Overlapping Group Lasso Via ADMM in Python

Daniel Kessler

December 3, 2018

Abstract

In this report, we present a brief summary of the overlapping group Lasso, showing how it can be motivated as an extension first of the regular lasso to a group setting, and then as a further generalization thereof. We then provide some background on the Alternating Direction Method of Multipliers (ADMM) algorithm, and show why it is a reasonable choice for solving overlapping group Lasso problems. Next, we derive the ADMM algorithm for overlapping group Lasso, and present a software implementation in python that implements this algorithm. Finally, we show some experimental results on synthetic data and comment on parameter tuning, as well as alternative formulations of the ADMM algorithm as applied to this problem.

1 Note to Reviewer

The implementation of the procedure was incomplete at the time of submission, and thus simulation results are absent from this version of the draft. The code and this report are both on github at `http://github.com/dankessler/608a-project`, and the reviewer is asked to please conduct their review on the latest version of this PDF[1], which is available directly at this link.

2 Notation

First, we fix notation. Much of our derivations are based on [1], we will use different notation than is typically deployed in more statistics-oriented treatments of the Lasso. Let $A \in \mathbb{R}^{m \times n}$ be a (fixed) design matrix, with m observations and n covariates, and $b \in \mathbb{R}^m$ a vector of observations. We assume that our y follows $y = X\beta + \epsilon$, where $\beta \in \mathbb{R}^p$ is an unknown weight vector, and ϵ are independent and identically distributed errors (for a simple case, we can take them to follow $\mathcal{N}(0, \sigma^2)$ for some fixed, but unknown, σ^2. When norms are not otherwise specified, they are taken to be the 2-norm, i.e., $\|\cdot\| \triangleq \|\cdot\|_2$. We will generally be interested in minimizing the least squares loss, i.e., finding $\hat{x} \in \operatorname{argmin}_x \|Ax - b\|$. Note: A great deal of the treatment below, including

[1] which by the time you read this, is hopefully more complete

that of the lasso, group lasso, overlapping group lasso, and ADMM is taken from [1]. We explicitly cite this text at key points, but we do not cite *every* claim which is based on [1] to avoid cluttering the text.

3 Background: Overlapping Group Lasso

In order to introduce the overlapping group Lasso, we will first discuss the regular lasso and then show how it can be extended to the (non-overlapping) group lasso setting. The Lasso is a highly popular method that is especially useful in high dimensional settings, i.e., where $n \gg m$. In this setting (presuming A is of full rank), the OLS estimate is no longer uniquely determined, as the there exist infinitely many candidate $\hat{x}$ that yield zero loss. Instead, one can instead minimize a *regularized problem* in order to obtain a *sparse solution*. While various (essentially equivalent) formulations of the lasso objective exist, for our purposes we will define the primal lasso problem as

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1, \tag{1}$$

where $\lambda > 0$ determines the amount of regularization. As $\lambda \to 0$, we see that (1) becomes the OLS problem, and as $\lambda \to \infty$, $x \to 0$. One advantage of the lasso is that it will typically recover a sparse solution, i.e., a solution where the minimizing $\hat{x}$ has many entries that are identically 0.

In the setting where covariates can be organized into groups, as may be natural in many applied settings (e.g., where the covariates are gene expression levels, and genes can be organized based on chromosome or location), we may wish not to simply select useful covariates, but instead to select useful *groups* of covariates. This motivates the use of *group lasso* [5], where we replace the objective in (1) with

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\sum_{i=1}^{N}\|x_i\|_2, \tag{2}$$

where x_i is a subvector of x containing only the coefficients corresponding to the i'th group, with $i \in [N]$. Note that when extending to the group lasso, the penalty term no longer involves the 1-norm but instead has the 2-norm. Although this may seem surprising, the 1-norm is separable, i.e., $\left\|\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}\right\|_1 = \|a_1\|_1 + \|a_2\|_1$, and this would devolve back to the original lasso. Critically, the 2-norm in the regularizer is not squared, which yields an analogous geometry to the lasso, with singularities corresponding to solutions that are group-sparse (see [5], Fig 1 for a helpful illustration of this phenomena). Note that for a singleton vector, $\|a\|_1 = \|a\|_2$, so when $N = n$, i.e., each feature is alone in its own atomic group, we can rewrite (2) as

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\sum_{i=1}^{N}\|x_i\|_2 = \min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\sum_{i=1}^{N}\|x_i\|_1 \quad = \min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1,$$

and we can recover the original lasso formulation in (1) as a special case of the group lasso.

Finally, the *group lasso* can be further extended to accommodate *overlapping groups* in the "overlapping group lasso" [6, 3]. In this setting, rather than partitioning x into disjoint subvectors, we let $G_i, i = [N]$ be an index set holding the indices of coefficients corresponding to the i'th group, i.e., x_{G_i} is a vector of coefficients for group i, x_{G_j} is a vector of coefficients for group j, and it may be the case that $G_i \cap G_j \neq \emptyset$. As a toy example, suppose $n = 3$, and in this simple setting we have two groups, with $G_1 = \{1, 2\}, G_2 = \{2, 3\}$, such that x_2 is common to both x_{G_1} and x_{G_2}, i.e., $G_1 \cap G_2 = \{2\}$. In this setting, the overlapping group lasso objective is given by

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda \sum_{i=1}^{N} \|x_{G_i}\|_2. \tag{3}$$

The geometry of this problem is rather complicated, and there is some work (e.g., [2]) that proposes addressing the overlapping group lasso through latent variables, in essence, performing variable duplication to render the problem non-overlapping, and then using the standard group lasso formulation in (2) to solve the problem. However, in the present work we will focus on directly optimize the objective given in (3), although as we shall see in 5, our algorithmic approach will involve a sort of variable duplication, but with an update step that pulls our duplicated variables back toward one another.

4 Background: ADMM

The Alternating Direction Method of Multipliers (ADMM) is an algorithmic approach to optimization well suited to solving problems that can be decomposed as the sum of two problems in distinct variables, subject to linear constraints. The background we provide here will closely follow [1], as this was our primary resource when endeavoring to learn the material. Our development here will be terse and limited, and we refer the reader to [1], as the exposition given below chiefly consists of key highlights from this very useful text.

ADMM is formulated to solve problems structured as

$$\begin{aligned} \min_{x,z} \quad & f(x) + g(z) \\ \text{subject to} \quad & Ax + Bz = c \end{aligned}. \tag{4}$$

It is closely related to the method of multipliers (a brief background is given in [1]) and proceeds by first constructing an augmented Lagrangian

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2, \tag{5}$$

where y is a dual variable, and the last term is the "augmenting" piece. Although augmenting may seem unnatural at first, we note that when the linear

constraints are satisfied, this last term is identically 0 and thus inconsequential for the objective function at the optimum, and its inclusion makes the use of the `prox` operator natural during the optimization. ADMM is an iterative procedure, which given some initial values for x, z, y, proceeds as

$$x^{k+1} \leftarrow \operatorname*{argmin}_x L_\rho(x, z^k, y^k) \tag{6}$$

$$z^{k+1} \leftarrow \operatorname*{argmin}_x L_\rho(x^{k+1}, z, y^k) \tag{7}$$

$$y^{k+1} \leftarrow y^k + \rho(Ax^{k+1} + Bz^{k+1} - c). \tag{8}$$

Of course, the rub lies in actually solving the subproblems given in (6)(7), but using the augmented Lagrangian L_ρ makes this tractable for certain problems.

5 Overlapping Group LASSO Via ADMM

A brief sketch of an ADMM approach for overlapping group lasso is given at the end of §6.4.2 of [1], which we expand upon here. How can we rewrite (3) in a form compatible with ADMM? In a similar spirit to the "latent" approach to overlapping group lasso [2, 4], we will create many *new* variables which in a strict sense do not overlap, but then use linear equality constraints to impose the requirement that they equal a common anchoring global variable, which will indirectly enforce equality in the shared components. Slightly overloading the meaning of x_i from the non-overlapping group lasso treatment, now let $x_i \triangleq x_{G_i}, x_i \in \mathbb{R}^{|G_i|}$, and now introduce a new variable $z \in \mathbb{R}^n$, which will play the role of an "anchoring variable." For notational convenience, let $\tilde{z}_i \triangleq z_{G_i}$, i.e., the components of z corresponding to group i for $i \in [N]$. Written in a form compatible with ADMM, we now have the objective

$$\begin{aligned} \min_{z, x_i, i \in [N]} \quad & \frac{1}{2}\|Az - b\|_2^2 + \lambda \sum_{i=1}^N \|x_i\|_2, \\ \text{such that} \quad & x_i - \tilde{z}_i = 0, \quad \forall i \in [N] \end{aligned} \tag{9}$$

which with minimal rearrangement (i.e., swapping the first and second terms) is written in a form compatible with ADMM. In addition, the term involving the sum of the x_i is now decomposable, and thus the update for the x_i can be done in parallel, which may be useful if the number of groups is very large. Now, our task is to give explicit forms for steps (6)(7). Our update can be performed as

$$x_i^{k+1} \leftarrow \operatorname*{argmin}_{x_i} (\lambda \|x_i\|_2 + \left(y_i^k\right)^T (x_i - \tilde{z}_i^k) + \frac{\rho}{2}\|x_i - \tilde{z}_i^k\|_2^2) \tag{10}$$

$$z^{k+1} \leftarrow (A^T A + \rho I)^{-1}(A^T b + \rho(\bar{x}^{k+1} - \bar{y}^k)) \tag{11}$$

$$y_i^{k+1} \leftarrow y_i^k + x_i^{k+1} - \tilde{z}_i^{k+1}, \tag{12}$$

where $\bar{x}, \bar{y}$ are obtained by averaging over i at the relevant components. The operation at (10) is precisely the proximal operator, which for group lasso is the

vector soft thresholding operation, as given in §6.4.2, i.e.,

$$x_i^{k+1} = (1 - \frac{\lambda}{\rho}\|\tilde{z}_i^{k+1} + u^k\|_2^{-1})_+(\tilde{z}_i^{k+1} + u^k) \tag{13}$$

We summarize the approach in Algorithm 1.

Algorithm 1 Group LASSO Via ADMM

Require: $x^0 \in$

6 Software Implementation

We implemented the procedure of Algorithm 1 in python 3.7. The package, which is for now incomplete, is available on github at `http://github.com/dankessler/608a-project`.

7 Experimental Results

References

[1] Stephen Boyd et al. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (July 26, 2011), pp. 1–122. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000016. URL: http://www.nowpublishers.com/article/Details/MAL-016.

[2] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. "Group Lasso with Overlap and Graph Lasso". In: *ICML2009*. ICML '09. New York, NY, USA: ACM, 2009, pp. 433–440. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553431. URL: http://doi.acm.org/10.1145/1553374.1553431.

[3] Julien Mairal et al. "Network Flow Algorithms for Structured Sparsity". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., 2010, pp. 1558–1566. URL: http://papers.nips.cc/paper/3965-network-flow-algorithms-for-structured-sparsity.pdf.

[4] Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. "Group Lasso with Overlaps: the Latent Group Lasso approach". In: *arXiv:1110.0413 [cs, stat]* (Oct. 3, 2011). arXiv: 1110.0413. URL: http://arxiv.org/abs/1110.0413.

[5] Ming Yuan and Yi Lin. "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), pp. 49–67. ISSN: 1467-9868. DOI: 10.1111/j.1467-9868.2005.00532.x. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2005.00532.x/abstract.

[6] Peng Zhao, Guilherme Rocha, and Bin Yu. "The composite absolute penalties family for grouped and hierarchical variable selection". In: *The Annals of Statistics* 37.6 (Dec. 2009), pp. 3468–3497. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/07-AOS584. URL: http://projecteuclid.org/euclid.aos/1250515393.