Reading Report of Van Den Berg et al.

10/10

(ECE 695, Reading Assignment 03, Spring 2018)

Overview of the Paper

In this paper [1], the ℓ_1-regularized problem of interest is basis pursuit denoise (BPDN) for solving an underdetermined linear system:

$$\text{minimize } \|x\|_1, \text{ subject to } \|Ax - b\|_2 \leq \sigma. \tag{1}$$

When compared to its peers such as LASSO and QP, BPDN is often preferred in practice because of the lack of a priori knowledge on sparsity and the availability of an approximate noise level. The proposed method, dubbed Pareto Probing (PP), solves BPDN indirectly by efficiently finding a point on the Pareto curve that resides on the boundary of the two-norm residual constraint in BPDN.

Prior Work

Homotopy and Lars for BP rely on solving the QP subproblem repeatedly for nearly all values of λ, descending from λ_{max}. Therefore, their overall performance is prone to the complexity of the least-square subproblem, which could become prohibitively expensive when A or part of A is needed explicitly. Along similar lines, the efficiency of Interior Point (IP) methods, which pose BP as a special case of a cone program, also predicates on the availability of the explicit representation of A. In the special case of $\sigma = 0$, where BP is reduced to a linear program, one can adopt IP solvers but they are shown to be less competitive.

Key Ideas of the Paper

The Pareto curve plots the optimal trade off between the two-norm residual and the one-norm of the solution. As proven by the authors, the Pareto curve is non-increasing and differentiable in the interior of the first quadrant. Consequently, BP can be solved exactly albeit indirectly by computing the root of the equivalence constraint on the residual $\phi(\tau) \equiv \|Ax_\tau - b\|_2 = \sigma$, where τ is the regularization parameter of the LASSO instance that yields the same optimal x.

While the root-finding is handled by Newton-Raphson in PP, the approach for computing the gradient $\phi(\tau)'$ used in finding the Newton step is novel. In fact, the residual gradient can be derived as $\phi(\tau)' = -\lambda_\tau$, where λ_τ is the Lagrangian multiplier in the dual problem of LASSO. The value of the dual variable λ_τ can be calculated in closed-form using the optimal solution of the primal. Additionally, the authors have demonstrated that the convergence rate of Newton-Raphson is proportional to the size of the duality gap. Therefore, the approach used in PP to solve LASSO efficiently and as accurately as possible constitutes another major contribution of the paper.

The method of choice for solving LASSO is Spectral Projected-Gradient (SPG). Within each iteration, a line search with the initial step size computed with the spectral Barzilai and Borwein method is conducted to find the next iterate of x, while ascertaining the LASSO sparsity constraint by projecting x onto the feasible set of $|x| \leq \tau$. The projection is done by subtracting a scalar d from x_i such that the one-norm of the new vector is bound-reinforced. In addition, the iterative calculation for d is optimized for speed, meanwhile $\|d\|_2$ is being minimized to reduce the cost.

Comments

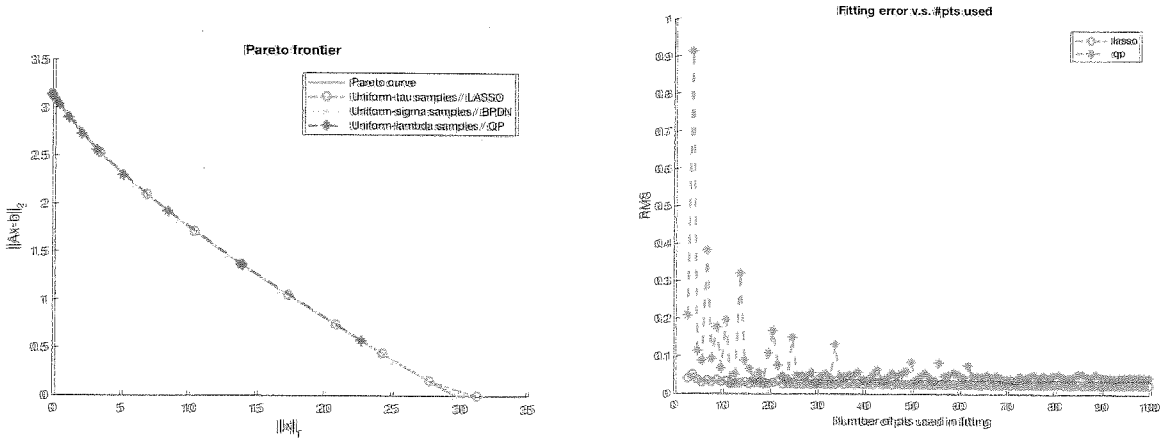
The main focus of my experiments is on the sampling of the Pareto frontier. Since computing the Pareto curve in high-resolution is often impractical, study on the optimal trade-off between the residual and one-norm is thus often carried out via sampling.

My first attempt is to reproduce the results regarding Pareto curve sampling, on a 1D sparse signal reconstruction problem. In the paper, the claim is that uniform sampling is more representative via BP than QP and my results turn out to support the authors' claim, as shown in Figure 1a. In fact, this should

not come as a surprise since the uniformly spaced constraint parameter σ in BP directly corresponds to the x-axis of the Pareto curve. Similarly, uniform τ values in LASSO also produce samples on the Pareto curve that is evenly spaced with respect to the y-axis. On the other hand, solving QP with uniform λs and the relative tolerance set to a reasonably small value, the samples taken by the *l1_ls* algorithm tend to aggregate near the y-axis, which will negatively affect the curve fitting quality. Now the question is, how much is the impact?

The quality of the Pareto curve fitting using LASSO and QP samples is visualized in Figure 1b, in which every marker represents a second-degree polynomial fitting attempt. The x- and y-axis represent the number of samples used to calculate the polynomial coefficients and the RMS error of the fitted Pareto curve v.s. the full resolution Pareto curve. For each fitting attempt, a random subset of samples is chosen from the 100 samples generated using uniform λ or τ values. The ground truth full resolution curve is composed of 100 samples generated from uniformly distributed τ values. The contrast is striking – at least around 30 samples are required for QP to produce an acceptable approximation of the underlying near-linear Pareto curve, while the same fitting quality can be matched with less than 10 samples if LASSO is used. Therefore, the conclusion can be reached that uniform sampling with σ is more representative of the Pareto curve than with λ.

Lastly, I wish to mention that their code is painfully outdated and nearly zero provided examples ran upfront without throwing an exception.



(a) (b)

References

[1] Ewout Van Den Berg and Michael P Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

Reading Report of Berg et al.

(ECE 695, Reading Assignment 03, Spring 2018)

Overview of the Paper

The paper proposed a method for solving basis pursuit problems by exploiting the convex and continuous properties of the Pareto curve. Because of these properties of the Pareto curve,it is possible to find the Lasso problem that would produce the same optimal solution to the original basis pursuit problem. The paper proves that an iterative approach can be adopted by using the residual and its derivative to gradually arrive at the right τ for the Lasso problem so that it will generate the exactly same optimal solution as the basis pursuit. Paper uses approximation when finding the residual and its derivative so that only matrix operations are needed. And it is shown in the experiments section that this method can be applied to large scale optimization problems. For prove of theorem 3.1. the paper assumes $\boldsymbol{A}$ to have full rank.

Prior Work

Because of the nondifferentiable nature of ℓ_1 regularized problems, generic methods such as ellipsoid method and subgradient methods can be used while being slow. And since ℓ_1 LSP can also be converted into a convex quadratic problem with linear inequality constraints, it can be solved by convex optimization solvers such as interior point method like MOSEK for small or medium sized problems. Specialized methods that exploits matrix-vector operations of $\boldsymbol{A}$ and $\boldsymbol{A}^T$ are able to handle large scale problems. Many homotopy approaches such as LARS can be used for solving basis pursuit problems. Interior-point method can also be used for solving the problem when the matrices are given explicitly. General cone program such as SEDUMI and MOSEK can also be used. In the special case where $\sigma = 0$, interior-points methods are quite effective. PDCO is also a possible candidate when $\boldsymbol{A}$ is used as an operator, but it often requires a lot matrix-vector operations to converge, which makes it not feasible for solving large-scale problems. There are also methods that solves the problem by sampling the Pareto curve.

Key Ideas of the Paper

Since the key of the proposed algorithm requires solving a sequence of Lasso problems, the first key point would be the spectral projected-gradient(SPG) method for approximating the solution. Possible the most important part of SPG is the line search of an solution such that the projection of the solution on the 1-norm sphere has a small residual. Specifically the projection operator is defined as

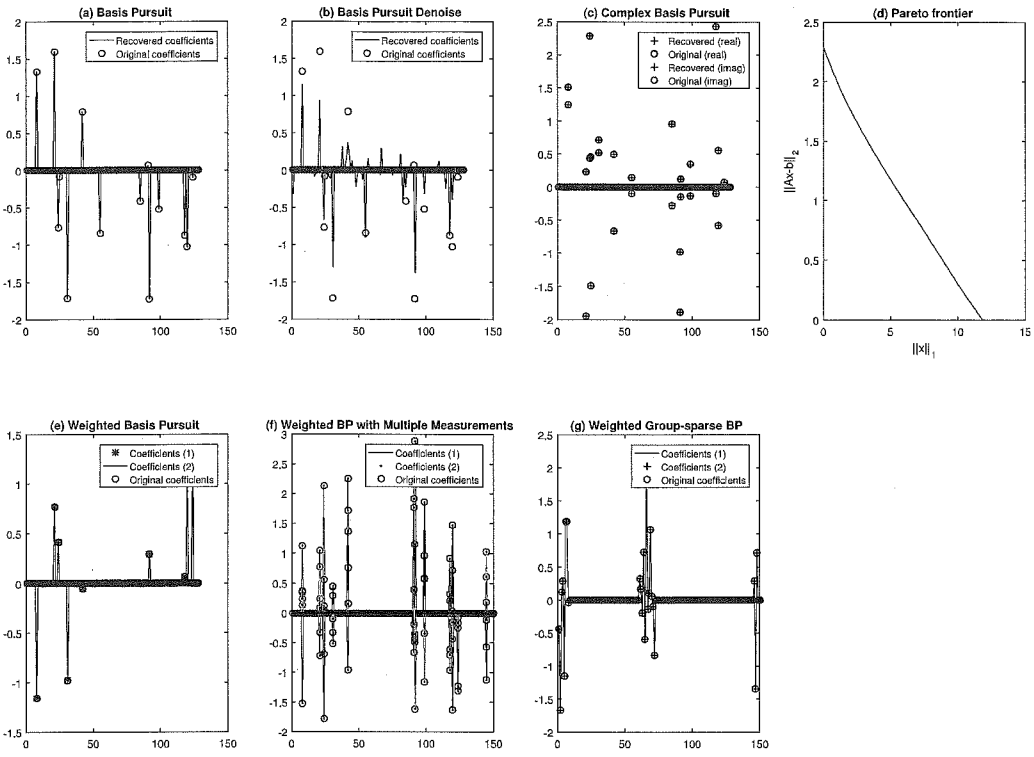
$$\boldsymbol{P}_\tau[c] = \{\underset{x}{\operatorname{argmin}} ||c - x||_2 \text{ subject to } ||x||_1 \leq \tau\}$$

This is reasonable since we want the solution to have a low 1-norm, which is exactly what we want to minimize in the original basis pursuit problem.

The second key idea is the formulation of the dual problem. The dual problem not only provides insights on the Pareto curve, it also provides a way to calculate ϕ' using the Lagrange multipliers and is crucial for derivation of the update rule for the τ parameter using the approximation $\bar{\phi}$ and $\bar{\phi}'$ in the proposed iterative algorithm.

The third key idea is the root finding method. The method uses Newton iteration to update τ used by the Lasso problem so that the solution of the Lasso problem will converge to the basis pursuit solution as defined by the parameter σ. The gradient term in the Newton iteration can be written in terms of the residual ϕ and its gradient ϕ', which can then be approximated using the SPG to solve the Lasso problem. It is also shown in the paper, the SPG only needs matrix operations when solving the approximate solution, which is desirable for large-scale problems.

The fourth key is the stopping criteria used in the line search algorithm. It not only lets the residual to compare with the residuals from several previous iterations which can makes the search more progressive but also considers the gradient of the least square term and the direction in which the solutions are heading to which is sort if conservative.



Comments

I ran the code provided by the author on a sparse signal reconstruction experiment. The result for the basis pursuit(BP) case is perfect. For the basis pursuit denoise(BPD) case the result is a little bit different from the ground truth, which is reasonable since the input signal is corrupted by noise. It is worth noting that the BPD case used only about half the iterations compared to the BP case, which might the counter intuitive considering the BPD is actually a harder problem and more iterations should be used. However, since the BP case corresponds to a σ value of 0 which in turn corresponds to a large τ based on the Pareto plot, this means with the same initial $\tau = 0$, it will takes longer time to reach τ_σ. The BPD case on the other hand has a larger σ value that corresponds to a smaller τ, which makes the convergence faster due to the same initial $\tau = 0$. As a result, it seems the progress of τ is rather linear, the initial τ would be a significant limitation on the runtime of the algorithm. It might be reasonable to use different τ_0 for difference problem formulations for faster convergence. For example, it might be worthwhile to design some mapping function or look up table that will tell us a better initial τ based on the σ value rather than blindly starting from 0.

For each case of weighted basis pursuit, weighted basis pursuit with multiple measurements and weighted group-sparse basis pursuit, two variants are tested. For the first variant where weights are applied on the $\boldsymbol{A}(\boldsymbol{A}\boldsymbol{W}^{-1}$ is used as the forward model), the algorithm was not able to find a solution within 1000 iterations, while for the second case where weights was not applied on $\boldsymbol{A}$ the algorithm was able to find the solution in less than 200 iterations. Further digging into the code, the $\boldsymbol{A}$ matrix was a unitary matrix generated using the orthogonal-triangular decomposition of a randomly generated matrix. Applying the weights to $\boldsymbol{A}$ apparently makes it not unitary any more as the weights are generated randomly as well. This means the proposed algorithm probably converges faster for a particular group of $\boldsymbol{A}$ matrices. My guess is that the infinity norm used in calculating the Lagrange multiplier $\lambda_\tau = ||\boldsymbol{A}^T y_\tau||_\infty$ is calculated using finding the maximum element of the matrix-vector product. When a randomly generated weight matrix is applied to $\boldsymbol{A}$ it will drastically change the behavior of the infinity norm operator.

Compared with the methods in the previous two assignments, this proposed method seems to be faster in finding the solution of the basis pursuit problem.

Reading Report of Van de Berg et al.

(ECE 695, Reading Assignment 03, Spring 2018)

Overview of the Paper

This paper introduces a method of solving basis pursuit denoise (BPDN) by finding points on the Pareto curve. This is done by finding roots of $\phi(\tau) = \sigma$ with Newton's method (where ϕ is a parameterization of the Pareto curve and σ is estimated noise). This in turn is done by finding the dual solution to a LASSO problem which is found by minimizing LASSO with spectral projected-gradient (SPG).

Prior Work

A few other methods the paper talks about for solving BPDN include homotopy methods that solve many quadratic programming problems to find a value λ that can be used to solve BPDN. But obtaining an accurate solution to the quadratic programming problem can eventually become expensive. Interior-point methods can also be used to solve BPDN. A variant of these methods was the subject of a previous assignment, but the paper mentions that we must explicitly know the matrices involved (e.g. A) for them to work effectively. We also already reported on another method known as projected gradient for solving the quadratic programming formulation of L_1 minimization. This paper incorporates the projected gradient SPG specifically for solving, of course, the BPDN formulation of L_1 minimization.

Key Ideas of the Paper

The overarching goal of the paper is to solve BPDN. They use a function ϕ to represent the Pareto curve and try to find points on the curve that will solve BPDN by finding roots of $\phi(\tau) = \sigma$ with Newtons method. Using Newton's method, they find a series of τ, with the hope that they converge in such a way that the solution to LASSO is also the solution to BPDN for that converged τ, τ_σ.

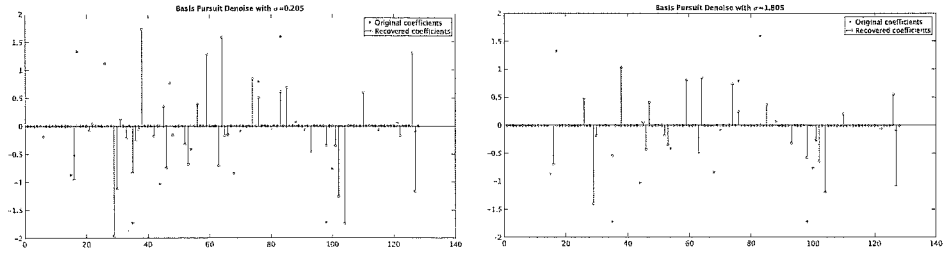
For an optimization problem, the Pareto curve (as the paper says it) gives the optimal trade-off between the one-norm of its solution and the two-norm of its residual (here the residual is give by $||Ax - b||$). By parameterizing the Pareto curve with ϕ as a function of the LASSO regularization parameter τ, we can find the optimal solution to LASSO. Because BPDN and LASSO are different formulations of L_1 regularization, we can use the Pareto curve to find the optimal solution for BPDN as well. We are able to find points on the Pareto curve using our root-finding algorithm because ϕ is convex, strictly decreasing, and differentiable with respect to τ. The paper goes on to prove convexity by reformulating ϕ and showing similarities between the reformulation and the conjugate functions used to find the LASSO dual. It proves that ϕ is differentiable by showing the subgradient at τ is unique. It shows that ϕ is strictly decreasing by showing that it is greater than zero for all τ in our region of interest along with the fact that it is differentiable.

In order to use Newton's method to find the roots of ϕ, we must approximate ϕ and ϕ' so we can update our τ. To do this, we must find the dual solution to LASSO, which in turn can be done by minimizing LASSO using SPG. Even though we are only approximating ϕ and ϕ', the paper shows that we can still converge with Newton's method as the duality gap decreases. The paper goes into detail finding the dual of LASSO and the optimality conditions needed for the primal-dual solution. It also describes how the duality gap can be used to determine how fast the algorithm converges since it effects the accuracy of our $\phi(\tau)$ estimate.

The SPG algorithm used to minimize LASSO involves looking for projected gradients that will result in a sufficiently small residual, then updating our solution, residual and gradient based on this. Once the LASSO duality gap is small enough, we stop. Because the one-norm projection of the gradient can be expensive to compute, the paper gives an algorithm for this projection which involves decreasing the norm of the vector we want to project and finding a solution based on this decrease. The paper also provides an algorithm for the projection onto the complex domain.

Comments

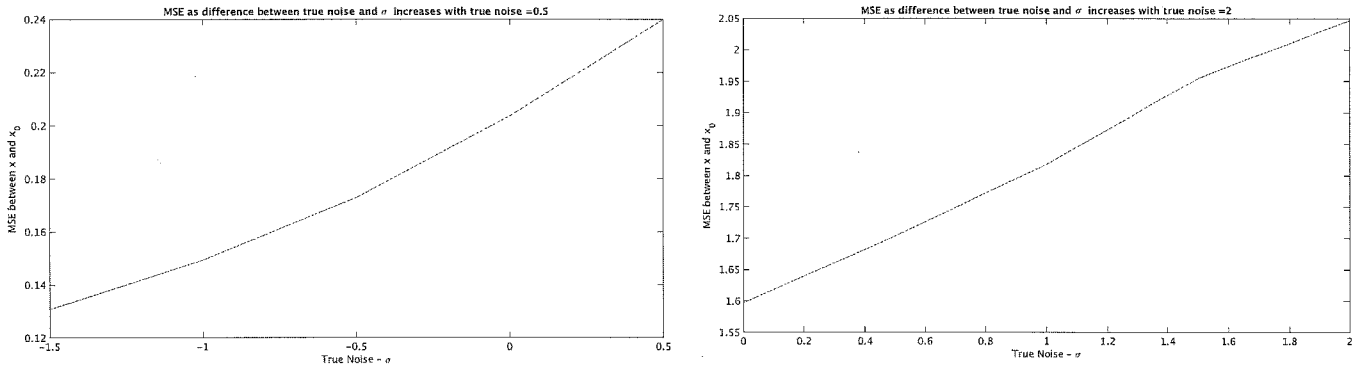
After reading the first few pages of this paper I was happy to see the authors gave a succinct overview of what they set out to do. But as I continued to read I became more confused as I was unsure how there overview translated into the multiple sections of the paper. In particular I was confused about the progression of subproblems and what task solves what (i.e. to solve BPDN we must solve problem X, to solve problem X we must solve problem Y, etc.). Fortunately I was able to clarify this progression with others in the class. I decided to look at the effects of our σ choice on the results. Using the basic example given on the code webpage, I tried increasing σ to look at the result. As we increase σ the BPDN has less recovered coefficients deviating from 0, even when they originally do. This can be seen in figure 1 below. If the algorithm thinks there is a lot of noise it will ignore deviations from overall pattern in because it assumes they are noise and not really a part of the solution. That is why I think there are more nonzero coefficients in $\sigma = 0.205$ than $\sigma = 1.805$. A problem with that though is that it will not be able to recover some nonzero coefficients at all.



(a) $\sigma = 0.205$ (b) $\sigma = 1.805$

Figure 1: Comparison of Ground Truth vs Algorithm Solution for BPDN with different values of σ

While I was able to see the effect of an increasing σ, I wanted to see if there was an effect if we increase the difference between the "true noise" (i.e. the random vector added to Ax_0 to get b in the example) and σ. I compared the solution recovered with the algorithm with the ground truth with MSE and saw how it changed as we increased the aforementioned difference. I tried this for different choices of the true noise, but regardless of choice, the MSE increased as the difference increased. Interestingly, the MSE was not as its lowest when the difference was zero, but seemed to decrease as the difference decreased. This suggests that if we overestimate σ, we have a better result than if we underestimate (though the change in MSE is less than 1, so its arguable if the improvement is significant). As described before, I think this is because if we set our noise estimate to be relatively large, the algorithm will not try as hard to approximate the nonzero coefficients of the model in its solution. These approximations of nonzero coefficients may result in larger error than if they weren't accounted for at all. So I think we are able to reduce the MSE by playing it safe and not trying to approximate every detail of the x_0. Also, we see overall that the MSE is larger in figure 2(b) than 2(a), which makes sense since with more noise it would be harder to approximate x_0.



(a) True Noise = 0.5 (b) True Noise = 2

Figure 2: MSE between x and x_0 as difference between true noise and σ increases

Reading Report of: *Probing the Pareto Frontier for Basis Pursuit Solutions* by van den Berg et al.

(ECE 695, Reading Assignment 01, Spring 2018)

Overview of the Paper

This paper is about the efficient solution of the basis pursuit denoising problem (BP_σ)

$$\min_x ||x||_1 \qquad \text{subject to} \qquad ||Ax - b||_2 \leq \sigma \tag{1}$$

It can be shown that this problem is equivalent to the ℓ_1-regularized least-squares problem (QP_λ) when choosing the correct regularization parameter, λ. As the problems are in theory in that sense equivalent, in practice the corresponding $\lambda = \lambda_\sigma$ for an chosen σ parameter is in general unknown beforehand.

In many applications the solution of the (BP_σ) problem can be advantageous in comparison to the (QP_λ) problem. That is because the user might not know the degree of sparsity of the unknown signal, x, and thus is unable to choose an appropriate λ or an upper bound for the ℓ-1 norm of x. In contrast one may have an idea of the underlying noise of the data term which determines a reasonable choice for the parameter, σ.

The work in this paper aims to

Prior Work

Several algorithms for the problem at hand have been know. The paper mentions prior work that solves the (QP_λ) problem successively stating with $\lambda = ||A^\top b||_\infty$ (which results in $x = \mathbf{0}$) and exhaustively narrowing down the correct value of λ to λ_σ. Since each of the solutions of the (QP_λ) problem is an iterative process the overall process of finding λ_σ is doubly iterative and thus in practice slow.

Furthermore, some of the existing software required the explicit formulation of the A-matrix, thus excluding applications where the A-matrix operation exhibits fast implementations via FFT or wavelet transform. Also, approaches have been brought forward that solve (QP_λ) for uniform λ's which however result in uneven sampling of the Pareto frontier.

Key Ideas of the Paper

The method in this paper makes use of the function

$$\phi(\tau) = ||Ax_\tau - b||_2 \tag{2}$$

where x_τ solves the least squares problem subject to $||x||_1 \leq \tau$. It is shown that the function ϕ is non-increasing, convex and continuously differentiable for all points of interest. It follows that $\phi(\tau) = \sigma$ if and only if x_τ solves (BP_σ). Thus, the solution of (BP_σ) reduces to solution to the single variable equation $\phi(\tau) = \sigma$. The function ϕ having said properties the roots of that equation are found using the Newton method.

In this paper the authors present a method in which the values of $\phi(\tau)$ and $\phi'(\tau)$ are only known approximately while still ensuring convergence to the overall problem.

Comments

The premise of the paper is that the (BP_σ), parametrized by a meaningful σ, is often more practical than the (QP_λ) problem that is parametrized by a less intuitive regularization parameter, λ. For my experiments I wanted to find out how in a controlled, noisy environment the σ parameter has to be chosen to result in the best prediction of the unknown signal, x.

I used the following parameters for my experiments:

$$b = Ax_0 + \xi \qquad \text{where} \qquad \xi \sim \mathcal{N}(0, I_n \sigma^2_{\text{noise}}) \tag{3}$$

where A is a $m \times n = 100 \times 100$ unitary matrix, x_0 is a sparse vector with 5 i.i.d. standard Gaussian nonzero entries and $\sigma_{\text{noise}} = 0.01$.

For the first experiment I ran the SPG code for *one particular instance* of the random structures A, x_0 and ξ while varying the σ parameter over a wide range. I recorded the mean squared error and the ℓ_1 and ℓ_2 norm for every σ and plotted the results in Figure 1.

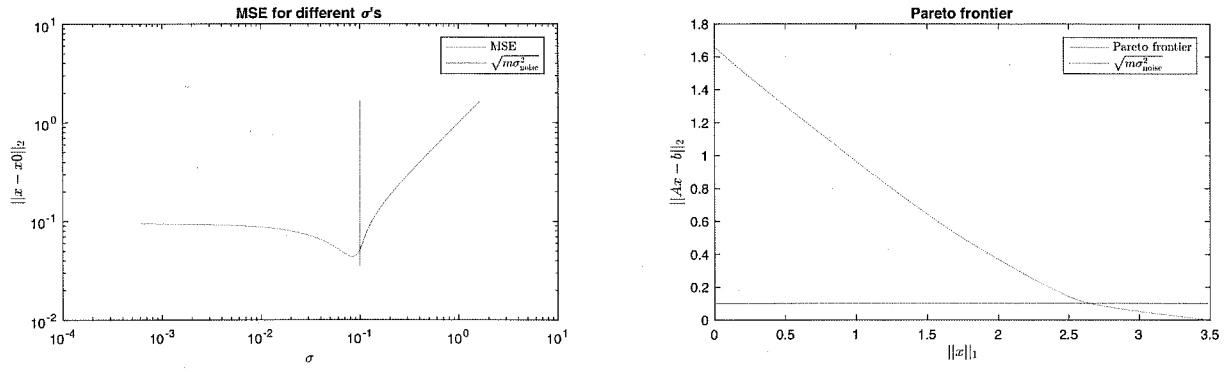


Figure 1: Left: Mean squared error of the solution of the (BP_σ) problem for varying values of the parameter σ also the noise standard variation σ_{noise} is indicated in red. Right: Pareto frontier derived from the same data, while indicating the same noise standard variation σ_{noise}.

Also, I reasoned that the value $\sigma = \mathbb{E}\left[||\xi||_2\right]$ would be a good estimate for the σ that minimizes the MSE:

$$\sigma_{\min} = \arg\min_{\sigma} \left\{ \min_{x} ||x||_1 \qquad \text{subject to} \qquad ||Ax - b||_2 \leq \sigma \right\} \tag{4}$$

Also, $\mathbb{E}\left[||\xi||_2\right] \approx \sqrt{m\sigma^2_{\text{noise}}} = 0.1$ seemed like a reasonable approximation of that value. I plotted $\sqrt{m\sigma^2_{\text{noise}}}$ as a red line both in the MSE plot and the Pareto frontier. In most cases the estimate for the best σ seemed to be close to the actual $\sigma_{\min}$ but usually too large. Thus, for my second experiment I wanted to explore the true distribution of the best upper bound parameter $\sigma_{\min}$.

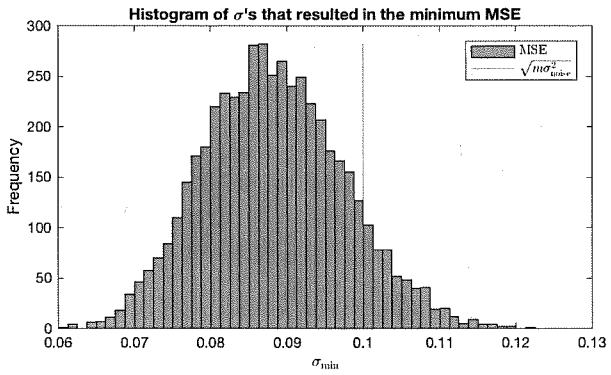


Figure 2: Histogram of the best σ for many runs of the first experiment with different random

In this experiment I reused the same choice for A, x_0 and ξ, while empirically finding the minimizer $\sigma_{\min}$. I reran this 5000 times with *different instances* of these random objects each yielding a different MSE minimizer. The histogram of all $\sigma_{\min}$'s is shown in Figure 2. Disregarding that $\sigma_{\min} \approx \sqrt{m\sigma^2_{\text{noise}}}$ is only a rough estimate, I observe that the distribution of the correct $\sigma_{\min}$ is relatively narrow and behaved. This confirms the premise that the (BP_σ) problem gives good, predictable estimates of the unknown, x when the noise distribution is known.