An Homotopy Algorithm for the Lasso with Online Observations

Authored by:

Laurent E. Ghaoui
Pierre Garrigues

Abstract

It has been shown that the problem of ell_1-penalized least-square regression commonly referred to as the Lasso or Basis Pursuit DeNoising leads to solutions that are sparse and therefore achieves model selection. We propose in this paper an algorithm to solve the Lasso with online observations. We introduce an optimization problem that allows us to compute an homotopy from the current solution to the solution after observing a new data point. We compare our method to Lars and present an application to compressed sensing with sequential observations. Our approach can also be easily extended to compute an homotopy from the current solution to the solution after removing a data point, which leads to an efficient algorithm for leave-one-out cross-validation.

1 Paper Body

It has been shown that the problem of ‘1 -penalized least-square regression commonly referred to as the Lasso or Basis Pursuit DeNoising leads to solutions that are sparse and therefore achieves model selection. We propose in this paper RecLasso, an algorithm to solve the Lasso with online (sequential) observations. We introduce an optimization problem that allows us to compute an homotopy from the current solution to the solution after observing a new data point. We compare our method to Lars and Coordinate Descent, and present an application to compressive sensing with sequential observations. Our approach can easily be extended to compute an homotopy from the current solution to the solution that corresponds to removing a data point, which leads to an efficient algorithm for leave-one-out cross-validation. We also propose an algorithm to automatically update the regularization parameter after observing a new data point.

1

Introduction

Regularization using the ‘1 -norm has attracted a lot of interest in the statistics [1], signal processing [2], and machine learning communities. The ‘1 penalty indeed leads to sparse solutions, which is a desirable property to achieve model selection, data compression, or for obtaining interpretable results. In this paper, we focus on the problem of ‘1 -penalized least-square regression commonly referred to as the Lasso [1]. We are given a set of training examples or observations (yi , xi) ? R ? Rm , i = 1 . . . n. We wish to fit a linear model to predict the response yi as a function of xi and a feature vector ? ? Rm , yi = xTi ? + ?i , where ?i represents the noise in the observation. The Lasso optimization problem is given by n

min ?

1X T (x ? ? yi)2 + ?n k?k1 , 2 i=1 i

(1)

where ?n is a regularization parameter. The solution of (1) is typically sparse, i.e. the solution ? has few entries that are non-zero, and therefore identifies which dimensions in xi are useful to predict yi . The ‘1 -regularized least-square problem can be formulated as a convex quadratic problem (QP) with linear equality constraints. The equivalent QP can be solved using standard interior-point methods (IPM) [3] which can handle medium-sized problems. A specialized IPM for large-scale problems was recently introduced in [4]. Homotopy methods have also been applied to the Lasso to compute the full regularization path when ? varies [5] [6][7]. They are particularly efficient when the solution is very sparse [8]. Other methods to solve (1) include iterative thresholding algorithms [9][10][11], feature-sign search [12], bound optimization methods [13] and gradient projection algorithms [14]. 1

We propose an algorithm to compute the solution of the Lasso when the training examples (yi , xi)i=1...N are obtained sequentially. Let ?(n) be the solution of the Lasso after observing n training examples and ?(n+1) the solution after observing a new data point (yn+1 , xn+1) ? R ? Rm . We introduce an optimization problem that allows us to compute an homotopy from ?(n) to ?(n+1) . Hence we use the previously computed solution as a ?warm-start?, which makes our method particularly efficient when the supports of ?(n) and ?(n+1) are close. In Section 2 we review the optimality conditions of the Lasso, which we use in Section 3 to derive our algorithm. We test in Section 4 our algorithm numerically, and show applications to compressive sensing with sequential observations and leave-one-out cross-validation. We also propose an algorithm to automatically select the regularization parameter each time we observe a new data point.

2

Optimality conditions for the Lasso

The objective function in (1) is convex and non-smooth since the ‘1 norm is not differentiable when ?i = 0 for some i. Hence there is a global minimum at ? if and only if the subdifferential of the objective function at ? contains the 0-vector. The subdifferential of the ‘1 -norm at ? is the following set

vi = sgn(?i) if —?i — ¿ 0 ?k?k1 = v ? Rm : . vi ? [?1, 1] if ?i = 0 Let X ? Rn?m be the matrix whose ith row is equal to xTi , and y = (y1 , . . . , yn)T . The optimality conditions for the Lasso are given by X T (X? ? y) + ?n v = 0, v ? ?k?k1 . We define as the active set the indices of the elements of ?

that are non-zero. To simplify notations we assume that the active set appears first, i.e. ?T = (?1T , 0T) and v T = (v1T , v2T), where v1i = sgn(?1i) for all i, and ?1 ? v2j ? 1 for all j. Let X = (X1 X2) be the partitioning of X according to the active set. If the solution is unique it can be shown that X1T X1 is invertible, and we can rewrite the optimality conditions as

?1 = (X1T X1)?1 (X1T y ? ?n v1) . ??n v2 = X2T (X1 ?1 ? y) Note that if we know the active set and the signs of the coefficients of the solution, then we can compute it in closed form.

3 3.1

Proposed homotopy algorithm Outline of the algorithm

Suppose we have computed the solution ?(n) to the Lasso with n observation and that we are given an additional observation (yn+1 , xn+1) ? R ? Rm . Our goal is to compute the solution ?(n+1) of the augmented problem. We introduce the following optimization problem

2 1 X y

?(t, ?) = arg min ?? (2)

+ ?k?k1 . T ty tx

n+1 2 n+1 ? 2

(n)

(n+1)

We have ? = ?(0, ?n) and ? from ?(n) to ?(n+1) in two steps:

= ?(1, ?n+1). We propose an algorithm that computes a path

Step 1 Vary the regularization parameter from ?n to ?n+1 with t = 0. This amounts to computing the regularization path between ?n and ?n+1 as done in Lars. The solution path is piecewise linear and we do not review it in this paper (see [15][7][5]). Step 2 Vary the parameter t from 0 to 1 with ? = ?n+1 . We show in Section 3.2 how to compute this path.

2

3.2

Algorithm derivation

We show in this Section that ?(t, ?) is a piecewise smooth function of t. To make notations lighter we write ?(t) := ?(t, ?). We saw in Section 2 that the solution to the Lasso can be easily computed once the active set and signs of the coefficients are known. This information is available at t = 0, and we show that the active set and signs will remain the same for t in an interval [0, t?) where the solution ?(t) is smooth. We denote such a point where the active set changes a ?transition point? and show how to compute it analytically. At t? we update the active set and signs which will remain valid until t reaches the next transition point. This process is iterated until we know the active set and signs of the solution at t = 1, and therefore can compute the desired solution ?(n+1) . We suppose as in Section 2 and without loss of generality that the solution at t = 0 is such that ?(0) = (?1T , 0T) and v T = (v1T , v2T) ? ?k?(0)k1 satisfy the optimality conditions. Lemma 1. Suppose ?1i 6= 0 for all i and —v2j — ¡ 1 for all j. There exist t? ¿ 0 such that for all t ? [0, t?), the solution of (2) has the same support and the same sign as ?(0). P ROOF. The optimality conditions of (2) are given by

X T (X? ? y) + t2 xn+1 xTn+1 ? ? yn+1 + ?w = 0,

(3)

where w ? ?k?k1 . We show that there exists a solution ?(t)T = (?1 (t)T , 0T) and w(t)T = (v1T , w2 (t)T) ? ?k?(t)k1 satisfying the optimality conditions for t sufficiently small. We partition xTn+1 = (xTn+1,1 , xTn+1,2) according to the active set. We rewrite the optimality conditions as

T X1 (X1 ?1 (t) ? y) + t2 xn+1,1 xn+1,1 T ?1 (t) ? yn+1 + ?v1 = 0

. X2T (X1 ?1 (t) ? y) + t2 xn+1,2 xn+1,1 T ?1 (t) ? yn+1 + ?w2 (t) = 0 Solving for ?1 (t) using the first equation gives ?1 (t) = X1T X1 + t2 xn+1,1 xn+1,1 T

?1

X1T y + t2 yn+1 xn+1,1 ? ?v1 .

(4)

We can see that ?1 (t) is a continuous function of t. Since ?1 (0) = ?1 and the elements of ?1 are all strictly positive, there exists t?1 such that for t ¡ t?1 , all elements of ?1 (t) remain positive and do not change signs. We also have

??n+1 w2 (t) = X2T (X1 ?1 (t) ? y) + t2 xn+1,2 xn+1,1 T ?1 (t) ? yn+1 . (5) Similarly w2 (t) is a continuous function of t, and since w2 (0) = v2 , there exists t?2 such that for t ¡ t?2 all elements of w2 (t) are strictly smaller than 1 in absolute value. By taking t? = min(t?1 , t?2) we obtain the desired result. The solution ?(t) will therefore be smooth until t reaches a transition point where either a component of ?1 (t) becomes zero, or one of the component of w2 (t) reaches one in absolute value. We now show how to compute the value of the transition point.

X y ? = ? ? ? Let X and y ? = . We partition X = X X 1 2 according to the active yn+1 xn+1 T set. We use the Sherman-Morrison formula and rewrite (4) as ?1 (t) = ??1 ?

(t2 ? 1)? e u, 1 + ?(t2 ? 1)

?T X ? 1)?1 (X ? T y? ? ?v1), e? = xn+1,1 T ??1 ? yn+1 , ? = xn+1,1 T (X ?T X ? 1)?1 xn+1,1 and where ??1 = (X 1 1 1 T ? ?1 ? u = (X1 X1) xn+1,1 . Let t1i the value of t such that ?1i (t) = 0. We have

?1 ! 21 e?ui t1i = 1 + ?? , ??1i We now examine the case where a component of w2 (t) reaches one in absolute value. We first notice that (xn+1,1 T ?1 (t) ? yn+1 = 1+?(te?2 ?1) 2 , e ? ? 1 ?1 (t) ? y? = e? ? (t ?1)? X X1 u 2 1+?(t ?1)

3

? 1 ??1 ? y?. We can rewrite (5) as where e? = X ? 2T e? + ??w2 (t) = X e?(t2 ? 1) ? 2T X ? 1 u). (xn+1,2 ? X 1 + ?(t2 ? 1)

? 2 , and x(j) the j th element of xn+1,2 . The j th component of w2 (t) Let cj be the j th column of X will become 1 in absolute value as soon as

2 T (j) T ? cj e? + e?(t ? 1) = ?. x ? c X u 1 j

1 + ?(t2 ? 1) ? Let t+ 2 j (resp. t2 j) be the value such that w2j (t) = 1 (resp. w2j (t) = ?1). We have ? ?1 ! 12

? ? ? e?(x(j) ?cT ?+ j X1 u) ? ?? ? ?t2 j = 1 + ? ???cT j e ?1 ! 12 .

? (j) T ? ? e ? (x ?c X u) ? 1 ? j ? ?? ? ?t2 j = 1 + ? ??cT j e ? Hence the transition point will be equal to t0 = min{mini t1i , minj t+ 2 j , minj t2 j

} where we restrict ourselves to the real solutions that lie between 0 and 1. We now have the necessary ingredients to derive the proposed algorithm.

Algorithm 1 RecLasso: homotopy algorithm for online Lasso 1: Compute the path from ? (n) = ?(0, ?n) to ?(0, ?n+1). 2: Initialize the active set to the non-zero coefficients of ?(0, ?n+1) and let v = sign (?(0, ?n+1)).

? 1 the Let v1 and xn+1,1 be the subvectors of v and xn+1 corresponding to the active set, and X ? submatrix of X whose columns correspond to the active set. ?T X ? 1)?1 (X ? T y? ? ?v1). Initialize ??1 = (X 1 1 Initialize the transition point t0 = 0. 3: Compute the next transition point t0 . If it is smaller than the previous transition point or greater than 1, go to Step 5. Case 1 The component of ?1 (t0) corresponding to the ith coefficient goes to zero: Remove i from the active set. Update v by setting vi = 0. Case 2 The component of w2 (t0) corresponding to the j th coefficient reaches one in absolute value: Add j to the active set. If the component reaches 1 (resp. ?1), then set vj = 1 (resp. vj = ?1). ? 1 and xn+1,1 according to the updated active set. 4: Update v1 , X ? ?T X ? 1)?1 (X ? T y? ? ?v1) (rank 1 update). Update ?1 = (X 1 1 Go to Step 3. 5: Compute final value at t = 1, where the values of ? (n+1) on the active set are given by ??1 . The initialization amounts to computing the solution of the Lasso when we have only one data point (y, x) ? R ? Rm . In this case, the active set has at most one element. Let i0 = arg maxi —x(i) — and v = sign(yx(i0)). We have (1 (i0) ? ?1 v)ei0 if —yx(i0) — ¿ ?1 (i) 2 (yx ?(1) = (x 0) . 0 otherwise We illustrate our algorithm by showing the solution path when the regularization parameter and t are successively varied with a simple numerical example in Figure 1. 3.3

Complexity

?T X ? 1 at each transition The complexity of our algorithm is dominated by the inversion of the matrix X 1 point. The size of this matrix is bounded by q = min(n, m). As the update to this matrix after a 4

Figure 1: Solution path for both steps of our algorithm. We set n = 5, m = 5, ?n = .1n. All the values of X, y, xn+1 and yn+1 are drawn at random. On the left is the homotopy when the regularization parameter goes from ?n = .5 to ?n+1 = .6. There is one transition point as ?2 becomes inactive. On the right is the piecewise smooth path of ?(t) when t goes from 0 to 1. We can see that ?3 becomes zero, ?2 goes from being 0 to being positive, whereas ?1 , ?4 and ?5 remain active with their signs unchanged. The three transition points are shown as black dots. transition point is rank 1, the cost of computing the inverse is O(q 2). Let k be the total number of transition points after varying the regularization parameter from ?n to ?n+1 and t from 0 to 1. The complexity of our algorithm is thus O(kq 2). In practice, the size of the active set d is much lower than q, and if it remains ? d throughout the homotopy, the complexity is O(kd2). It is instructive to compare it with the complexity of recursive least-square, which corresponds to ?n = 0 for all n and n ¿ m. For this problem the solution typically has m non-zero elements, and therefore the cost of updating the solution after a new observation is O(m2). Hence if the solution is sparse (d small) and the active set does not change much (k small), updating the solution of the Lasso will be faster than updating the solution to the non-penalized least-square problem. Suppose that we applied Lars directly to the problem with n + 1 observations without using knowledge of ?(n) by varying the regularization parameter from a large value where the size of the active set is 0 to ?n+1 . Let k 0 be the number of transition points. The complexity of this approach is O(k 0 q 2), and we can therefore compare the efficiency of these two approaches by comparing the number of transition points.

4 4.1

Applications Compressive sensing

Let ?0 ? Rm be an unknown vector that we wish to reconstruct. We observe n linear projections yi = xTi ?0 + ?i , where ?i is Gaussian noise of variance ? 2 . In general one needs m such measurement to reconstruct ?0 . However, if ?0 has a sparse representation with k non-zero coefficients, it has been shown in the noiseless case that it is sufficient to use n ? k log m such measurements. This approach is known as compressive sensing [16][17] and has generated a tremendous amount of interest in the signal processing community. The reconstruction is given by the solution of the Basis Pursuit (BP) problem min k?k1 subject to X? = y. ?

If measurements are obtained sequentially, it is advantageous to start estimating the unknown sparse signal as measurements arrive, as opposed to waiting for a specified number of measurements. Algorithms to solve BP with sequential measurements have been proposed in [18][19], and it has been shown that the change in the active set gives a criterion for how many measurements are needed to recover the underlying signal [19]. In the case where the measurements are noisy (? ¿ 0), a standard approach to recover ?0 is to solve the Basis Pursuit DeNoising problem instead [20]. Hence, our algorithm is well suited for 5

compressive sensing with sequential and noisy measurements. We compare our proposed algorithm to Lars as applied to the entire dataset each time we receive a new measurement. We also compare our method to coordinate descent [11] with warm start: when receiving a new measurement, we initialize coordinate descent (CD) to the actual solution. We sample measurements of a model where m = 100, the vector ?0 used to sample the data has 25 non- zero elements whose values are Bernoulli ?1, xi ? N (0, Im), ? = 1, and we set ?n = .1n. The reconstruction error decreases as the number of measurements grows (not plotted). The parameter that controls the complexity of Lars and RecLasso is the number of transition points. We see in Figure 2 that this quantity is consistently smaller for RecLasso, and that after 100 measurements when the support of the solution does not change much there are typically less than 5 transition points for RecLasso. We also show in Figure 2 timing comparison for the three algorithms that we have each implemented in Python. We observed that CD requires a lot of iterations to converge to the optimal solution when n ¡ m, and we found difficult to set a stopping criterion that ensures convergence. Our algorithm is consistently faster than Lars and CD with warm- start.

Figure 2: Compressive sensing results. On the x- axis of the plots are the iterations of the algorithm, where at each iteration we receive a new measurement. On the left is the comparison of the number of transition points for Lars and RecLasso, and on the right is the timing comparison for the three algorithms. The simulation is repeated 100 times and shaded areas represent one standard deviation. 4.2

Selection of the regularization parameter

We have supposed until now a pre- determined regularization schedule, an assumption that is not practical. The amount of regularization depends indeed on the variance of the noise present in the data which is not known a priori. It is therefore not obvious how to determine the amount of regularization. We write ?n = n?n such that ?n is the weighting factor between the average meansquared error and the ‘1 - norm. We propose an algorithm that selects ?n in a data-driven manner. The problem with n observations is given by n

1 X T ?(?) = arg min (xi ? ? yi)2 + ?k?k1 . 2n ? i=1 We have seen previously that ?(?) is piecewise linear, and we can therefore compute its gradient unless ? is a transition point. Let err(?) = (xTn+1 ?(?)?yn+1)2 be the error on the new observation. We propose the following update rule to select ?n+1 log ?n+1

=

? ?n+1

=

?err (?n) log ?n ? ? ? log ? n o ?n ? exp 2n?xTn+1,1 (X1T X1)?1 v1 (xTn+1 ?1 ? yn+1) ,

where the solution after n observations corresponding to the regularization parameter ?n is given by (?1T , 0T), and v1 = sign(?1). We therefore use the new observation as a test set, which allows us to update the regularization parameter before introducing the new observation by varying t from 0 6

to 1. We perform the update in the log domain to ensure that ?n is always positive. We performed simulations using the same experimental setup as in Section 4.1 and using ? = .01. We show in Figure 3 a representative example where ? converges. We compared this value to the one we would obtain if we had a training and a test set with 250 observations each such that we could fit the model on the training set for various values of ?, and see which one gives the smallest prediction error on the test set. We obtain a very similar result, and understanding the convergence properties of our proposed update rule for the regularization parameter is the object of current research. 4.3

Leave-one-out cross-validation

We suppose in this Section that we have access to a dataset (yi , xi)i=1...n and that ?n = n?. The parameter ? is tied to the amount of noise in the data which we do not know a priori. A standard approach to select this parameter is leave-one-out cross-validation. For a range of values of ?, we use n ? 1 data points to solve the Lasso with regularization parameter (n ? 1)? and then compute the prediction error on the data point that was left out. This is repeated n times such that each data point serves as the test set. Hence the best value for ? is the one that leads to the smallest mean prediction error. Our proposed algorithm can be adapted to the case where we wish to update the solution of the Lasso after a data point is removed. To do so, we compute the first homotopy by varying the regularization parameter from n? to (n ? 1)?. We then compute the second homotopy by varying t from 1 to 0 which has the effect of removing the data point that will be used for testing. As the algorithm is very similar to the one we proposed in Section 3.2 we omit the derivation. We sample a model with n = 32 and m = 32. The vector ?0 used to generate the data has 8 non-zero elements. We add Gaussian noise of variance 0.2 to the observations, and select ? for a range of 10 values. We show in Figure 4 the histogram of the number of transition points for our algorithm when solving the Lasso with n ? 1 data points (we solve this problem 10 ? n times). Note that in the majority cases there are very few transition points, which makes our approach very efficient in this setting.

Figure 3: Evolution of the regularization parameter when using our proposed update rule.

5

Figure 4: Histogram of the number of transition points when removing an observation.

Conclusion

We have presented an algorithm to solve ‘1 -penalized least-square regression with online observations. We use the current solution as a ?warm-start? and introduce an optimization problem that allows us to compute an homotopy from the current solution to the solution after observing a new data point. The algorithm is particularly efficient if the active set does not change much, and we show a computational advantage as compared to Lars and Coordinate Descent with warm-start for applications such as compressive sensing with sequential observations and leave-one-out cross-validation. We have also proposed an algorithm to automatically select the regularization parameter where each new measurement is used as a test set. 7

Acknowledgments We wish to acknowledge support from NSF grant 0835531, and Guillaume Obozinski and Chris Rozell for fruitful discussions.

2 References

[1] R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B, 58(1):267?288, 1996. [2] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. SIAM Review, 43(1):129? 159, 2001. [3] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge Univ. Press, 2004. [4] S-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. IEEE Journal of Selected Topics in Signal Processing, 1(4):606?617, 2007. [5] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. Annals of Statistics, 32(2):407?499, 2004. [6] M.R. Osborne, B. Presnell, and B.A. Turlach. A new approach to variable selection in least squares problems. IMA Journal of Numerical Analysis, 20:389?404, 2000. [7] D.M. Malioutov, M. Cetin, and A.S. Willsky. Homotopy continuation for sparse signal representation. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Philadelphia, PA, March 2005. [8] I. Drori and D.L. Donoho. Solution of ‘1 minimization problems by lars/homotopy

methods. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, May 2006. [9] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on Pure and Applied Mathematics, 57:1413?1541, 2004. [10] C.J. Rozell, D.H. Johnson, R.G. Baraniuk, and B.A. Olshausen. Locally competitive algorithms for sparse approximation. In Proceedings of the International Conference on Image Processing (ICIP), San Antonio, TX, September 2007. [11] J. Friedman, T. Hastie, H. Hoefling, and R. Tibshirani. Pathwise coordinate optimization. The Annals of Applied Statistics, 1(2):302?332, 2007. [12] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. In Proceedings of the Neural Information Processing Systems (NIPS), 2007. [13] M. Figueiredo and R. Nowak. A bound optimization approach to wavelet-based image deconvolution. In Proceedings of the International Conference on Image Processing (ICIP), Genova, Italy, September 2005. [14] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE Journal of Selected Topics in Signal Processing, 1(4):586?597, 2007. [15] M Osborne. An effective method for computing regression quantiles. IMA Journal of Numerical Analysis, Jan 1992. [16] E. Cand‘es. Compressive sampling. Proceedings of the International Congress of Mathematicians, 2006. [17] D.L. Donoho. Compressed sensing. IEEE Transactions on Information Theory, 52(4):1289?1306, 2006. [18] S. Sra and J.A. Tropp. Row-action methods for compressed sensing. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, May 2006. [19] D. Malioutov, S. Sanghavi, and A. Willsky. Compressed sensing with sequential observations. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, NV, March 2008. [20] Y. Tsaig and D.L. Donoho. Extensions of compressed sensing. Signal Processing, 86(3):549?571, 2006.

8