A Comparative Framework for Preconditioned Lasso Algorithms

Authored by:

Nebojsa Jojic
Fabian L. Wauthier
Michael I. Jordan

Abstract

The Lasso is a cornerstone of modern multivariate data analysis, yet its performance suffers in the common situation in which covariates are correlated. This limitation has led to a growing number of emph{Preconditioned Lasso} algorithms that pre-multiply X and y by matrices P_X, P_y prior to running the standard Lasso. A direct comparison of these and similar Lasso-style algorithms to the original Lasso is difficult because the performance of all of these methods depends critically on an auxiliary penalty parameter $lambda$. In this paper we propose an agnostic, theoretical framework for comparing Preconditioned Lasso algorithms to the Lasso without having to choose $lambda$. We apply our framework to three Preconditioned Lasso instances and highlight when they will outperform the Lasso. Additionally, our theory offers insights into the fragilities of these algorithms to which we provide partial solutions.

1 Paper Body

Variable selection is a core inferential problem in a multitude of statistical analyses. Confronted with a large number of (potentially) predictive variables, the goal is to select a small subset of variables that can be used to construct a parsimonious model. Variable selection is especially relevant in linear observation models of the form y = X? ? + w

with w ? N (0, ? 2 In?n),

(1)

?

where X is an n ? p matrix of features or predictors, ? is an unknown p-dimensional regression parameter, and w is a noise vector. In high-dimensional settings where n p, ordinary least squares is generally inappropriate. Assuming that ? ? is sparse (i.e., the support set S(? ?) , {i—?i? 6= 0} has cardinality k ¡ n), a mainstay algorithm for such settings is the Lasso [10]: 1 2 Lasso: ?? = argmin??Rp ——y ? X?——2 + ? ——?——1 . 2n

(2)

For a particular choice of ?, the variable selection properties of the Lasso can be analyzed by quan? approximates the true support S(? ?). More careful tifying how well the estimated support S(?) analyses focus instead on recovering the signed support S? (? ?), (+1 if ?i? ¿ 0 ? ?1 if ?i? ¡ 0 . S? (?i) , (3) 0 o.w. Theoretical developments during the last decade have shed light onto the support recovery properties of the Lasso and highlighted practical difficulties when the columns of X are correlated. These developments have led to various conditions on X for support recovery, such as the mutual incoherence or the irrepresentable condition [1, 3, 8, 12, 13]. 1

In recent years, several modifications of the standard Lasso have been proposed to improve its support recovery properties [2, 7, 14, 15]. In this paper we focus on a class of ?Preconditioned Lasso? algorithms [5, 6, 9] that pre-multiply X and y by suitable matrices PX and Py to yield ? = PX X, y? = Py y, prior to running Lasso. Thus, the general strategy of these methods is X

1 ? = argmin ? ——?—— . ? 2 + ? Preconditioned Lasso: ?? y? ? X? (4) ??Rp 1 2 2n Although this class of algorithms often compares favorably to the Lasso in practice, our theoretical understanding of them is at present still fairly poor. Huang and Jojic [5], for example, consider only empirical evaluations, while both Jia and Rohe [6] and Paul et al. [9] consider asymptotic consistency under various assumptions. Important and necessary as they are, consistency results do not provide insight into the relative performance of Preconditioned Lasso variants for finite data sets. In this paper we provide a new theoretical basis for making such comparisons. Although the focus of the paper is on problems of the form of Eq. (4), we note that the core ideas can also be applied to algorithms that right-multiply X and/or y with some matrices (e.g., [4, 11]). For particular instances of X, ? ? , we want to discover whether a given Preconditioned Lasso algorithm following Eq. (4) improves or degrades signed support recovery relative to the standard Lasso of Eq. (2). A major roadblock to a one-to-one comparison are the auxiliary penalty param? which trade off the ‘1 penalty to the quadratic objective in both Eq. (2) and Eq. (4). eters, ?, ?, A correct choice of penalty parameter is essential for signed support recovery: If it is too small, the algorithm behaves like ordinary least squares; if it is too large, the estimated support may be empty. Unfortunately, in all but the simplest cases, pre-multiplying data X, y by matrices PX , Py changes the relative geometry of the ‘1 penalty contours to the elliptical objective contours in a nontrivial way. Suppose we wanted to compare the Lasso to the Preconditioned Lasso by choosing ? in Eq. (4). For a fair comparison, the resulting mapfor each ? in Eq. (2) a suitable, matching ? ping would have to capture the change of relative geometry induced by preconditioning of X, y, ? = f (?, X, y, PX , Py). It seems difficult to theoretically characterize such a mapping. Furi.e. ? thermore, it seems unlikely that a comparative framework could be built by independently choosing ? Meinshausen and B?uhlmann [8], for example, demonstrate that a ?ideal? penalty parameters ?, ?: seemingly reasonable oracle estimator of ? will not lead to consistent support recovery in the Lasso. In the Preconditioned Lasso literature this problem is commonly sidestepped

either by resorting to asymptotic comparisons [6, 9], empirically comparing regularization paths [5], or using modelselection techniques which aim to choose reasonably ?good? matching penalty parameters [6]. We deem these approaches to be unsatisfactory?asymptotic and empirical analyses provide limited insight, and model selection strategies add a layer of complexity that may lead to unfair comparisons. It is our view that all of these approaches place unnecessary emphasis on particular choices of penalty parameter. In this paper we propose an alternative strategy that instead compares the Lasso to the Preconditioned Lasso by comparing data-dependent upper and lower penalty parameter bounds. Specifically, we give bounds (?u , ?l) on ? so that the Lasso in Eq. (2) is guaranteed to recover the signed support iff ?l ¡ ? ¡ ?u . Consequently, if ?l ¿ ?u signed support recovery is not possible. ? = PX X, y? = Py y and will thus induce new The Preconditioned Lasso in Eq. (4) uses data X ? ? ? bounds (?u , ?l) on ?. The comparison of Lasso and Preconditioned Lasso on an instance X, ? ? ? The advantage of this approach is that then proceeds by suitably comparing the bounds on ? and ?. the upper and lower bounds are easy to compute, even though a general mapping between specific penalty parameters cannot be readily derived. To demonstrate the effectiveness of our framework, we use it to analyze three Preconditioned Lasso algorithms [5, 6, 9]. Using our framework we make several contributions: (1) We confirm intuitions about advantages and disadvantages of the algorithms proposed in [5, 9]; (2) We show that for an SVD-based construction of n ? p matrices X, the algorithm in [6] changes the bounds deterministically; (3) We show that in the context of our framework, this SVD-based construction can be thought of as a limit point of a Gaussian construction. The paper is organized as follows. In Section 2 we will discuss three recent instances of Eq. (4). We outline our comparative framework in Section 3 and highlight some immediate consequences for [5] and [9] on general matrices X in Section 4. More detailed comparisons can be made by considering a generative model for X. In Section 5 we introduce such a model based on a block-wise SVD of X and then analyze [6] for specific instances of this generative model. Finally, we show that in terms of signed support recovery, this generative model can be thought of as a limit point of a Gaussian 2

construction. Section 6 concludes with some final thoughts. The proofs of all lemmas and theorems are in the supplementary material.

2

Preconditioned Lasso Algorithms

Our interest lies in the class of Preconditioned Lasso algorithms that is summarized by Eq. (4). Extensions to related algorithms, such as [4, 11] will follow readily. In this section we focus on three recent Preconditioned Lasso examples and instantiate the matrices PX , Py appropriately. Detailed derivations can be found in the supplementary material. For later reference, we will denote each algorithm by the author initials. Huang and Jojic [5] (HJ). Huang and Jojic proposed Correlation Sifting [5], which, although not presented as a preconditioning algorithm, can be rewritten as one. Let the SVD of X be X = U DV ¿ . Given an algorithm parameter q, let UA be the set of q smallest left singular vectors of X 1 . Then HJ amounts to setting ¿ PX = Py = UA UA .

(5)

Paul et al. [9] (PBHT). An earlier instance of the preconditioning idea was put forward by Paul et al. [9]. For some algorithm parameter q, let A be the q column indices of X with largest absolute correlation to y, (i.e., where —Xj¿ y—/——Xj ——2 is largest). Define UA to be the q largest left singular vectors of XA . With this, PBHT can be expressed as setting ¿ Py = UA UA .

PX = In?n

(6)

Jia and Rohe [6] (JR). Jia and Rohe [6] propose a preconditioning method that amounts to whitening the matrix X. If X = U DV ¿ is full rank, then JR defines2 ?1/2 ¿ PX = Py = U DD¿ U . (7) ¿ ¿ ¿ ¿ ¿ ¿ ?X ? = PX XX P ? In?n and if n ¿ p then X ? X ? = X P PX X ? Ip?p . If n ¡ p then X X X Both HJ and PBHT estimate a basis UA for a q-dimensional subspace onto which they project y and/or X. However, since the methods differ substantially in their assumptions, the estimators differ also. Empirical results in [5] and [9] suggest that the respective assumptions are useful in a variety of situations. In contrast, JR reweights the column space directions U and requires no extra parameter q to be estimated.

3

Comparative Framework

In this section we propose a new comparative approach for Preconditioned Lasso algorithms which ? We first derive upper and lower bounds for ? avoids choosing particular penalty parameters ?, ?. ? ? satisfy the bounds. and ? respectively so that signed support recovery can be guaranteed iff ? and ? We then compare estimators by comparing the resulting bounds. 3.1

Conditions for signed support recovery

Before proceeding, we make some definitions motivated by Wainwright [12]. Suppose that the support set of ? ? is S , S(? ?), with —S— = k. To simplify notation, we will assume throughout that S = {1, . . . , k} so that the corresponding off-support set is S c = {1, . . . , p} S, with —S c — = p ? k. Denote by Xj column j of X and by XA the submatrix of X consisting of columns indexed by set A. Define the following variables: For all j ? S c and i ? S, let w ?j = Xj¿ XS (XS¿ XS)?1 sgn(?S?) ?j = Xj¿ In?n ? XS (XS¿ XS)?1 XS¿ (8) n

?1

?1 1 ¿ 1 ¿ w ?i = e¿ X XS sgn(?S?) i = e ¿ X XS XS¿ . (9) i i n S n S n 1

The choice of smallest singular vectors is considered for matrices X with sharply decaying spectrum. We note that Jia and Rohe [6] let D be square, so that it can be directly inverted. If X is not full rank, the pseudo-inverse of D can be used. 2

3

1 ? = S ?(? ?)) P (S ? (?)

? = S ?(? ?)) P (S ? (?)

1 0.8 0.6 0.4 0.2 0 0.5

1 f

0.8 0.6 0.4 0.2 0 0.5

1.5

(a) Signed support recovery around ?l .

1 f

1.5

(b) Signed support recovery around ?u .

Figure 1: Empirical evaluation of the penalty parameter bounds of Lemma 1. For each of 500 synthetic Lasso problems (n = 300, p = 1000, k = 10) we computed ?l , ?u as per Lemma 1. Then we ran Lasso using penalty parameters f ?l in Figure (a) and f ?u in Figure (b), where the factor f = 0.5, . . . , 1.5. The figures show the empirical probability of signed support recovery as a function of the factor f for both ?l and ?u . As expected, the probabilities change sharply at f = 1. For the traditional Lasso of Eq. (2), results in (for example) Wainwright [12] connect settings of ? with instances of X, ? ? , w to certify whether or not Lasso will recover the signed support. We invert these results and, for particular instances of X, ? ? , w, derive bounds on ? so that signed support recovery is guaranteed if and only if the bounds are satisfied. Specifically, we prove the following Lemma in the supplementary material. Lemma 1. Suppose that XS¿ XS is invertible, —?j — ¡ 1, ?j ? S c , and sgn(?i?)?i ¿ 0, ?i ? S. Then ? = S? (? ?)) if and the Lasso has a unique solution ?? which recovers the signed support (i.e., S? (?) only if ?l ¡ ? ¡ ?u , where ?

? i + i ?j

, ?l = maxc (10) ?u = min j?S (2J?j ¿ 0K ? 1) ? ?j i?S ?i +

J?K denotes the indicator function and — ? —+ = max(0, ?) denotes the hinge function. On the other hand, if XS¿ XS is not invertible, then the signed support cannot in general be recovered. Lemma 1 recapitulates well-worn intuitions about when the Lasso has difficulty recovering the signed support. For instance, assuming that w has symmetric distribution with mean 0, if 1 ? —?j — is small (i.e., the irrepresentable condition almost fails to hold), then ?l will tend to be large. In extreme cases we might have ?l ¿ ?u so that signed support recovery is impossible. Figure 1 empirically validates the bounds of Lemma 1 by estimating probabilities of signed support recovery for a range of penalty parameters on synthetic Lasso problems. 3.2

Comparisons

In this paper we propose to compare a preconditioning algorithm to the traditional Lasso by comparing the penalty parameter bounds produced by Lemma 1. As highlighted in Eq. 4, the precondition? = PX X, y? = Py y. For the purpose of applying ing framework runs Lasso on modified variables X Lemma 1, these transformations induce a new noise vector ? ? = Py (X? ? + w) ? PX X? ? . w ? = y? ? X? (11) ? ? ? we can Note that if PX = Py then w ? = Py w. Provided the conditions of Lemma 1 hold for X, ? ? ? can define updated variables ? ?j , ??i , ??j , ?i from which the bounds ?u , ?l on the penalty parameter ? be derived. In order for our comparison to be scale-invariant, we will compare algorithms by ratios of resulting penalty parameter bounds. That is, we deem a Preconditioned Lasso algorithm to be ? u /? ? l ¿ ?u /?l . Intuitively, the upper bound ? ? u is then more effective than the traditional Lasso if ? ? ?3. disproportionately larger than ?l relative to ?u and ?l , which in principle

allows easier tuning of ? ? ? ? ? We will later encounter the special case ?u 6= 0, ?l = 0 in which case we define ?u /?l , ? to ? u /? ? l ¡ 1 then signed support recovery is indicate that the preconditioned problem is very easy. If ? ? u /? ? l , 0 if ? ?u = ? ? l = 0. in general impossible. Finally, to match this intuition, we define ? 3 ?l , ? ? u could also be considered. However, we find the ratio to be a particuOther functions of ?l , ?u and ? larly intuitive measure.

4

4

General Comparisons

We begin our comparisons with some immediate consequences of Lemma 1 for HJ and PBHT. In order to highlight the utility of the proposed framework, we focus in this section on special cases of PX , Py . The framework can of course also be applied to general matrices PX , Py . As we will see, both HJ and PBHT have the potential to improve signed support recovery relative to the traditional Lasso, provided the matrices PX , Py are suitably estimated. The following notation will be used during our comparisons: We will write A? A to indicate that random variable A stochastically ? that is, ?t P(A? ? t) ? P(A ? t). We also let US be a minimal basis for the column dominates A,

space of the submatrix XS , and define span(US) = x ?c ? Rk s.t. x = US c ? Rn . Finally, we let US c be a minimal basis for the orthogonal complement of span(US). ¿ Consequences for HJ. Recall from Section 2 that HJ uses PX = Py = UA UA , where UA is a column basis estimated from X. We have the following theorem: Theorem 1. Suppose that the conditions of Lemma 1 are met for a fixed instance of X, ? ? . If span(US) ? span(UA), then after preconditioning using HJ the conditions continue to hold, and ?u ?u ? ? , (12) ?l ?l

where the stochasticity on both sides is due to independent noise vectors w. On the other hand, if ¿ PX XS is not invertible, then HJ cannot in general recover the signed support. XS¿ PX We briefly sketch the proof of Theorem 1. If span(US) ? span(UA) then plugging in the definition of PX into ? ?j , ??i , ??j , ?i , one can derive the following ? ?j = ?j ??i = ?i (13)

¿ ¿ ¿w ??j = Xj In?n ? US US UA UA ?i = i . (14) n If span(UA) = span(US), then it is easy to see that ??j = 0. Notice that because ? ?j and ??i are unchanged, if the conditions of Lemma 1 hold for the original Lasso problem (i.e., XS¿ XS is invertible, —?j — ¡ 1 ?j ? S c and sgn(?i?)?i ¿ 0 ?i ? S), they will continue to hold for the preconditioned problem. Suppose then that the conditions set forth in Lemma 1 are met. With some additional work one can show that

?i? + ?i ??j ? ? l = max

= ?u ? ?l . (15) ?u = min

c i?S j?S (2J? ??i ?j ¿ 0K ? 1) ? ? ?j + ? l , ?l are both independent of ? ? u = ?u . Note that if The result then follows by showing that ? ? ? ? span(UA) = span(US), then ?l = 0 and so ?u /?l , ?. In the more common case when span(US) 6? span(UA) the performance of the Lasso depends on how misaligned UA and US are. In ¿ extreme cases, XS¿ PX PX XS is singular and so signed support recovery is not in general possible. ¿ Consequences for PBHT.

Recall from Section 2 that PBHT uses PX = In?n , Py = UA UA , where UA is a column basis estimated from X. We have the following theorem. Theorem 2. Suppose that the conditions of Lemma 1 are met for a fixed instance of X, ? ? . If span(US) ? span(UA), after preconditioning using PBHT the conditions continue to hold, and ?u ?u ? (16) ? , ?l ?l

where the stochasticity on both sides is due to independent noise vectors w. On the other hand, if span(US c) = span(UA), then PBHT cannot recover the signed support. As before, we sketch the proof to build some intuition. Because PBHT does not set PX = Py as HJ ¿ does, there is no danger of XS¿ PX PX XS becoming singular. On the other hand, this complicates ¿ the form of the induced noise vector w. ? Plugging PX and Py into Eq. (11), we find w ? = (UA UA ? ? ¿ In?n)X? + UA UA w. However, even though the noise has a more complicated form, derivations in the supplementary material show that if span(US) ? span(UA), then ? ?j = ?j ??i = ?i (17)

¿ ¿ ¿w ??j = Xj In?n ? US US UA UA ?i = i . (18) n 5

2.5

0.6 0.4 0.2 0

Orthogonal data Gaussian data

2 Lasso 55 35 25 15 10 0

1000

2000

3000

4000

? u/ ? ?l ? ? u/ ? l

P (? u /? l ¡ t)

1 0.8

1.5 1 0.5 0 0.2

5000

t

0.4

0.6

0.8

1

1.2

1.4

f

(a) Empirical validation of Theorems 1 and 2.

(b) Evaluation of JR on Gaussian ensembles.

Figure 2: Experimental evaluations. Figure (a) shows empirical c.d.f.?s of penalty parameter bounds ratios estimated from 1000 variable selection problems. Each problem consists of Gaussians X and w, and ? ? , with n = 100, p = 300, k = 5. The blue curve shows the c.d.f. for ?u /?l estimated on ¿ the original data (Lasso). Then we projected the data using PX = Py = UA UA , where span(US) ? span(UA) but dim(UA) = dim(span(UA)) is variable (see legend), and estimated the resulting ? u /? ? l . As predicted by Theorems 1 and 2, ?u /?l ? ? u /? ? l . In c.d.f. for the updated bounds ratio ? 2 Figure

(b) the blue curve shows the scale factorp (p ? k)/(n + p? ? k) predicted by Theorem 3 for problems constructed from Eq. (19) for ? = f 1 ? (n/p). The red curve plots the corresponding factor estimated from the Gaussian construction in Eq. (25) (n = 100, m = 2000, p = 200, k = 5) using the same ?S , ?S c as in Theorem 3, averaged over 50 problem instances and with error bars for one standard deviation. As in Theorem 3, the factor is approximately 1 if f = 1.

As with HJ, if span(UA) = span(US), then ??j = 0. Because ? ?j and ??i are again unchanged, the conditions of Lemma 1 will continue to hold for the preconditioned problem if they hold for the original Lasso problem. With the previous equalities established, the remainder of the proof is identical to that of Theorem 1. The fact that the above ? ?j , ??j , ??i , ?i are identical to those of HJ depends crucially on the fact that span(US) ? span(UA). In general the values will differ because PBHT sets PX = In?n , but HJ does not. On the other hand, if span(US) 6? span(UA) then the distribution of ?i depends on how misaligned UA and US are. In the extreme case when span(US c) = span(UA), one can show that ?i = ??i? , ? u = 0, ? ? l ?l . Because P(? ? l ? 0) = 1, signed support recovery is not possible. which results in ? Remarks. Our theoretical analyses show that both HJ and PBHT can indeed lead to improved signed support recovery relative to the Lasso on finite datasets. To underline our findings, we empirically validate Theorems 1 and 2 in Figure 2(a), where we plot estimated c.d.f.?s for penalty parameter bounds ratios of Lasso and Preconditioned Lasso for various subspaces UA . Our theorems focussed on specific settings of PX , Py and ignored others. In general, the gains of HJ and PBHT over Lasso depend on how much the decoy signals in XS c are suppressed and how much of the true signal due to XS is preserved. Further comparison of HJ and PBHT must thus analyze how the subspaces span(UA) are estimated in the context of the assumptions made in [5] and [9]. A final note concerns the dimension of the subspace span(UA). Both HJ and PBHT were proposed with the implicit goal of finding a basis UA that has the same span as US . This of course requires estimating —S— = k by q, which adds another layer of complexity to these algorithms. Theorems 1 and 2 suggest that underestimating k can be more detrimental to signed support recovery than overestimating it. By overestimating q ¿ k, we can trade off milder improvement when span(US) ? span(UA) against poor behavior should we have span(US) 6? span(UA).

5

Model-Based Comparisons

In the previous section we used Lemma 1 in conjunction with assumptions on UA to make statements about HJ and PBHT. Of course, the quality of the estimated UA depends on the specific instances X, ? ? , w, which hinders a general analysis. Similarly, a direct application of Lemma 1 to JR yields bounds that exhibit strong X dependence. It is possible to crystallize prototypical examples by specializing X and w to come from a generative model. In this section we briefly present this model and will show the resulting penalty parameter bounds for JR. 6

5.1

Generative model for X

As discussed in Section 2, many preconditioning algorithms can be phrased as truncating or reweighting column subspaces associated with X [5, 6, 9]. This suggests that a natural generative model for X can be formulated in terms of the SVD of submatrices of X. Assume p ? k ¿ n and let ?S , ?S c be fixed-spectrum matrices of dimension n ? k and n ? p ? k respectively. We will assume throughout this paper that the top left ?diagonal? entries of ?S , ?S c are positive and the remainder is zero. Furthermore, we let U, VS , VS c be orthonormal bases of dimension n ? n, k ? k and p ? k ? p ? k respectively. We assume that these bases are chosen uniformly at random from the corresponding Stiefel manifold. As before and without loss of generality, suppose S = {1, . . . , k}. Then we let the Lasso problem be

y = X? ? + w with X = U ?S VS¿ , ?S c VS¿c w ? N (0, ? 2 In?n), (19)

To ensure that the column norms of X are controlled, we compute the spectra ?S , ?S c by normal? S and ? ? S c with arbitrary positive elements on the diagonal. Specifically, we let izing spectra ? ?S =

?S ? ? kn ? S ——F ——?

?S c =

? Sc p ? (p ? k)n. ? S c ——F ——?

(20)

We verify in the supplementary material that with these assumptions the squared column norms of X are in expectation n (provided the orthonormal bases are chosen uniformly at random). Intuition.

Note that any matrix X can be decomposed using a block-wise SVD as

X = [XS , XS c] = U ?S VS¿ , T ?S c VS¿c ,

(21)

with orthonormal bases U, T, VS , VS c . Our model in Eq. (19) is only a minor restriction of this model, where we set T = In?n . To develop more intuition, let us temporarily set VS = Ik?k , VS c = Ip?k?p?k . Then X = [XS , XS c] = U [?S , ?S c] and we see that up to scaling XS equals the first k columns of XS c . The difficulty for Lasso thus lies in correctly selecting the columns in XS , which are highly correlated with the first few columns in XS c . 5.2

Piecewise constant spectra

For notational clarity we will now focus on a special case of the above model. To begin, we develop some notation. In previous sections we used US to denote a basis for the column space of XS . We will continue to use this notation, and let US contain the first k columns of U . Accordingly, we ? S , ?S c , ? ? Sc denote the last n ? k columns of U by US c . We let the diagonal elements of ?S , ? ?S be identified by their column indices. That is, the diagonal entries ?S,c of ?S and ? ?S,c of ? ? are indexed by c ? {1, . . . , k}; the diagonal entries ?S c ,c of ?S c and ? ?S c ,c of ?S c are indexed by c ? {1, . . . , n}. Each of the diagonal entries in ?S , ?S c is associated with a column of U . The set of diagonal entries of ?S and ?S c associated with US is ?(S) = {1, . . . , k} and the set of diagonal entries in ?S c associated with US c is ?(S c) = {1, . . . , n}?(S). We will construct spectrum matrices ?S , ?S c that are piecewise constant on their diagonals. For some ? ? 0, we let ? ?S,i = 1, ? ?S

c ,i = ? ?i ? ?(S) and ? ?S c ,j = 1 ?j ? ?(S c). ?1/2 ¿ Consequences for JR. Recall that for JR, if X = U DV ¿ , then PX = Py = U DD¿ U . We have the following theorem. Theorem 3. Assume the Lasso problem was generated according to the generative model of Eq. (19) with p ?i ? ?(S), ? ?S,i = 1, ? ?S c ,i = ? and ?j ? ?(S c), ? ?S c ,j = 1 and that ? ? ¡ n ? k/ k(p ? k ? 1). Then the conditions of Lemma 1 hold before and after preconditioning using JR. Moreover, ?u ? (p ? k) ?u = . ? n + p?2 ? k ?l ?l

(22)

In other words, JR deterministically scales the ratio of penalty parameter bounds. The proof idea is as follows. It is easy to see that XS¿ XS is always invertible. Furthermore, one can show that if 7

p ? ? ¡ n ? k/ k(p ? k ? 1), we have —?j — ¡ 1, ?j ? S c and sgn(?i?)?i ¿ 0, ?i ? S. Thus, by our assumptions, the preconditions of Lemma 1 are satisfied for the original Lasso problem. Plugging in the definitions of ?S , ?S c into Eq. (19) we find that the SVD becomes X = U DV ¿ , where U is the same column basis as in Eq. (19), and the diagonal elements of D are determined by ?. Substituting this into the definitions of ? ?j , ??i , ??j , ?i , we have that after preconditioning using JR

n(p ? k)?2 ? ?j = ?j ??i = n + 2 ?i (23) k? + n ? k (k?2 + n ? k) ??j = ?j ?i = i . (24) n(p ? k) Thus, if the conditions of Lemma 1 hold for X, ? ? , they will continue to hold after preconditioning using JR. Furthermore, notice that (2J? ?j ¿ 0K ? 1) ? ? ?j = (2J?j ¿ 0K ? 1) ? ?j . Applying ? u /? ? l as claimed. According to Theorem 3 the ratio ? ? u /? ? l will Lemma 1 then gives the new p ratio ? p be larger than ?u /?l iff ? ¡ 1 ? (n/p). Indeed, if ? = 1 ? (n/p) then PX = Py ? In?n and so JR coincides with standard Lasso. 5.3

Extension to Gaussian ensembles

The construction in Eq. (19) uses an orthonormal matrix U as the column basis of X. At first sight this may appear to be restrictive. However, as we show in the supplementary material, one can construct Lasso problems using a Gaussian basis W m which lead to penalty parameter bounds ratios that converge in distribution to those of the Lasso problem in Eq. (19). For some fixed ? ? , VS , VS c , ?S and ?S c , generate two independent problems: One using Eq. (19), and one according to

1 m y m = X m ? ? + wm with X m = ? W m ?S VS¿ , ?S c VS¿c wm ? N 0, ? 2 Im?m , (25) n n where W m is an m ? n standard Gaussian ensemble. Note that an X so constructed is low rank if n ¡ p. The latter generative model bears some resemblance to Gaussian models considered in Paul et al. [9] (Eq. (7)) and Jia and Rohe [6] (Proposition 2). Note that while the problem in Eq. (19) uses n observations with noise variance ? 2 , Eq. (25) has m observations with noise variance ? 2 m/n. The increased variance is necessary because the matrix W m has expected column length m, while columns in U are of length 1. We will think of n as fixed and will let m ? ?. Let the penalty parameter bounds ratio induced by the problem in Eq. (19) be ?u /?l and that induced by Eq. (25) m be ?m u /?l . Then we have the following result. Theorem 4. Let VS , VS c , ?S , ?S c and ? ? be fixed. If the conditions of Lemma 1 hold for X, ? ? , then for m large enough they will hold for X m , ? ? . Furthermore, as m

? ? ?m u d ?u ? , (26) m ?l ?l where the stochasticity on the left is due to W m , wm and on the right is due to w. Thus, with respect to the bounds ratio ?u /?l , the construction of Eq. (19) can be thought of as the limiting construction of Gaussian Lasso problems in Eq. (25) for large m. As such, we believe that Eq. (19) is a useful proxy for less restrictive generative models. Indeed, as the experiment in Figure 2(b) shows, Theorem 3 can be used to predict the scaling factor for penalty parameter bounds ? u /? ? l / (?u /?l)) with good accuracy even for Gaussian ensembles. ratios (i.e., ?

6

Conclusions

This paper proposes a new framework for comparing Preconditioned Lasso algorithms to the standard Lasso which skirts the difficulty of choosing penalty parameters. By eliminating this parameter from consideration, finite data comparisons can be greatly simplified, avoiding the use of model selection strategies. To demonstrate the framework?s usefulness, we applied it to a number of Preconditioned Lasso algorithms and in the process confirmed intuitions and revealed fragilities and mitigation strategies. Additionally, we presented an SVD-based generative model for Lasso problems that can be thought of as the limit point of a less restrictive Gaussian model. We believe this work to be a first step towards a comprehensive theory for evaluating and comparing Lasso-style algorithms and believe that the strategy can be extended to comparing other penalized likelihood methods on finite datasets. 8

2 References

[1] D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. Information Theory, IEEE Transactions on, 52(1):6?18, 2006. [2] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association, 96:1348?1360, 2001. [3] J.J. Fuchs. Recovery of exact sparse representations in the presence of bounded noise. Information Theory, IEEE Transactions on, 51(10):3601?3608, 2005. [4] H.-C. Huang, N.-J. Hsu, D.M. Theobald, and F.J. Breidt. Spatial Lasso with applications to GIS model selection. Journal of Computational and Graphical Statistics, 19(4):963?983, 2010. [5] J.C. Huang and N. Jojic. Variable selection through Correlation Sifting. In V. Bafna and S.C. Sahinalp, editors, RECOMB, volume 6577 of Lecture Notes in Computer Science, pages 106?123. Springer, 2011. [6] J. Jia and K. Rohe. ?Preconditioning? to comply with the irrepresentable condition. 2012. [7] N. Meinshausen. Lasso with relaxation. Technical Report 129, Eidgen?ossische Technische Hochschule, Z?urich, 2005. [8] N. Meinshausen and P. B?uhlmann. High-dimensional graphs and variable selection with the Lasso. Annals of Statistics, 34(3):1436?1462, 2006. [9] D. Paul, E. Bair, T. Hastie, and R. Tibshirani. ?Preconditioning? for feature selection and regression in high-dimensional problems. Annals of Statistics, 36(4):1595?1618, 2008. [10] R. Tibshirani. Regression shrinkage and selection via the Lasso. Journal

of the Royal Statistical Society, Series B, 58(1):267?288, 1994. [11] R.J. Tibshirani. The solution path of the Generalized Lasso. Stanford University, 2011. [12] M.J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ‘1 -constrained quadratic programming (Lasso). IEEE Transactions on Information Theory, 55(5):2183?2202, 2009. [13] P. Zhao and B. Yu. On model selection consistency of Lasso. Journal of Machine Learning Research, 7:2541?2563, 2006. [14] H. Zou. The Adaptive Lasso and its oracle properties. Journal of the American Statistical Association, 101:1418?1429, 2006. [15] H. Zou and T. Hastie. Regularization and variable selection via the Elastic Net. Journal of the Royal Statistical Society, Series B, 67:301?320, 2005.

9