On Group-Wise ℓ_p Regularization: Theory and Efficient Algorithms

Duc-Son Pham[a]

[a]*Department of Computing, Curtin University, Perth, Australia.*
Email: dspham@ieee.org

Abstract

Following advances in compressed sensing and high-dimensional statistics, many pattern recognition methods have been developed with ℓ_1 regularization, which promotes sparse solutions. In this work, we instead advocate the use of ℓ_p ($2 \geq p > 1$) regularization in a group setting which provides a better trade-off between sparsity and algorithmic stability. We focus on the simplest case with squared loss, which is known as group bridge regression. On the theoretical side, we prove that group bridge regression is uniformly stable and thus generalizes, which is an important property of a learning method. On the computational side, we make group bridge regression more practically attractive by deriving provably convergent and computationally efficient optimization algorithms. We show that there are at least several values of p over (1,2) at which the iterative update is analytical, thus it is even suitable for large-scale settings. We demonstrate the clear advantage of group bridge regression with the proposed algorithms over other competitive alternatives on several datasets. As ℓ_p-regularization allows one to achieve flexibility in sparseness/denseness of the solution, we hope that the algorithms will be useful for future applications of this regularization.

Keywords: ℓ_p regularization, convex optimization algorithms, ADMM, FISTA, algorithmic stability, Lasso, group Lasso, bridge regression, group bridge regression, splice detection

1. Introduction

Regularization is an important issue in pattern recognition for developing learning algorithms with high predictive power. In this work, we consider algorithms

for solving a regularization problem of the following form

$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + g(\mathbf{x}). \tag{1}$$

Here, we restrict our attention to the squared loss function and norm-based regularization $g(\mathbf{x})$. Extensions to other convex loss functions, such as logistic, may be obtained similarly.

In learning theory, such a regularization is known to avoid over-fitting and thus it allows the developed algorithm to generalize. Regularization has been an important principle in machine learning and statistics [1], especially when one is faced with increasing challenges of massive data-sets wherein the dimension can be very large [2]. Recently, with the explosive growth of interest in compressed sensing [3, 4] and high-dimensional statistics [2], a great deal of literature has been devoted to study the learning problem with ℓ_1 regularization, i.e. $g(\mathbf{x}) = \lambda\|\mathbf{x}\|_1 = \lambda\sum_i |x_i|$. The theoretical arguments for such a choice have been put forward in, for example, [3, 4, 5, 6]. It is known that ℓ_1 regularization promotes sparsity, which is conceived to be desirable in many learning problems. As such, optimization algorithms have been specifically developed to solved the Lasso-type problem [5] efficiently. The compressed sensing repository[1] contains numerous references on optimization algorithms for solving compressed sensing recovery via ℓ_1 regularization. Consequently, the literature has seen an increasing number of applications of ℓ_1 regularization, such as face recognition [7], graph optimization [8], object categorization [9].

As structure constraints are shown to be beneficial to learning algorithms [10], the statistics literature has also seen an extension of the basic Lasso scheme to situations where grouped variables are available, known as group Lasso [11], [12], [2]. In this setting, the variable vector $\mathbf{x}$ is naturally divided into G groups

$$\mathbf{x} = [\mathbf{x}_1;\ \mathbf{x}_2;\ \ldots;\ \mathbf{x}_G] \qquad \mathbf{A} = [\mathbf{A}_1\ \mathbf{A}_2\ \ldots,\ \mathbf{A}_G] \tag{2}$$

$$\mathbf{A}\mathbf{x} \ = \ \sum_{i=1}^{G} \mathbf{A}_i\mathbf{x}_i. \tag{3}$$

Encouraging applications that exploits the group information can be found in a wide range of problems from image tagging [13] to face recognition [6].

However, there are cases where ℓ_1 regularization does not achieve competi-

[1] http://dsp.rice.edu/cs

tive results as other dense regularization [14]. Theoretically, Xu *et al.* [15] has established that certain sparse algorithms, including Lasso and group Lasso, are not algorithmically stable, an important property of a good learning algorithm. Whilst not being algorithmically stable does not mean that sparsity algorithms do not generalize, it implies that they can potentially have poor predictive performance in the worst case scenarios. More recently, it has been shown in [16] in the context of multiple kernel learning that dense solution via ℓ_p-norm performs better than sparse solutions and achieves state-of-art performance over a wide range of problems. Likewise, [17] found that ℓ_p regularization with group settings attains best compromise between prediction and robustness for $p \in [1.5, 2]$. It appears that ℓ_p regularization is an alternative that provides a natural trade-off between sparsity and stability [15]. However, ℓ_p regularization is still of infrequent use in practice, especially in the group setting. This could be of two reasons, both theoretically and computationally. On the theoretical side, though there are some published works in the statistics literature such as [18, 19], little is known about the generalization property of group bridge regression. On the computational side, efficient algorithms for ℓ_p regularization in general, especially in large-scale problems, seem to be lacking compared with ℓ_1 regularization. We note that ℓ_p regularization is strictly convex for $p > 1$, and thus gradient techniques can be used. However, they tend to have rather poor convergence property especially when p is close to 1 (which we demonstrate subsequently).

In this work, we further advocate the use of ℓ_p regularization in a group setting. For the squared loss, this is known as group bridge regression [18]. Though it is not new, we revisit this powerful regression method in the large-scale pattern recognition context and make two contributions. Theoretically, we prove that group bridge regression is also algorithmically stable, and thus it generalizes. Computationally, we develop the novel and efficient algorithms under two powerful optimization frameworks: alternative directions method of multipliers (ADMM) [20] and fast iterative shrinkage thresholding (FISTA) [21]. We show that there are values of p distributed over the range [1,2] where group bridge regression have *analytical* solutions for the iterative updates, just like the Lasso. This implies one can achieve varying degrees of sparseness in the solution efficiently with the proposed algorithms. This is particularly useful in cases where compressible data is present [22]. When analytical updates are not available, we propose an algorithm to compute the updates with an efficient warm-start strategy. Whilst the studied examples in this work subsequently show the advantage of ℓ_p regularization over ℓ_1, note that we do not claim it is always better. There will be cases where ℓ_1 might be more suitable. What we try to convey here is an al-

ternative method for pattern recognition, which clearly allows flexibility between achieving sparse or dense solutions with the most desirable property of a learning algorithm.

The paper is organized as follows. Section II establishes the algorithmic stability of ℓ_p regularization in group bridge regression settings. In Section III, we derive efficient ADMM- and FISTA-based algorithms for solving group bridge regression. Section IV examines the numerical properties of the proposed algorithms and demonstrate the competitive advantage of group bridge regression over other sparse alternatives on a synthetic dataset and a real-world splice detection problem. Finally, Section V concludes.

The Matlab implementation of all developed methods is made publicly available at the following website `https://sites.google.com/site/dspham`.

2. Algorithmic Stability with ℓ_p-norm Regularization

Algorithmic stability [23] is one powerful concept for assessing the predictive power of a supervised learning method. We now show that group bridge regression of problem (1) with

$$g(\mathbf{x}) = \lambda \sum_{i=1}^{G} \|\mathbf{x}_i\|_2^p = \lambda \|\mathbf{x}\|_{\ell_2/\ell_p}, \quad p \in (1,2], \tag{4}$$

is indeed algorithmically stable. Our approach is based on the key result in [24], and we tailor it to the group setting.

First, we briefly revisit the common setting in supervised learning, where a set of data points $\mathbf{z} = \{(\mathbf{a}_1, y_1), \ldots, (\mathbf{a}_n, y_n)\}$, and $\mathbf{a}_i \in \mathbb{R}^d$. The aim is to learn a function f from $\mathbf{z}$ that allow us to predict y given a future $\mathbf{a}$. Here, for the formulation (1) the function to be learnt is linear $f(\mathbf{a};\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ and the squared loss function $V(y_1, y) = \frac{1}{2}(y_1 - y)^2$. Up to a scaling by a factor of $1/n$, the formulation (1) is known in learning theory as empirical risk minimization where the first term essential represents the empirical risk $R_{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} V(\mathbf{a}_i^T\mathbf{x}, y_i)$. An algorithm is said to be consistent if the empirical risk converges asymptotically to the risk, i.e.

$$\lim_{n\to\infty} R_{\mathbf{z}} = R = E_{\mathbf{a}}(1/2)(y - \mathbf{a}^T\mathbf{x})^2,$$

assuming bounded risk. For the finite sample case, the learning theory is interested in the bound on the deviation of the empirical risk from the risk. In algorithmic stability theory[23], an algorithm is said to be uniformly β-stable if there exists a

finite β that upper bounds the maximum deviation in the loss due to replacement of the sample $\mathbf{z}$ with any possible $\mathbf{z}'$ from the same distribution. Under regularity conditions on the loss function, including convexity, boundedness at $\mathbf{0}$, and L-Lipschitz in the first variable (which are met by the actual squared loss considered here), [23] showed that the bound is

$$|R_{\mathbf{z}} - R| \leq \beta + (2n\beta + L\sqrt{\kappa}\tau + B)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{5}$$

with the probability of at least $1-\delta$. Here, κ is an upper bound on the feature functions in the functional space of f, i.e. $\kappa = \sup \|\mathbf{x}\|_2^2$, and B is an upper bound on the loss function when the first variable is zero, i.e. $B = \frac{1}{2}\max y^2$, and L is the Lipschitz constant of the lost function $V(y_1, y)$ in terms of the first variable y_1 subject to the regular conditions, i.e. $L = y_{\max} - y_{\min}$. Detail can be found in [23, 24]. Clearly, when $\beta = o(n^{-1/2})$ then stability implies generalization. Though uniform stability appears rather strict, it requires no further assumptions on the data than other weaker notion of stability in the literature [24].

Though algorithmic stability is a powerful tool to characterize a learning algorithm, there was not an easy way to verify uniform stability for a particular method until recently when Wibisono *et al.* [24] discovered a sufficient condition to do so. Consider the class of norm regularization where $g(\mathbf{x}) = \lambda P(\mathbf{x})$ where λ is regularization parameter and $P(\mathbf{x})$ is some suitable norm. Denote as $\mathbf{x}_{\mathbf{z}}$ and $\mathbf{x}_{\mathbf{z}^j}$ respectively the solution of the regularized empirical risk minimization on original data $\mathbf{z}$ and when the jth sample is replaced with another from the same distribution. It was established in [24] that:

Theorem 2.1. *Suppose that for some constant $C > 0$ and $\xi > 1$, the penalty function satisfies*

$$P(\mathbf{x}_{\mathbf{z}}) + P(\mathbf{x}_{\mathbf{z}^j}) - 2P\left(\frac{\mathbf{x}_{\mathbf{z}} + \mathbf{x}_{\mathbf{z}^j}}{2}\right) \geq C\|\mathbf{x}_{\mathbf{z}} - \mathbf{x}_{\mathbf{z}^j}\|_2^{\xi}$$

then the regularization is uniformly β-stable with $\beta = \left(\frac{L^{\xi}\kappa^{\xi/2}}{n\lambda C}\right)^{\frac{1}{\xi-1}}$.

Using this important result and following the strategy in [24], we also establish algorithmic stability for group bridge regression as follows:

Theorem 2.2. *For group bridge regression, there holds*

$$P(\mathbf{x}_\mathbf{z}) + P(\mathbf{x}_{\mathbf{z}^j}) - 2P\left(\frac{\mathbf{x}_\mathbf{z} + \mathbf{x}_{\mathbf{z}^j}}{2}\right) \geq \frac{p(p-1)}{4}\left(\frac{B}{\lambda}\right)^{\frac{p-2}{p}} \|\mathbf{x}_\mathbf{z} - \mathbf{x}_{\mathbf{z}^j}\|_2^2$$

thus it is uniformly β-stable with $\beta = \frac{4L^2\kappa}{n\lambda p(p-1)}\left(\frac{\lambda}{B}\right)^{\frac{p-2}{p}}$.

The proof of this result is detailed in the Appendix. Complementing the existing knowledge in the statistics literature [18] in estimation context, this results further justifies group bridge regression in the prediction context.

3. Optimization Algorithms

We now derive ADMM and FISTA algorithms to solve group bridge regression. Despite the differences between the frameworks, we show that there is a fundamental and common convex optimization sub-problem. For some values of the bridge order p, this sub-problem has analytical solution just like Lasso, which implies computational advantage. For other cases, we exploit properties of the sub-problem to construct an efficient algorithm.

3.1. ADMM algorithm

Alternating direction method of multipliers (ADMM) is a simple but powerful framework in optimization, which is suited for today's large-scale problems arising in machine learning and signal processing. The method was in fact developed a long ago before advanced computing power was available, and re-discovered many times under different perspectives. Recently, [20] has unified the framework in a simple and concise explanation. Consider the problem (1) with ℓ_p regularization. As the variables are coupled due to the smooth loss, this makes it even harder when combine with the regularization term. In principle, the problem is easier to tackle if the variables can be decoupled, so that the problem can be solved element-wise or group-wise. Using a clever trick, the ADMM framework suggests to separate the regularization term from the smooth term by introducing an additional variable $\mathbf{z}$, which is tied to the original variable via an affine constraint:

$$\min_{\mathbf{x},\mathbf{z}} \quad f(\mathbf{x}) + h(\mathbf{z}) \quad \text{s.t} \quad \mathbf{x} - \mathbf{z} = 0. \tag{6}$$

Here, for group bridge regression we have $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$, and $h(\mathbf{z}) = \lambda \sum_{i=1}^{G} \|\mathbf{z}\|_2^p$. For this type of regularized objective function, ADMM considers the following augmented Lagrangian

$$L_\rho(\mathbf{x},\mathbf{z},\mathbf{y}) = f(\mathbf{x}) + h(\mathbf{z}) + \mathbf{y}^T(\mathbf{x}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{x}-\mathbf{z}\|_2^2. \tag{7}$$

Here, ρ is the parameter associated with the augmentation $\frac{\rho}{2}\|\mathbf{x}-\mathbf{z}\|_2^2$, and this is to improve the numerical stability of the algorithm. The strategy for minimizing this augmented Lagrangian is iterative updating of the primal and dual variables. With a further normalization on the dual variable $\mathbf{u} = (1/\rho)\mathbf{y}$, it is shown [20] that the updates for the parameters are

$$\begin{aligned}
\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x}-\mathbf{z}^k+\mathbf{u}^k\|_2^2 \right\} && (8)\\
\mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} \left\{ h(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{x}^{k+1}-\mathbf{z}+\mathbf{u}^k\|_2^2 \right\} && (9)\\
\mathbf{u}^{k+1} &= \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1}. && (10)
\end{aligned}$$

The iterative updates are repeated until the following primal and dual residuals are sufficiently small

$$\begin{aligned}
\mathbf{r}_p^k &= \mathbf{x}^k - \mathbf{z}^k && (11)\\
\mathbf{r}_d^k &= \rho(\mathbf{z}^k - \mathbf{z}^{k-1}). && (12)
\end{aligned}$$

For the squared loss being considered, it is noted that the update step for $\mathbf{x}$ is exact

$$\mathbf{x}^{k+1} = (\mathbf{A}^T\mathbf{A}+\rho\mathbf{I})^{-1}\left(\mathbf{A}^T\mathbf{y}+\rho(\mathbf{z}^k-\mathbf{u}^k)\right). \tag{13}$$

Furthermore, $\mathbf{A}^T\mathbf{A}+\rho\mathbf{I}$ is fixed and thus its inversion can be pre-computed for better efficiency. The remaining challenge is to find the update step for $\mathbf{z}$ via the following sub-problem

$$\begin{aligned}
\mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} \left\{ \frac{1}{2}\|\mathbf{v}-\mathbf{z}\|_2^2 + \frac{\lambda}{\rho}\sum_{g=1}^{G}\|\mathbf{z}_g\|_2^p \right\} && (14)\\
&= \arg\min_{\mathbf{z}} \sum_{g=1}^{G} \left\{ \frac{\lambda}{\rho}\|\mathbf{z}_g\|^p + \frac{1}{2}|\mathbf{v}_g-\mathbf{z}_g\|^2 \right\} && (15)
\end{aligned}$$

where $\mathbf{v} = \mathbf{x}^{k+1} + \mathbf{u}^k$ for the ADMM case. Here, we have dropped the subscript for notational simplicity. Note that this cannot be decomposed further into element-wise form due to the coupling induced by $\|\mathbf{z}\|_2^p$. We now discuss how to solve this problem for $2 \geq p > 1$. We have the following result.

Lemma 3.1. *Let* $\mathbf{e} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$ *be the direction of* $\mathbf{v}$*, and let* $v = \|\mathbf{v}\|_2$*. Then the solution of (15) has the form* $\mathbf{z} = \eta\mathbf{e}$ *where* $\eta \geq 0$ *is the minimizer of*

$$\frac{\lambda}{\rho}|\eta|^p + \frac{1}{2}(v-\eta)^2. \tag{16}$$

The significance of this result is that it converts a multidimensional optimization problem (15) to an univariate optimization problem (16). This result can be proved by simple geometrical arguments. Indeed, denote $\mathbf{z}^*$ as the solution of (15), then we consider all points $\mathbf{z}$ such that $\|\mathbf{v}-\mathbf{z}\|_2 = \|\mathbf{v}-\mathbf{z}^*\|_2 = R$. It turns out that these points are lying on the ball with center at $\mathbf{v}$ and radius R. Among these points, only the point that satisfies $\mathbf{z} = \eta\mathbf{e}$, i.e. intersection of the ball and the vector $\mathbf{v}$, will have minimum ℓ_2 norm, which minimizes the second term in (15), then (16) follows immediately.

We shall discuss numerical algorithms for solving (16) subsequently. Next, we show that (16) is also a central problem in the FISTA algorithm.

3.2. FISTA algorithm

Another approach to effectively decouple the variables when solving bridge regression optimization problems is to directly approximate the loss function by a decoupled quadratic function. This approach was proposed by [21], which also shares the same philosophy as an unpublished work of [25]. The name of the method is actually motivated from the Lasso problem, but its general principle can be used for the more general case of bridge regression. FISTA exploits two key strategies:

- *Decoupling variables:* Instead of dealing with the original loss function $\frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$ where variables are coupled through $\mathbf{A}$, it iteratively solves a series of decoupled problems of the form

$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{v}-\mathbf{x}\|_2^2 + \lambda \sum_{g=1}^{G} \|\mathbf{x}_g\|_p^p. \tag{17}$$

  where $\mathbf{v} = \mathbf{z}^k - (\mathbf{A}^T\mathbf{A}\mathbf{z}^k - \mathbf{A}^T\mathbf{y})/L$, $L = \lambda_{\max}(\mathbf{A}^T\mathbf{A})$. Although being similar to the $\mathbf{z}$ step in ADMM, FISTA always works on the original *primal*

variables. This is similar to (14) and thus shares the same core problem (16) with ADMM.

- *History update:* FISTA also uses historical points to speed up convergence

$$\mathbf{z}^{k+1} = \mathbf{x}^k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}^k - \mathbf{x}^{k-1}), \tag{18}$$

$$t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2} \tag{19}$$

For further detail of FISTA, please see [21]. Next, we discuss technical aspects of solving (16).

3.3. Solving the sub-problem (16)

As the problem is strictly convex for $2 \geq p > 1$, the solution of (16) satisfies

$$z + \gamma p|z|^{p-1}\text{sign}(z) = v, \tag{20}$$

where $\gamma = \lambda/\rho$ for ADMM algorithms and $\gamma = \lambda/L$ for FISTA algorithms. Denotes as $z^* = \Omega(v, p; \gamma)$ the solution of this problem as a function of v, p and γ. We note that the left hand side of (20) is a monotonically increasing function of z, and hence for a given v it has exactly one solution. For some values of p, (20) has analytical solutions. Otherwise, (20) needs to be solved numerically with one-dimensional techniques such as bi-section. To facilitate fast one-dimensional search, one needs to specify a suitable interval for the search. A tight interval or a starting point close to the true solution is the key to speeding up (20) when solved numerically. In what follows, we describe properties of the solution $\Omega(v, p; \gamma)$ and their implications to search strategies:

- *Sign consistency*: The solution must have the same sign as v_i, i.e. $\Omega(v, p; \gamma) = \text{sign}(v)\Omega(|v|, p; \gamma)$, with a note that $\Omega(0, p; \gamma) = 0$. This implies that when solving for each z_i, we can consider the magnitude of v_i for simplicity, and adjust the sign latter. Consequently, this also implies that $\Omega(v, p; \gamma) = -\Omega(-v, p; \gamma)$.

- *Monotonicity:* The function $\lambda p|z|^{p-1}\text{sign}(z) + \rho z$ is monotonically increasing and odd-symmetric. Thus, if $|v_i| \leq |v_j|$ then $\Omega(|v_i|, p; \gamma) \leq \Omega(|v_j|, p; \gamma)$.

- *Bounded range:* Due to the monotonicity property, it is easy to show that

$$0 < \Omega(|v_i|, p; \gamma) < \min\left\{ |v_i|, \left(\frac{\rho|v_i|}{\lambda p}\right)^{\frac{1}{p-1}} \right\}.$$

  We shall improve the bound subsequently.

- *Analytical solution:* for few cases where the equilibrium equation can be posed as finding roots of a polynomial. This happens when $k = 1/(p-1)$ is an integer and thus the following form is equivalent to (20) with variable $t \geq 0$ such that $z = \text{sign}(v)t^k$:

$$t^k + p\gamma t - |v| = 0. \tag{21}$$

  For some small values of k, we can derive analytical solution as follows:

  $p = 2$: $t = \frac{|v|}{1+2\gamma}$.
  $p = 3/2$: $t = -\frac{3\gamma}{4} + \frac{1}{2}\sqrt{\frac{9\gamma^2}{4} + 4|v|}$.
  $p = 4/3$: $t = \left(\frac{|v|}{2} + \sqrt{\frac{|v|^2}{4} + \frac{64\gamma^3}{729}}\right)^{1/3} + \left(\frac{|v|}{2} - \sqrt{\frac{|v|^2}{4} + \frac{64\gamma^3}{729}}\right)^{1/3}$.
  $p = 5/4$: $t = \sqrt{\frac{|v|}{2\sqrt{2t_0}} - \frac{t_0}{2}} - \sqrt{\frac{t_0}{2}}, t_0 = \left(\frac{25\gamma^2}{128} + \sqrt{\frac{625\gamma^4}{65536} + \frac{|v|^2}{27}}\right)^{1/3} + \left(\frac{25\gamma^2}{128} - \sqrt{\frac{625\gamma^4}{65536} + \frac{|v|^2}{27}}\right)^{1/3}$.

  It is also noted that the limiting Lasso case, i.e. $p = 1$, is not governed by (20), but also has analytical solution $z = \text{sign}(v)\max(|v| - \gamma, 0)$, i.e. soft-thresholding.

- *Continuity:* It is easy to verify that for $|v_i| > 0$ $\lim_{\varepsilon \to 0, \delta \to 0} \Omega(|v_i| + \delta, p, \gamma) = \Omega(|v_i|, p, \gamma)$. The continuity property implies that for a fix p, if δ is sufficiently small and we know $\Omega(|v_i|, p, \gamma)$ then $\Omega(|v_i|, p, \gamma)$ should also be close to $\Omega(|v_i|, p; \gamma)$. This implies that an effective warm-up strategy can be exploited by using $\Omega(|v_i|, p; \gamma; \lambda)$ as a starting point for finding $\Omega(|v_i| + \delta, p; \gamma)$. Indeed, if δ is sufficiently small, one may use linear approximation

to the equilibrium equation and show that

$$\begin{aligned}\Omega(|v_i|+\delta,p;\gamma) &\approx \Omega(|v_i|,p;\gamma) \\ &\quad + \left(1+\frac{\lambda p(p-1)}{\rho\Omega(|v_i|,p;\gamma)^{p-2}}\right)^{-1}\delta.\end{aligned}$$

Similarly, for a given $|v_i|$, $\Omega(|v_i|,p;\gamma) \approx \Omega(|v_i|,p_0;\gamma)$ where p_0 is the nearest value of p that has an analytical solution. Furthermore, it is easily shown by taking the derivative of $h(p) = z + \gamma p z^{p-1}$ for $z > 0$, we can show that for $p_l \leq p \leq p_u$ it holds

$$\Omega(|v_i|,p_l;\gamma) \leq \Omega(|v_i|,p;\gamma) \leq \Omega(|v_i|,p_u;\gamma) \tag{22}$$

for $z > e^{-1/p}$ and the reverse inequalities if $z < e^{-1/p}$.

The above discussion suggests that if the user can select p to be one of the above special values in the range $(1,2]$ then the core problem in both ADMM and FISTA can be solved analytically, and thus group bridge regression can be solved as fast as Lasso-type problem. The special values of p in the range cover a wide range of sparse/dense models and may be sufficient for most cases. In the discussion to follow subsequently, we further outline other special values of p close to 1 that can be obtained almost nearly analytically. It is only when the value of p is not one of the special values, the core problem has to be solved numerically. Note that we need to solve a number of entries in the form of (16). Denote as $\mathbf{v}^k = \mathbf{x}^{k+1} + \mathbf{u}^k$. The following effective warm-up strategy is suggested when solving (16) for all entries.:

- Step 1: Sort the entries $\mathbf{v}^k$ in the increasing order of *magnitude*, ignoring the sign.
- Step 2: Identify the best upper p_u and lower p_l bounds on p that are known to have analytical solutions (the known values are 1, 5/4, 4/3, 3/2, 2).
- Step 3: Compute $\Omega(|v_i|,p;\gamma)$ for the smallest $|v_i|$. Note that $\Omega(0,p;\gamma) = 0$
- Step 4: Suppose that $|v_{i+1}|$ is the next entry in the sorted list. If $|v_{i+1}| = |v_i|$ then $\Omega(|v_{i+1}|,p;\gamma) = \Omega(|v_i|,p;\gamma)$. Otherwise, we check if $\delta = |v_{i+1}| - |v_i|$ is sufficiently larger than a designed threshold Δ. If so, we search over the range $[\Omega(|v_{i+1}|,p_l;\gamma), \Omega(|v_{i+1}|,p_u;\gamma)]$. Otherwise, we can improve the

lower bound of the ranger further with

$$\Omega(|v_i|,p;\gamma)+\left(1+\frac{\lambda p(p-1)}{\rho\Omega(|v_i|,p;\gamma)^{p-2}}\right)^{-1}\delta.$$

- Step 5: Repeat Step 4 until all entries are computed.
- Step 6: Adjust the sign of the solution if necessary.

3.4. Discussion

- There are certainly cases where group selection is a clear-cut choice, in which case sparse group selection like group Lasso might be better used. Thus, group bridge regression is useful when there are no clear subsets of dominant groups that influence predictivity.
- There are other flexible regularization alternatives to bridge regression, one of which is Elastic-net [26]. From a computational viewpoint, we note that Elastic-net can be posed as a special Lasso problem, and hence ADMM and FISTA algorithms for solving bridge regression is readily applied.
- We have shown that analytical solution is available for polynomial up to order 4 at $p=5/4$. In principle, it is possible to extend this, at least approximately to higher-order polynomials, i.e. smaller values of p. Considering equation (20) with $p=1+1/k$. We note that it is always possible to convert it to a form

$$s^k+Cs=b \tag{23}$$

for some fixed coefficient $C>0$ and $b=\frac{C|v|}{(1+1/p)\gamma}$ via the transformation $t=\left(\frac{(1+1/k)\gamma}{C}\right)^{\frac{1}{k-1}}s$. Thus, it is possible to study the polynomial s^k+Cs for given k,C so that the non-negative root of (23) may be pre-computed for a given accuracy. This is particular useful when p is close to one, i.e. when k is very large. For example, we can set $C=1$ and consider (23) for a very large value of k, i.e. $s^k+s=b$. We observe that this function is dominated by s when $s\in[0,1)$ and by s^k if $s>1$. The transition region at $s=1$ can be approximated smoothly by a second-order polynomial for example.
- Under the ADMM framework, it also appears that extensions to include other popular constraints, such as non-negative or affine, are possible. These

constraints may improve further the performance of machine learning methods based on ℓ_p regularization.

- Finally, we note that numerical algorithms for group bridge regression are not discussed in sufficient depth in the literature. The original work [18] does not outline such an algorithm but only concentrates on asymptotic analysis. The only exception is [27] where a discussion of the non-group is presented but for the case where $p < 2$ as the interest is on variable selection.

4. Experiments

This section consists of two parts: first we study the numerical properties of the proposed algorithms, then we illustrate the usefulness of ℓ_p regularization on some learning problems with the proposed algorithms. For completeness, we also compare group bridge regression with its non-group counter part, i.e. bridge regression, which is a very special case when the group size is actually 1. Two other flexible regularization non-group methods are also considered for comparison, including Elastic-net [26] and ℓ_p-norm multiple kernel learning (MKL) [16]. For ℓ_p-MKL, we select the linear, RBF, 2nd- and 3rd-order polynomials, and tanh kernels as the kernel set.

For $2 \geq p > 1$, the objective function of both bridge and group bridge regression is strictly convex. Hence, unconstrained convex optimization methods can be used. For example, [28] suggested the classic Newton-Raphson method bridge regression. However, as the second-derivative of the regularization term does not exist for when $p < 2$, this will run into numerical problems, especially when the solution is sparse, i.e. p close to 1. Another method to iteratively solve a number of Lasso sub-problem is proposed in [18] but only for the case $p < 1$. Thus, for large-scale learning problem, gradient methods with backtracking line-search [29] is often considered, for example [16]. For group bridge regression, the group-wise derivative is

$$\boldsymbol{\nabla} = (\mathbf{A}_g^T\mathbf{A} + \lambda\, p\|\mathbf{x}\|_2^{p-1}\mathbf{I})\mathbf{x}_g - \mathbf{A}_g^T\mathbf{y},$$

where $|\mathbf{x}|^{p-1}$ denotes a vector with the ith entry being $|x_i|^{p-1}$ and $\mathbf{A}_g$ is the submatrix of $\mathbf{A}$ that correspond to the group variable $\mathbf{x}_g$.

All algorithms are implemented in Matlab, and roughly optimized. The experiments are carried out on a Duo-core 3.3GHz 32-bit desktop computer.

4.1. Numerical properties

We generate synthetic data from the compressed sensing model $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a Gaussian random matrix with variance $1/m$. We set $n = 256, m = 64$ and divide $\mathbf{x}$ into $G = 16$ groups, each of dimension $n_g = 16$. In the first experiment, we set the coefficients of group 2 all to 1, and other to zero. A small Gaussian noise with standard deviation of 0.05 is injected to $\mathbf{y}$. We then examine the convergence behavior of different ℓ_p algorithms for $p = 5/4$ and we set $\lambda = 0.1\lambda_{\max}$ where $\lambda_{\max} = \|\mathbf{A}^T\mathbf{y}\|_\infty$.

To best compare different optimization algorithms, we study two aspects. One is the reduction of the error against the iterations $\|\mathbf{x}^k - \mathbf{x}_{\text{true}}\|_2$. Here, $\mathbf{x}_{\text{true}}$ is the true value of $\mathbf{x}$ and k is the iteration number. Note that $\mathbf{x}_{\text{true}}$ is not necessarily the optimizer of the corresponding problem, as different formulations have different optimizers. Measuring the error with reference to the true value not only reveal convergence properties, but also indicates which formulation give better results. The other related aspect is the actual computational time taken to reach a certain accuracy. Whilst this is intuitively closely related to the former, different algorithms have different complexities at each iteration. Hence, the number of iterations is not the actual indication of how fast an algorithm is.

We study the numerical properties of the FISTA, ADMM, and gradient algorithms for both bridge and group bridge formulations in two cases: $p = 5/4$ where the FISTA and ADMM algorithms have analytical updates, and $p = 1.2$ where the FISTA and ADMM algorithms need to solve the update step numerically. Results for reduction of the error versus iterations are shown in Figures 1 and 3, whilst results for the computational time taken to reach certain accuracy are shown in Figures 2 and 4. We make the following observations:

- Under the group bridge formulation, the ADMM and FISTA produce the best convergence and reach good final error. On the contrary, it appears that the gradient algorithm converges to some other value which has much larger error.

- In both group and non-group formulations, the ADMM algorithm tends to require less iterations and time to reach a specific accuracy than the FISTA algorithm, though both of them should reach the same value eventually.

- Compared with the exact case, both ADMM and FISTA algorithms take approximately the same number of iterations to reach a similar accuracy, but are about ten times slower due to the need to solve the update step numerically. The tolerance for the numerical update is set presently at 10^{-4}.

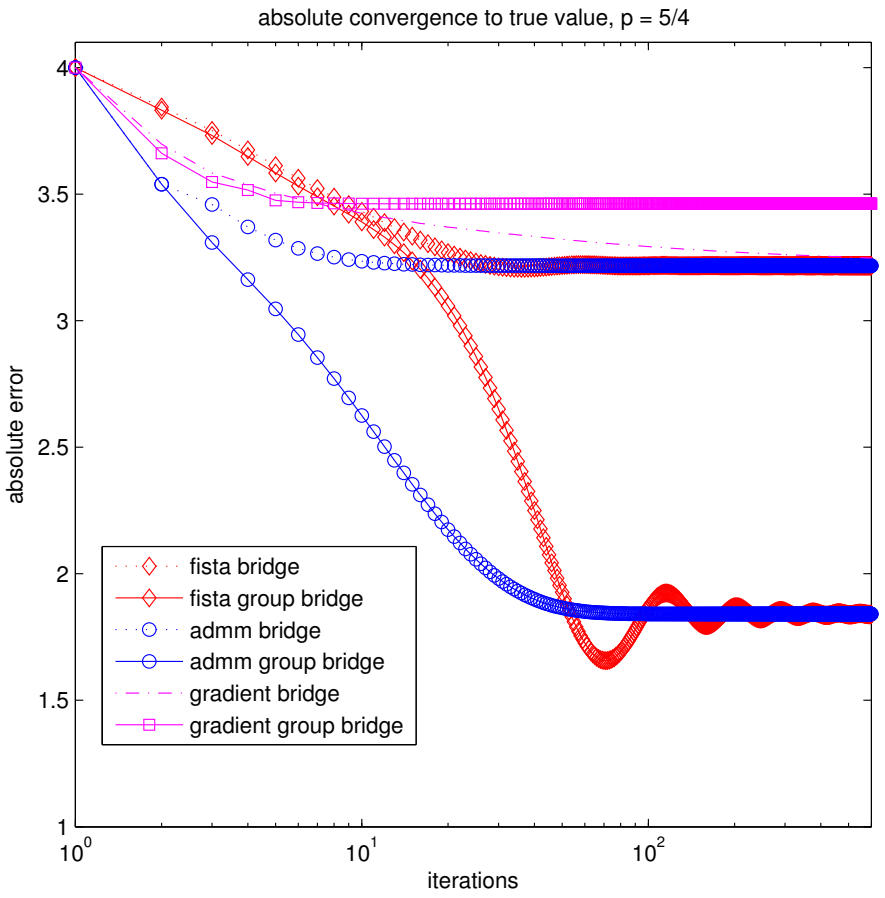

Figure 1: Absolute convergence in an exact case $p = 5/4$

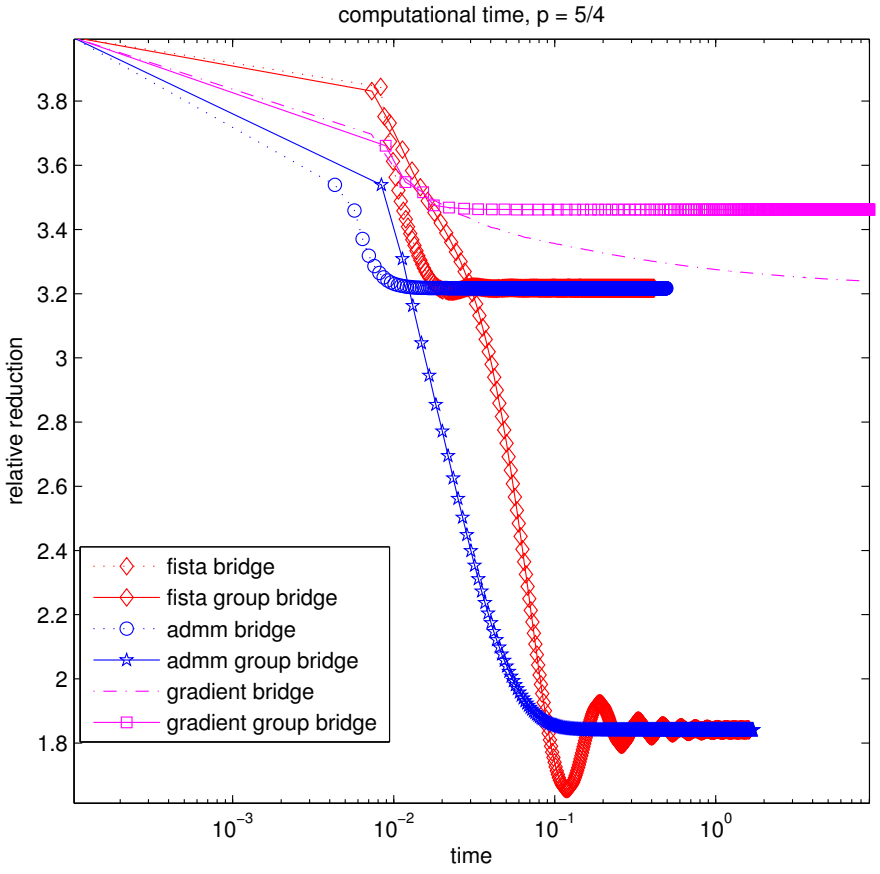

Figure 2: Computational time in an exact case $p = 5/4$

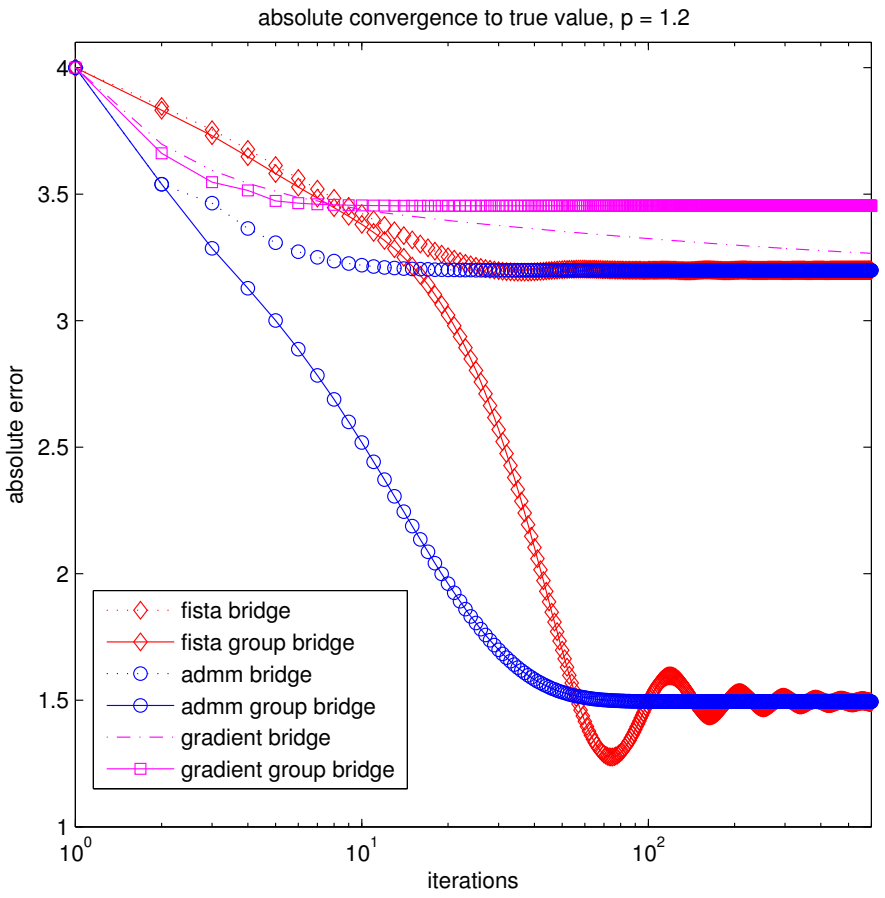

Figure 3: Absolute convergence in an inexact case $p = 1.2$

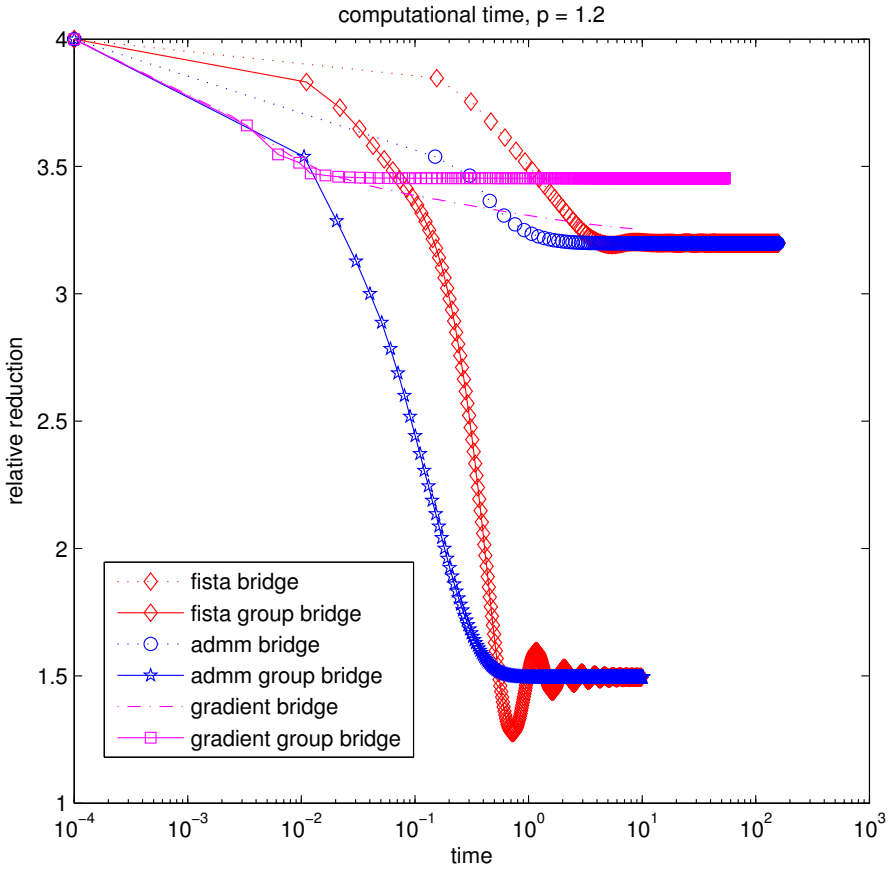

Figure 4: Computational time in an inexact case $p = 1.2$

Optimizing this might lead to reduced time, but it would be a matter of preference in each specific application.

- Clearly, the bridge formulation is inferior to the group bridge formulation as it does not exploit the information about the structure of the data. Consequently, this leads to larger errors, regardless of which algorithm used to solve it.

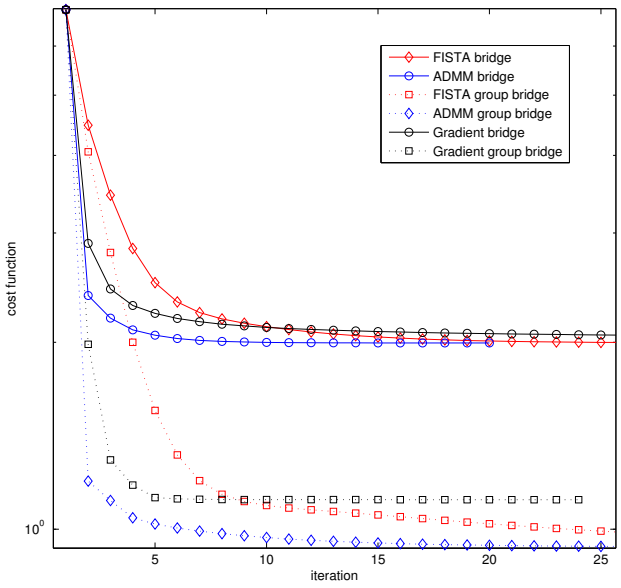

Figure 5: Convergence example of different algorithms

Such a result is shown in Fig. 5, which is a typical pattern that we observe over a wide range of settings. It is observed that overall ADMM achieves the best performance in terms of both convergence rate and accuracy. We found that FISTA converges more slowly but eventually reaches the accuracy of ADMM for a reasonable number of iterations. Gradient method is shown to have inferior accuracy to the other FISA and ADMM algorithms. We also note that the group bridge formulation yields more accurate result than bridge formulation, which is as expected due to the group constraint.

With the same settings, we also measure the average running time of the compared algorithms with $p = 5/4$ (analytical update) and $p = 1.2$ (numerical update)

Table 1: Running time comparison

	Bridge regression			Group bridge regression		
p	ADMM	FISTA	Gradient	ADMM	FISTA	Gradient
5/4	0.58 ± 0.04	0.53 ± 0.05	0.52 ± 0.06	0.14 ± 0.02	0.11 ± 0.03	0.07 ± 0.05
1.2	8.40 ± 0.23	4.74 ± 0.70	0.55 ± 0.05	0.54 ± 0.02	0.57 ± 0.11	0.08 ± 0.05

over 10 random repetitions. All algorithms are implemented in Matlab and initialized with a zero vector and the same convergence criterion of 0.001, i.e. they are terminated when $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 \leq 0.001\|\mathbf{x}^k\|_2$. The result is shown in Table. 1. As expected, gradient algorithms are relatively independent of the bridge order, whilst for both FISTA and ADMM algorithms, the analytical case is much faster than the numerical case. It is of interest to note that both ADMM and FISTA are competitive against the gradient alternative for the analytical case. Finally, the group formulation yields much better computational advantage as the effective problem size is reduced.

We note that whilst the gradient is fast (when relative change is used as a convergence criterion), we observe that its accuracy is always a problem, especially when p is close to 1. In many learning problems, it often fails to generate consistent performance when p varies. As a result, the ADMM and FISTA implementations are recommended. In what follows, we select the ADMM implementation for its accuracy.

4.2. Sparse-dense flexibility

Next, we demonstrate the effectiveness of ℓ_p regularization for compressible data. To do so, we revisit the above example and consider a decaying pattern of the form $x_i = \exp\{-\eta i\}$ (see Fig. 6). Such a pattern may be a more faithful description of real-world data [22]. We select the ADMM algorithm as the representative implementation for ℓ_p regularization and vary p between 1 and 2. For each value of p, we select the regularization parameter λ such that it yields the minimum error $\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$. The result is shown in Fig. 6. We observe that the selected values of $p = 4/3$ gives the most accurate recovery result. In this particular case, it shows that ℓ_p regularization is indeed effective in addressing the underlying characteristics of the data.

4.3. Regression and prediction

The above experiments are conducted in the compressed sensing flavor. Now, we consider experiments in a true machine learning perspective, i.e. attention is

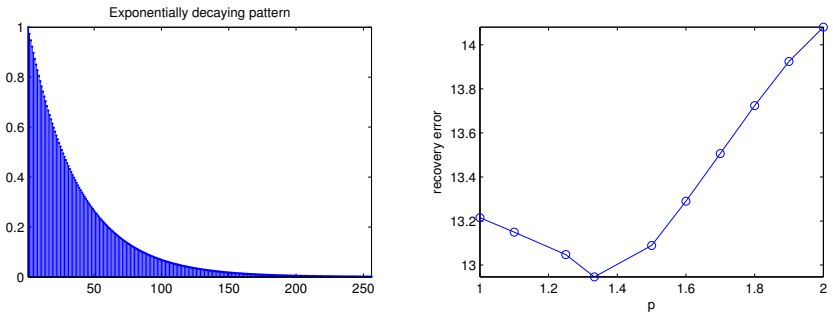

Figure 6: ℓ_p recovery error for an exponentially-decaying pattern

now paid to the predictive performance rather than recovery aspects. Like previous works on group Lasso [30, 31, 11], the Pearson score on the test set is used to compare different methods. The regularization parameter λ is optimized by selecting solution corresponding to the best Pearson score of the validation set over the search grid $\lambda \in [0.001\ 0.002\ 0.005\ 0.01\ 0.02\ 0.05\ 0.1\ 0.2\ 0.5]\lambda_{\max}$ where $\lambda_{\max} = \max_{g\in\mathscr{G}} \|\mathbf{X}_g^T\mathbf{y}\|$ [30]. The column of the design matrix is normalized to unit norm.

4.3.1. Synthetic learning problem

First, we consider a synthetic learning problem mentioned in [31] where we can control the group sparseness. The true vector $\mathbf{x}$ has 100 dimensions and is divided into 10 blocks of ten. The variables the first N_g blocks have random weights ± 1, whilst the rest is zero. The data points are generated by $\mathbf{a}_i = \mathbf{L}\mathbf{v}_i$ where $\mathbf{v}_i$'s are sampled from the distribution $\mathscr{N}(\mathbf{0},\mathbf{I})$, and $\mathbf{L}$ the Cholesky decomposition of a correlation matrix $\mathbf{\Sigma}$ with the i,jth entry being $0.2^{|i-j|}$. Then each $\mathbf{a}_i$ is normalized to unit norm. Then we generate $y_i = \text{sign}(\mathbf{a}^T\mathbf{x} + \varepsilon)$ where ε is Gaussian noise with variance of 0.1.

Three sets, training, validation, and test, are generated with equal size $N = 1000$. We vary the number of active groups N_g from 1 to 10, and collect the average Pearson score. For ℓ_p regularization, we consider both bridge and group bridge settings. The values for p are selected to cover the range $[1,2]$. Note that when $p = 1$ it is the Lasso/group Lasso, and when $p = 2$ both the group and non-group setting is the same (i.e. ridge regression).

The best Pearson scores and standard deviations for all methods are tabulated in Table 2. Clearly, bridge regression appears to be the best method for this problem and consistently outperforms other compared methods. Between bridge and

Table 2: Results on synthetic data.

N_g	GROUP BRIDGE	BRIDGE	ELASTIC-NET	ℓ_p MKL
1	$\mathbf{0.495 \pm 0.021}$	0.476 ± 0.021	0.474 ± 0.021	0.375 ± 0.020
2	$\mathbf{0.580 \pm 0.021}$	0.567 ± 0.021	0.563 ± 0.021	0.471 ± 0.033
3	$\mathbf{0.645 \pm 0.026}$	0.630 ± 0.026	0.625 ± 0.023	0.520 ± 0.015
4	$\mathbf{0.673 \pm 0.025}$	0.648 ± 0.025	0.651 ± 0.025	0.554 ± 0.032
5	$\mathbf{0.686 \pm 0.014}$	0.676 ± 0.014	0.668 ± 0.019	0.566 ± 0.022
6	$\mathbf{0.708 \pm 0.024}$	0.693 ± 0.024	0.687 ± 0.014	0.600 ± 0.024
7	$\mathbf{0.718 \pm 0.023}$	0.701 ± 0.023	0.701 ± 0.022	0.591 ± 0.029
8	$\mathbf{0.735 \pm 0.023}$	0.723 ± 0.023	0.717 ± 0.027	0.600 ± 0.022
9	$\mathbf{0.726 \pm 0.021}$	0.717 ± 0.021	0.717 ± 0.023	0.615 ± 0.026
10	$\mathbf{0.741 \pm 0.030}$	0.741 ± 0.030	0.730 ± 0.036	0.623 ± 0.033

group bridge, we can see a minor improvement when group information is exploited. Elastic-net is quite close to bridge regression, which hints that the flexibility in both methods probably have the similar effect. The only method that performs rather inferior in this case is ℓ_p-MKL.

Table 2 only shows the best scores for ℓ_p regularization methods. To illustrate of the dependence of the Pearson score on the actual p, we show the error bar plot of the Pearson score's variation against p for group bridge, bridge, and ℓ_p-MKL in Fig. 7 for two cases: 1 (sparse) and 7 (dense) active groups. We observe that the top scores for both bridge and group bridge are attained within the range $[1, 4/3]$, and reduced when $p \to 2$. We also see a clear gap between bridge and group bridge in both cases. For ℓ_p-MKL, we observe that it tends to favor dense solution regardless of the active group number.

4.3.2. Real-world learning problem

The splice detection on the MEMset data-set [2] has been considered as a real-world application of group Lasso in previous works [30, 31, 11]. This is one important genomic problem in computational biology.

A gene is a very long sequence which contains exons (coding) and introns (non-coding) segments. Exons are those sequence which remain after RNA splicing which removes introns. Splice sites are the regions between exons and in-

[2] `genes.mit.edu/burgelab/maxent/ssdata`

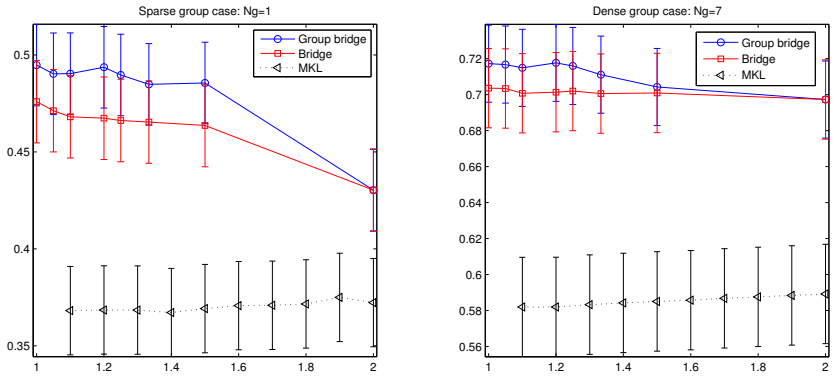

Figure 7: Performance on synthetic problem

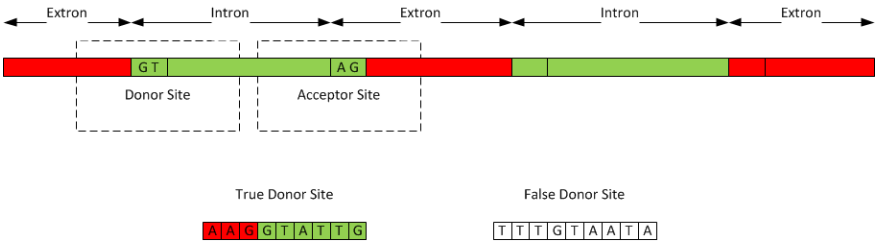

Figure 8: Examples of true and false donor sites in the splice detection problem

trons. There are two types of splice sites: donor and acceptor. We only consider the donor site detection problem. A donor site typically begins with the canonical 'GT' letters. The problem is to distinguish between the real and false splice sites, so this is essentially a binary classification problem. In a real splice donor site, the first subsequence corresponds to the last three bases of the preceeding exon, whilst the last subsequence corresponds to the six bases of the following intron, separated from the former by the consensus 'GT'. False splice donor sites are subsequences of the DNA which also have the 4th and 5th positions as the consensus 'GT'. It is typically hypothesized that there is a hidden structure in the DNA sequence that distinguishes between real and false splice sites.

The original training and testing sets are quite imbalanced. We follow the methodology in previous work and randomly divide the data to obtain a balanced training with 5,610 true and 5,610 false donor sites, and an unbalanced validation set with 2,805 true and 59,804 false donor sites, which has the same ratio of true/false as the original test set. After the pre-processing step that removes the consensus "GT", each data instance is a sequence of length 7 with 4 levels $\{$A, C, G, T$\}$. To extract the feature for each data instance, we consider modeling the co-occurance of level at different positions up to the second-order, which generates a binary feature vector of dimension 2604. Then we divide the feature vector into 63 groups, each corresponding to interactions for the same set of locations regardless of the levels. There are: 7 zero-order groups of size 4, 21 first-order groups of size 16, and 35 second-order groups of size 64. For each data instance, a feature vector is sparse and has only has 63 active entries (1), whilst the rest is inactive (0).

The best scores for all methods are shown in the left subplot of Fig. 9. The best method is group bridge (**0.6633** at $p = 1.2$), followed by bridge (0.6609 at $p = 4/3$), ℓ_p-MKL (0.6471 at $p = 1.8$), and elastic-net (0.5562). To the best of our knowledge, the score for group bridge regression here is the best result ever reported for this splice detection problem setting. Further detailed results on the dependence of the score on p are shown in the right subplot of Fig. 9. Here, we see again two characteristics: one is the advantage of the group formulation, and the other is the optimal range of p for bridge and group bridge regression.

5. Conclusion

As learning with ℓ_p regularization is a more flexible alternative and algorithmically stable than the ℓ_1 counter part, we have developed efficient algorithms for solving the associated challenging problem. Under the powerful ADMM and

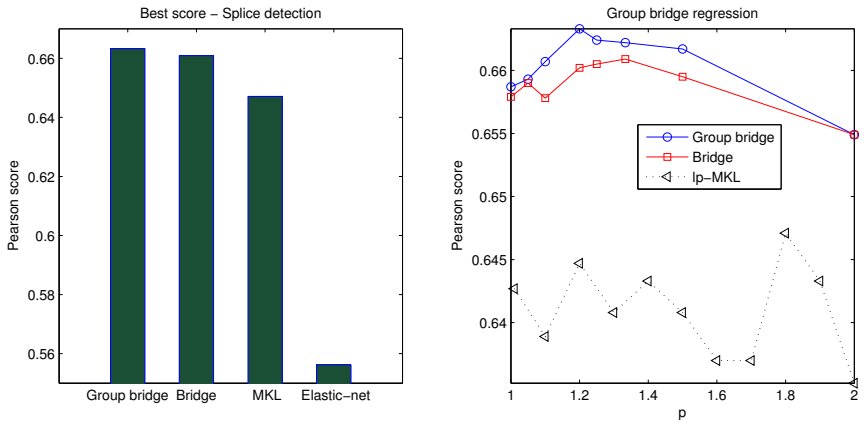

Figure 9: Performance on splice detection problem

FISTA frameworks, we have demonstrated that there is a common sub-problem that is the key for ℓ_p regularized optimization. Interestingly, we found that like Lasso, ℓ_p regularization can have analytical updates for a number of values of p over the range [1,2], and thus the developed algorithms provide both flexibility in the sparsity-stability trade-off and computational efficiency. For any other value of p, the warm-start strategy is proposed to solve the common sub-problem numerically and efficiently. We have demonstrated the numerical properties of the proposed algorithms, consistent with optimization theory. The applications of the proposed algorithm on several machine learning problems have revealed their potential to achieve state-of-the-art performance against other alternatives.

Appendix A. Proof of Theorem 3

We prove Theorem 3 by extending the proof technique in [24] to the group bridge regression case with. We note the following:

- While the sufficient condition would be met by a strongly convex regularization, it is not necessary to be so. The sufficient condition only requires the inequality to hold at the minimizers, which mean that strong convexity on bounded domain is sufficient.

- For the class of norm regularization, it was shown that $P(\mathbf{x_z}) \leq (B/\lambda)$ where $B = \sup_y V(0,y)$ (see Lemma 3.1 in [24]). Consequently $\|\mathbf{x_z}\|_2$ is bounded by some $\tau = (B/\lambda)^{1/p}$ [24].

- The sufficient condition does not make any specific assumptions about the loss function as long as it satisfies the general conditions, so the focus is on the regularization term.

For notational simplicity, we prove that for two vectors $\mathbf{x}$ and $\mathbf{y}$ such that their magnitude are bounded by $\tau^{1/p}$ where $\tau = (B/\lambda)$, i.e. $\|\mathbf{x}\|_2, \|\mathbf{y}\|_2 \leq \tau^{1/p}$, and that $1 < p \leq 2$, then there exits a constant ρ such that

$$\|\mathbf{x}\|_2^p + \|\mathbf{y}\|_2^p - 2\|\frac{\mathbf{x}+\mathbf{y}}{2}\|_2^p \geq \rho\|\mathbf{x}-\mathbf{y}\|_2^2. \tag{A.1}$$

For group bridge regression, the constant is precisely

$$\rho = (1/4)p(p-1)\tau^{\frac{p-2}{p}} = (1/4)p(p-1)\,(B/\lambda)^{\frac{p-2}{p}}$$

.

For the case when x and y are scalar quantities, the proof in [24] uses the second-order mean value theorem, which essentially says that for a convex function f on the bounded domain with continuous derivative up to the second order, then there exists c, where $\min\{x,y\} \leq c \leq \max\{x,y\}$ such that

$$f(x) + f(y) - 2f(\frac{x+y}{2}) = \frac{1}{4}f''(c)$$

Here, f'' denotes the second-order derivative. In [24], the convex function $f(t) = |t|^p$ is used to derive the result and that $|x|^p + |y|^p - 2|\frac{x+y}{2}|^p = \frac{(y-x)^2}{4}p(p-1)|c|^{p-2}$.

We also use the same technique, but instead consider the function $f(t) = \|\mathbf{x} + t\mathbf{u}\|_2^p$ where $\mathbf{u} = \mathbf{y} = \mathbf{x}$. We note that

$$f(0) = \|\mathbf{x}\|_2^p, f(1) = \|\mathbf{y}\|_2^p, f(1/2) = \|\frac{\mathbf{y}+\mathbf{x}}{2}\|_2^p.$$

So according to the second-order mean value theorem, there exists some $c \in [0,1]$ such that

$$f(0) + f(1) - 2f(1/2) = \frac{1}{4}f''(c).$$

We now evaluate $f''(c)$. To do so, we explicitly write

$$f(t) = \left(\sum_i (x_i + tu_i)^2\right)^{p/2}.$$

It is straightforward to show that

$$\begin{aligned} f'(t) &= p\|\mathbf{x}+t\mathbf{u}\|_2^{p-2}\left((\mathbf{x}+t\mathbf{u})^T\mathbf{u}\right) \\ f''(t) &= p(p-2)\left((\mathbf{x}+t\mathbf{u})^T\mathbf{u}\right)\|\mathbf{x}+t\mathbf{u}\|_2^{p-4}\left((\mathbf{x}+t\mathbf{u})^T\mathbf{u}\right)+p\|\mathbf{x}+t\mathbf{u}\|_2^{p-2}\|\mathbf{u}\|_2^2. \end{aligned}$$

Denote as $\mathbf{z}=\mathbf{x}+c\mathbf{u}$, then

$$f'(c)=p\|\mathbf{z}\|_2^{p-2}\mathbf{z}^T\mathbf{u},\ f''(c)=p(p-2)\mathbf{z}^T\mathbf{u}\|\mathbf{z}\|_2^{p-4}\mathbf{z}^T\mathbf{u}+p\|\mathbf{z}\|_2^{p-2}\|\mathbf{u}\|_2^2.$$

Now let θ be the angle between $\mathbf{z}$ and $\mathbf{u}$ so that $\mathbf{z}^T\mathbf{u}=\|\mathbf{z}\|_2\|\mathbf{u}\|_2\cos(\theta)$, then we can simplify the second-order derivative as

$$f''(c)=\|\mathbf{u}\|_2^2\|\mathbf{z}\|_2^{p-2}(p(p-2)\cos^2(\theta)+p).$$

Next, we obtain a lower bound of $f''(c)$. As $\|\mathbf{x}\|_2$ and $\|\mathbf{y}\|_2\leq\tau^{1/p}$ and that $1<p\leq 2$, it follows that $p-2<0$ and hence

$$\|\mathbf{z}\|_2^{p-2}\geq\sup\max\{\|\mathbf{x}\|_2^{p-2},\|\mathbf{y}\|_2^{p-2}\}.$$

This yields $\|\mathbf{z}\|_2^{p-2}\geq\tau^{\frac{p-2}{p}}$. Meanwhile, note that $p(p-2)<0$ and $\cos^2(\theta)\leq 1$, hence

$$p(p-2)\cos^2(\theta)+p\geq p(p-2)+p=p(p-1)>0.$$

Noting that $\mathbf{u}=\mathbf{y}-\mathbf{x}$, then these results yield

$$f''(c)\geq p(p-1)\tau^{\frac{(p-2)}{p}}\|\mathbf{y}-\mathbf{x}\|_2^2,$$

and thus an application of the second-order mean value theorem immediately completes the proof.

References

[1] B. Schölkopf, A. Smola, Learning with kernels, The MIT Press, 1998.

[2] S. Negahban, M. Wainwright, Simultaneous support recovery in high dimensions: Benefits and perils of block ℓ_1/ℓ_∞-regularization, IEEE Trans. Info. Theory 57 (6) (2011) 3841–3863.

[3] D. Donoho, Compressed sensing, IEEE Trans. Info. Theory 52 (4) (2006) 1289–1306.

[4] E. Candès, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Info. Theory 52 (2) (2006) 489–509.

[5] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) (1996) 267–288.

[6] J. Yang, L. Zhang, Y. Xu, J.-y. Yang, Beyond sparsity: The role of l1-optimizer in pattern classification, Pattern Recognition 45 (3) (2012) 1104–1118.

[7] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE PAMI 31 (2) (2009) 210–227.

[8] L. Zhang, S. Chen, L. Qiao, Graph optimization for dimensionality reduction with sparsity constraints, Pattern Recognition 45 (3) (2012) 1205–1210.

[9] A. Rebai, A. Joly, N. Boujemaa, Blasso for object categorization and retrieval: Towards interpretable visual models, Pattern Recognition 45 (6) (2012) 2377–2389.

[10] J. Huang, T. Zhang, The benefit of group sparsity, The Annals of Statistics 38 (4) (2010) 1978–2004.

[11] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) (2006) 49–67.

[12] F. Bach, Consistency of the group lasso and multiple kernel learning, The Journal of Machine Learning Research 9 (2008) 1179–1225.

[13] Y. Yang, Z. Huang, Y. Yang, J. Liu, H. T. Shen, J. Luo, Local image tagging via graph regularized joint group sparsity, Pattern Recognition 46 (5) (2013) 1358–1368.

[14] Q. Shi, A. Eriksson, A. van den Hengel, C. Shen, Is face recognition really a compressive sensing problem?, in: Proc. CVPR, IEEE, 2011, pp. 553–560.

[15] H. Xu, C. Caramanis, S. Mannor, Sparse algorithms are not stable: A no-free-lunch theorem, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (1) (2012) 187–193.

[16] M. Kloft, U. Brefeld, S. Sonnenburg, A. Zien, Lp-norm multiple kernel learning, Journal of Machine Learning Research 12.

[17] J. E. Vogt, V. Roth, A Complete Analysis of the $\ell_{1,p}$ Group-Lasso, in: Proc. ICML, 2012.

[18] J. Huang, S. Ma, H. Xie, C. Zhang, A group bridge approach for variable selection, Biometrika 96 (2) (2009) 339–355.

[19] J. Huang, J. Horowitz, S. Ma, Asymptotic properties of bridge estimators in sparse high-dimensional regression models, The Annals of Statistics 36 (2) (2008) 587–613.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Foundations and Trends in Machine Learning, Vol. 3, Now Publisher, 2011, Ch. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, pp. 1–122.

[21] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (1) (2009) 183–202.

[22] J. Haupt, R. Nowak, Signal reconstruction from noisy random projections, IEEE Trans. Info. Theory 52 (9) (2006) 4036–4048.

[23] O. Bousquet, A. Elisseeff, Stability and generalization, The Journal of Machine Learning Research 2 (2002) 499–526.

[24] A. Wibisono, L. Rosasco, T. Poggio, Sufficient conditions for uniform stability of regularization algorithms, Tech. rep., MIT (2009).

[25] Y. Nesterov, Gradient methods for minimizing composite objective function, CORE, 2007.

[26] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67 (2) (2005) 301–320.

[27] C. Park, Y. J. Yoon, Bridge regression: adaptivity and group selection, Journal of Statistical Planning and Inference 141 (11) (2011) 3506–3519.

[28] W. Fu, Penalized regressions: the bridge versus the lasso, Journal of computational and graphical statistics (1998) 397–416.

[29] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge Univ Pr, 2004.

[30] L. Meier, S. Van De Geer, P. Bühlmann, The group lasso for logistic regression, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70 (1) (2008) 53–71.

[31] H. Yang, Z. Xu, I. King, M. Lyu, Online learning for group lasso, in: Proc. ICML, 2010.