshirani and Taylor (2011) for details).

In general, there are many ways to solve the above problem. Masarotto and Varin (2012), e.g., derive an augmented Lagrangian method to solve the problem. However, when incorporating specific ranking-based structures on the coefficients, the dimension of D quickly increases, and consequently such algorithms do not scale reasonably well. In this paper, we present an approach using interior point methods and modern conic programming principles. Such an approach has been proven to be reliable for similar convex problems in Schwendinger et al. (2021) and specifically works well in the context of high-dimensional D. Furthermore, we discuss an application for ranking soccer players, where the matrix D has a particular and high-dimensional structure, which is distinct from the classical ranking lasso case of Eq. (1).

2 Adaptive generalized logistic lasso via conic programming

We rewrite Eq. (2) and consider the adaptive generalized logistic lasso (AGLL) problem given as

$$\min_{\boldsymbol{\beta}} -\Big(\sum_{i=1}^{n} y_i \log(h_{\boldsymbol{\beta}}(x_i)) + (1-y_i)\log(1-h_{\boldsymbol{\beta}}(x_i))\Big) + \lambda \sum_{j=1}^{m} w_j |D_j|, \qquad (3)$$

with logistic function $h_{\boldsymbol{\beta}}(x) = 1/\big(1+\exp(-\boldsymbol{\beta}^\top x)\big)$. In this case, D_j is the j-th component of $D\boldsymbol{\beta}$, with $D \in \mathbb{R}^{m\times p}$.

In order to use conic programming techniques to solve Problem (3), we first define a conic program (CP) as:

$$\begin{aligned} &\text{minimize } a_0^T x \\ &\quad \text{s.t. } Ax + s = b, \quad s \in \mathcal{K}, \end{aligned} \qquad (4)$$

where the set $\mathcal{K}$ is a composition of simple convex cones. Any convex optimization problem can be reformulated into a conic program by expressing it in its equivalent epigraph form. Specifically, Eq. (3) can be reformulated in the following way[1]:

$$\begin{aligned} \min_{(\boldsymbol{r},\boldsymbol{\beta},\boldsymbol{s},\boldsymbol{t},\boldsymbol{z}_1,\boldsymbol{z}_2)} & \sum_{i=1}^{n} t_i + \lambda r \\ \text{s.t. } & (z_{i1}, 1, u_i - t_i) \in K_{\exp}, \\ & (z_{i2}, 1, -t_i) \in K_{\exp}, \\ & z_{i1} + z_{i2} \le 1, && i = 1,\dots,n, \\ & -s_j \le D_j \le s_j, && j = 1,\dots,m, \\ & r - w_1 D_1 - \cdots - w_m D_m \ge 0. \end{aligned} \qquad (5)$$

[1] Note that there are other equivalent variants of rewriting the objective function in epigraph form, especially for the likelihood part (see e.g. Schwendinger et al. (2024)).

In the above formulation, the variables $\boldsymbol{r}, \boldsymbol{\beta}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{z}_1, \boldsymbol{z}_2$, and $\boldsymbol{u}$ are auxiliary variables. The initial problem (3) is thus rewritten as a conic programming problem as in Eq. (4) on an augmented set of variables $\Theta = (\boldsymbol{r}, \boldsymbol{\beta}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{z}_1, \boldsymbol{z}_2)$. Note that the formulation as CP only requires two distinct simple convex cones, namely the exponential cone for the first two lines of Eq. (5) and the linear cone for lines 3-5. These are defined as

$$\begin{aligned} K_{\text{exp}} &= \{x \in \mathbb{R}^3 | x_1 \geq x_2 \exp(x_3/x_2), x_1 > 0, x_2 > 0\}, \quad \text{and} \\ K_{\text{lin}} &= \{x \in \mathbb{R} | x \geq 0\} \end{aligned} \tag{6}$$

respectively. We omit further details on the reformulation procedure in this short paper and refer to Boyd and Vandenberghe (2004) for more information on rewriting convex problems into their epigraph forms.

There are various advantages to using the presented conic approach as opposed to using other methods, such as augmented Lagrangian procedures. First, a wide range of problems can be solved with only a few number of convex cones. For example, many of the most popular models from the GLM family can be modeled with only 3 types of cones: linear, second-order, and exponential cone (Schwendinger et al. (2024)). Furthermore, extensions to penalized versions of GLMs are straightforward by adding cone constraints (e.g. for the lasso penalty, only additional linear cones are required). This makes the conic approach flexible and easily extensible to specific situations. Second, from an algorithmic point of view, the conic solvers do not rely on specific starting values and provide reliable results. That is, when the algorithm signals success in finding an optimal solution, one can be very certain that the found optimum is a global optimum (Schwendinger et al. (2021)). Finally, recent advances in conic programming have led to the development of a variety of solvers, which routinely solve high-dimensional conic problems to proven optimality.

Problem (5) can be solved using a convex optimization solver of choice. The only requirement on the solver is that it can handle the above two types of convex cones. In `R`, there is a range of suitable solvers available via the optimization infrastructure `ROI` (Theußl et al. (2020)). We develop an `R` routine leveraging the flexibility of `ROI`.

Similar to other works (Zou (2006), Masarotto and Varin (2012)), we consider an adaptive variant, where weights w_j are placed on the penalty structure of the components, to avoid inconsistency and reduce bias in the estimation of the effects. We follow Masarotto and Varin (2012) and select the weights to be inversely proportional to the maximum likelihood estimates with a small ridge penalty

$$w_j = \left| D_j \tilde{\boldsymbol{\beta}}_\epsilon \right|^{-1}, \quad \tilde{\boldsymbol{\beta}}_\epsilon = \arg\min_{\boldsymbol{\beta}} \left\{ -\ell(\boldsymbol{\beta}) + \epsilon \sum_i \beta_i^2 \right\}. \tag{7}$$

Therefore, if the structural differences in the effects (represented by the matrix D) are small, a stronger penalization on the structure for these effects is employed. Finally, in order to select the tuning parameter λ, we compute the solution of the AGLL for a sequence of λ-values and minimize an AIC-type criterion of the form

$$\mathrm{AIC}(\lambda) = -2\ell(\boldsymbol{\beta}) + 2\mathrm{enp}(\lambda). \tag{8}$$

This idea follows Tibshirani and Taylor (2011), where $\mathrm{enp}(\lambda)$ is estimated as the number of distinct groups formed by a certain λ.

3 Simulation

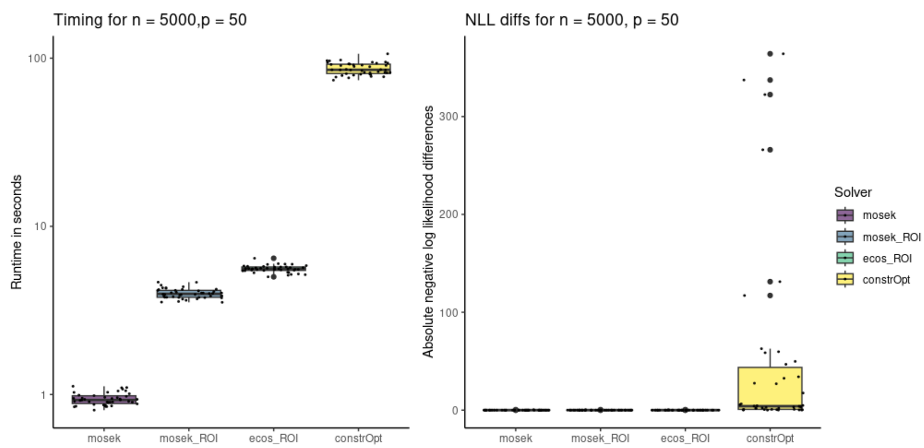

Fig. 1. Left: Comparison of runtime (in seconds) of different solvers. Right: Comparison of differences in negative log-likelihood between the best model and each other model. Displayed are 50 different runs of the simulation setup.

We briefly illustrate the efficiency of our approach in a simulation study. We simulate $n = 5000$ data points from a classical logistic regression model with $p = 50$ covariates. That is, the response variable is drawn from

$$y_i | x_{i1}, \ldots, x_{im} \sim \mathrm{Ber}(\pi_i), \quad m = 1, \ldots, \lfloor \frac{2}{3} p \rfloor, \tag{9}$$

where $\pi_i = 1/\big(1 + \exp(-\boldsymbol{\beta}^\top x_i)\big)$. The covariate matrix X is drawn from a Gaussian distribution and the coefficients are set such that 4 groups are present, and only 2/3 of the covariates are relevant, i.e. affect the outcome. In total, we compare 4 solvers, 3 of them are conic solvers. First, an implementation of the commercial solver from MOSEK via the `Rmosek`-package (MOSEK Aps (2022)). The other two leverage `ROI`, where we again use MOSEK via `ROI` as well as the open-source solver ECOS (Domahidi et al. (2013)). Finally, to compare the conic solvers to other approaches we use the `constrOpt` function from the `alabama`-package, which implements an augmented Lagrangian adaptive barrier minimization algorithm (Varadhan (2023)). We evaluate the methods based on runtime, as well as their ability to find the optimal solution for the objective function.

Figure 1 shows the results of our short simulation. It can be seen that the conic solvers are much more efficient in terms of runtime than the augmented Lagrangian solver (left panel of Figure 1). Using `ROI` produces a slight overhead, as seen by comparing the usage of the MOSEK solver directly versus via the package. However, the flexibility of `ROI` allows for easy usage of solvers like ECOS, which is slower than the commercial solver, but only by a small amount. It is important to mention that in our simulation with $p = 50$, the dimension of D is 1275×50. Compared to our application below, this is a rather small problem. Already in this setup, the runtime difference between conic solvers and the augmented Lagrangian method is drastic. In terms of finding the optimal solution (right panel of Figure 1) the conic solvers also outperform the `constrOpt` solver, with no notable difference between ECOS and MOSEK.

4 Ranking in sports

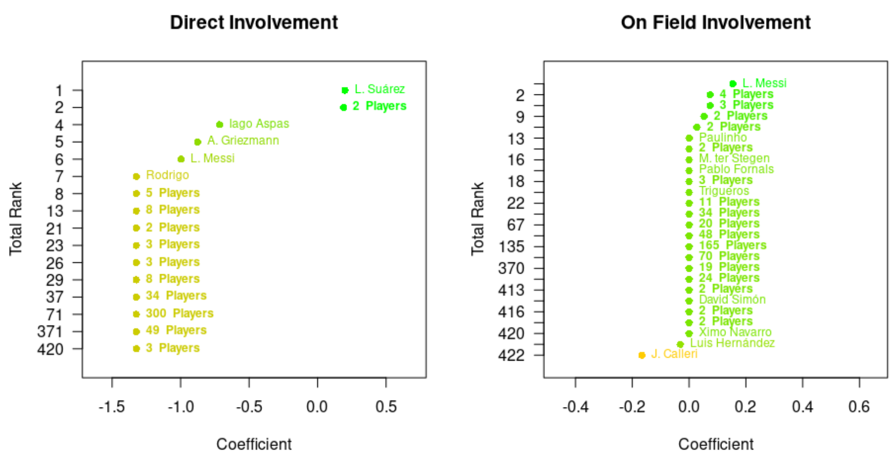

Fig. 2. Strength coefficients for direct (left) and off-ball (right) involvement. Selected groups are displayed with the corresponding group size. Groups of only one player are displayed by the name of the player.

We apply the AGLL to possessions in soccer, with the aim of ranking players based on their contributions to scoring goals. A possession is defined as a sequence of consecutive on-ball events, which ends either by the opponent team gaining possession or by an action of the referee. For each possession, two kinds of information are observed. First, we record which player is actively involved in a possession, and second, we record which player was on the field (offensively and defensively) during a possession, but not directly involved. This data structure allows to separate effects of being part of a possession, from effects of off-ball actions, both of which have substantial impact in soccer. To be more concrete, the units of observation are possessions, where each possession can end in a goal or no goal (binary outcome). The player configuration for each possession is collected in our design matrix X. In our data set, we have 422 players, and therefore 844 total columns of direct and indirect involvements. For ranking players, it then makes sense to differentiate between direct and indirect (off-ball) involvement. That is, we do not want to consider the classical ranking lasso penalty, where effects of direct and indirect involvement would be compared to each other. Instead, the penalty matrix D should account for the fact that we want to compare the direct effects of one player only to the direct ones of the other players and the same for indirect effects. This results in the following form for D:

$$D = \begin{pmatrix} D_{\text{dir}} & \mathbf{0} \\ \mathbf{0} & D_{\text{indir}} \end{pmatrix}. \tag{10}$$

For each of the two components, we are interested in applying a ranking penalty. That is, $D_{\text{dir}} = D_{\text{indir}}$, where

$$D_{\text{dir}} = (A, B, C) = \begin{pmatrix} A_1 & B_1 & C_1 \\ \vdots & \vdots & \vdots \\ A_{N_{\text{dir}}-1} & B_{N_{\text{dir}}-1} & C_{N_{\text{dir}}-1} \end{pmatrix}. \tag{11}$$

The A_i are matrices of zeros of dimension $(N_{\text{dir}} - i) \times (i - 1)$, each B_i is a vector of ones of length N_{dir}, and each C_i is the negative identity matrix of dimension $N_{\text{dir}} - 1$. In our example $N_{\text{dir}} = 422$, and thus the row-dimension of D_{dir} is 88831. Therefore, the total dimension of D is 177662×844. It is evident that an efficient procedure is necessary to solve the AGLL problem, and our conic programming approach works well and fast in such scenarios. The results of our analysis are displayed in Figure 2. These are in line with the intuition on ranking soccer players. They reflect the difficulty of ranking soccer players while simultaneously emphasizing the necessity of partial rankings, as most players are grouped in similar strength groups, and only a few stand out.

5 Summary

In this work, we consider the generalized lasso penalty for logistic regression. We transform this convex problem into its conic form and develop a framework for solving it using conic programming. The presented approach is flexible and easily extensible and simulations show that it is fast and robust. Finally, we use the methodology to rank soccer players into strength groups.

References

1. Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.
2. Domahidi A., Chu E., Boyd S. (2013). ECOS: an SOCP solver for embedded systems. In: *European control conference (ECC)*, pp 3071–3076.
3. Masarotto, G. and C. Varin (2012). The ranking lasso and its application to sport tournaments *The Annals of Applied Statistics*, 6, 1949–1970.
4. MOSEK Aps (2022). Rmosek: The R-to-MOSEK optimization interface *R package version 10.0.25*.
5. Schwendinger, F., B. Grün, and K. Hornik] (2021). A comparison of optimization solvers for log binomial regression including conic programming. *Computational Statistics*, 36 (3), 1721–1754.
6. Schwendinger, B., F. Schwendinger, and L. Vana (2024). Holistic Generalized Linear Models. *Journal of Statistical Software*, 108.7, p. 1–49.
7. Theußl, S., F. Schwendinger, and K. Hornik (2020). ROI: An extensible R optimization infrastructure. *Journal of Statistical Software*, 94 (15), 1–64.
8. Tibshirani, R. J. and J. Taylor (2011). The solution path of the generalized lasso. *The Annals of Statistics*, 39 (3), 1335–1371.
9. Varadhan R. (2023). alabama: Constrained Nonlinear Optimization. *R package version 2023.1.0*.
10. Zou, H (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101, 1418–1429.