Dynamic Updating for Sparse Time Varying Signals

M. Salman Asif and Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia, 30332, USA
Email: {sasif, jrom}@ece.gatech.edu

***Abstract*—Many signal processing applications revolve around finding a sparse solution to a (often underdetermined) system of linear equations. Recent results in compressive sensing (CS) have shown that when the signal we are trying to acquire is *sparse* and the measurements are *incoherent*, the signal can be reconstructed reliably from an incomplete set of measurements. However, the signal recovery is an involved process, usually requiring the solution of an ℓ_1 minimization program.**

In this paper we discuss the problem of estimating a time-varying sparse signal from a *series* of linear measurements. We propose an efficient way to *dynamically* update the solution to two types of ℓ_1 problems when the underlying signal changes. The proposed dynamic update scheme is based on homotopy continuation, which systematically breaks down the solution update into a small number of linear steps. The computational cost for each step is just a few matrix-vector multiplications.

I. INTRODUCTION

This paper is concerned with solving linear systems of equations using ℓ_1 regularization. In recent years, several methods have been proposed to perform this task reliably. In particular, the results in compressive sensing (CS) [1]–[3] have given some important theoretical limits and conditions under which the signal recovery is possible from under-determined system of equations. In the framework of compressive sensing, we observe a sparse signal indirectly by making a small number of linear *incoherent* measurements. The general setup is as follows: We are given measurements $y \in \mathbb{R}^m$ of the form

$$y = Ax + e, \tag{1}$$

where $x \in \mathbb{R}^n$ is the unknown signal, A is an $m \times n$ measurement matrix with $m \ll n$, and $e \in \mathbb{R}^m$ represents the error in the measurements. To reconstruct the original signal, we solve an optimization program that involves minimizing the ℓ_1 norm of the sparse signal under some *data fidelity* constraints.

In this paper we will consider two such optimization methods. The first one goes by the name of LASSO [4] or basis pursuit denoising (BPDN) [5], which solves the following optimization problem

$$\text{minimize} \;\; \frac{1}{2}\|A\tilde{x} - y\|_2^2 + \tau\|\tilde{x}\|_1 \tag{2}$$

for some suitable choice of $\tau > 0$. The second one is the Dantzig selector (DS) [6], which solves (for $\tau > 0$)

$$\text{minimize} \;\; \|\tilde{x}\|_1 \quad \text{subject to} \quad \|A^T(A\tilde{x} - y)\|_\infty \leq \tau. \tag{3}$$

This research was supported in part by an ONR Young Investigator Award and DARPA's Analog to Information program.

In this paper we consider the problem of estimating a time-varying sparse signal from a *series* of linear measurements. This type of problem can arise in many situations where we want to estimate some closely related sparse signals from linear measurements. For example, in real time MRI we want to reconstruct a series of closely related frames [7]. In communication systems, we are continuously trying to estimate a time-varying (often sparse) channel impulse response; particular applications include high-definition television (HDTV), broadband communication, and underwater acoustic communication [8]. Other applications might include video surveillance or object tracking where we want to update the estimate at regular intervals.

The problem description is as follows. Assume that we have solved (2) for the system in (1). Now say that the underlying signal x changes to $\breve{x}$ and we get a new set of m measurements given as

$$\breve{y} = A\breve{x} + \breve{e}. \tag{4}$$

This gives rise to the following updated optimization problem

$$\text{minimize} \;\; \frac{1}{2}\|A\tilde{x} - \breve{y}\|_2^2 + \tau\|\tilde{x}\|_1. \tag{5}$$

Since we expect that the signal changes only slightly between measurements, and so the reconstructions will be closely related. Our goal here is to avoid solving this new optimization problem from scratch, instead using information from the solution of (2) to quickly find the solution for (5). Our approach for doing this is based on the *homotopy continuation principle*.

Homotopy methods provide a general framework in which we can continuously transform one optimization problem to another by introducing a *homotopy parameter*. As this parameter is varied, we trace the *homotopy path* between the two problems. In our setting for *dynamic updating* of a solution from a new set of measurements, we first build a homotopy transformation between the already solved problem in (2) and the desired problem in (5). Then we solve a series of simple problems along the homotopy path, which lead us to the solution of (5). It is an iterative procedure, where every iteration involves a rank-1 update and 2 matrix-vector multiplications, so the cost for each homotopy step is $O(mn)$.

The outline of this paper is as follows: In Section II we give a brief review of the homotopy methods to solve LASSO and DS. In Section III and IV we discuss the solution update scheme for LASSO and DS. In Section V we present some simulation results for our solution update algorithm for LASSO and its comparison with some state-of-the-art algorithms with a "warm start".

II. Homotopy methods

In this section we will briefly review the homotopy algorithms to solve LASSO and DS for a given value of τ. We will also discuss some of their properties which we will use to derive the homotopy algorithms for the dynamic update.

A. LASSO homotopy

The homotopy algorithm for LASSO (sometimes also called LARS) has been extensively studied in the literature of statistics and signal processing [9]–[11]. The basic premise of LASSO homotopy is that the solution for (2) follows a piecewise linear path with respect to τ. The homotopy algorithm for LASSO traces the solution path to (2) by starting from a very large value of τ (where the solution will be zero) and reducing it towards the desired value. The homotopy path is followed by ensuring that certain optimality conditions are being maintained. To be a solution to (2), a vector x^* must obey the following conditions (constraints) [12]

$$\|A^T(Ax^* - y)\|_\infty \le \tau. \tag{L}$$

In addition, a sufficient condition for the optimality of x^* is that the set of locations for which the constraints in (L) are active (i.e., equal to τ) will be the same as the set of locations for which x^* is non-zero [13]. This gives us the following set of optimality conditions which must be obeyed by the solution x^* to (2) for any given value of τ

L1. $A_\Gamma^T(Ax^* - y) = -\tau z$
L2. $\|A_{\Gamma^c}^T(Ax^* - y)\|_\infty < \tau$,

where Γ is the support (index set for non-zero elements) of x^*, A_Γ is the $m \times |\Gamma|$ matrix formed from the columns of A indexed by elements in the set Γ, and z is a $|\Gamma|$-vector containing the signs of x^* on Γ. From this we can see that the solution x^* of (2) can be written as

$$x^* = \begin{cases} (A_\Gamma^T A_\Gamma)^{-1}(A_\Gamma^T y - \tau z) & \text{on } \Gamma \\ 0 & \text{otherwise} \end{cases},$$

which is piecewise linear w.r.t. τ. As we reduce τ, the solution moves along the direction ∂x, given as

$$\partial x = \begin{cases} (A_\Gamma^T A_\Gamma)^{-1} z & \text{on } \Gamma \\ 0 & \text{otherwise} \end{cases},$$

until one of the two things happens: an element of x^* shrinks to zero (removing an element from the support of x^*), or another constraint in (L) becomes active (indicating addition of an element in the support of x^*). These changes happen at certain *critical points* along the homotopy path corresponding to some critical values of τ, giving the problem its piecewise linear structure. In the LASSO homotopy we trace the homotopy path for solution of (2) as follows: We start from a large value of τ, at every critical point we update the solution, its support and sign sequence, we find an update direction using the active constraints and move to the next critical point. Repeat this procedure until τ has reduced to the desired value.

B. Dantzig selector homotopy

The homotopy algorithm for DS is similar in principle to the homotopy for LASSO. The essential difference between the two is that for DS homotopy we have to keep track of both primal and dual solutions for (3) [14], [15]. The dual problem to the DS in (3) can be written as

$$\text{maximize } -(\tau\|\lambda\|_1 + \langle \lambda, A^T y\rangle) \text{ subject to } \|A^T A\lambda\|_\infty \le 1, \tag{6}$$

where $\lambda \in \mathbb{R}^n$ is the dual optimization variable. We can derive the required optimality conditions by invoking the complementary slackness property and strong duality between the problems in (3) and (6) [16]. So any primal-dual solution pair (x^*, λ^*) to (3) and (6) for any given value of τ must satisfy the following conditions:

D1. $A_{\Gamma_\lambda}^T(Ax^* - y) = \tau z_\lambda$
D2. $A_{\Gamma_x}^T A\lambda^* = -z_x$
D3. $\|A_{\Gamma_\lambda^c}^T(Ax^* - y)\|_\infty < \tau$
D4. $\|A_{\Gamma_x^c}^T A\lambda^*\|_\infty < 1$,

where Γ_x and Γ_λ are the supports of x^* and λ^* respectively, z_x and z_λ are the sign sequences of x^* and λ^* on their respective supports. We will call (D1,D3) the *primal constraints*, and (D2,D4) the *dual constraints*.

It is convenient to divide every homotopy step for DS into two phases: 1) *primal update* and 2) *dual update*.

Primal update: We can see from D1 that the primal solution x^* to DS is also piecewise linear with respect to τ. As we reduce τ, the solution x^* moves in the direction ∂x given as

$$\partial x = \begin{cases} -(A_{\Gamma_\lambda}^T A_{\Gamma_x})^{-1} z_\lambda & \text{on } \Gamma_x \\ 0 & \text{otherwise} \end{cases}.$$

The primal update phase for DS homotopy is very similar to the homotopy step for the LASSO. We move our solution in the direction ∂x by reducing τ and at some point either an element in x^* will shrink to zero or a primal constraint in D3 will become active (indicating addition of a new element in the support of λ^*). This gives us a critical point on the homotopy path and a new critical value of τ. However, we do not know the value of the dual solution λ^* at this new critical value of τ.

Dual update: In the dual update, we use the information about the change in the support from primal phase to find an update direction $\partial\lambda$ for the dual vector, and consequently the new value for the dual solution λ^*. Assume that during primal update, a new element entered the support of λ^* at index γ. Then using D2 we can write the update direction $\partial\lambda$ as

$$\partial\lambda = \begin{cases} -z_\gamma (A_{\Gamma_x}^T A_{\Gamma_\lambda})^{-1} A_{\Gamma_x}^T a_\gamma & \text{on } \Gamma_\lambda \\ z_\gamma & \text{on } \gamma \\ 0 & \text{elsewhere} \end{cases}, \tag{7}$$

where a_γ is the γth column of A, z_γ is the sign of γth primal active constraint, which in fact is the sign of the new element in the new dual solution. This direction ensures that the dual constraints remain active on Γ_x and the new element in λ^* at index γ has the proper sign z_γ. So as we move our solution in the direction $\partial\lambda$, one of the two things will happen: either an element in λ^* will shrink to zero or a dual constraint in D4 will become active (indicating an addition of a new element in the support of x^*). The point where this happens gives us the new value of λ^*.

If instead an element of x^* shrinks to zero during primal update, then we can pick an "artificial" index $\gamma \in \Gamma_\lambda$ and treat it as if it were a new element in the support of λ^*. However, in this case there will be ambiguity about sign of update direction, since z_γ does not mean anything here. But we can verify the proper direction by looking at the dual constraints corresponding to the outgoing element of x^* [15].

So in every homotopy step for DS, we first update the primal vector and primal constraints, and then using the information about the change of support from primal update phase, we update the dual vector and corresponding constraints. In this way we move from one critical point to the next along the homotopy path, until τ has been reduced to its final value.

We can also use this piecewise linear structure to derive algorithms to *dynamically* update the solutions to both problems when the measurements change from y to $\breve{y}$.

III. LASSO DYNAMIC UPDATE

Let us now look at the dynamic update of solution for the LASSO. Assume that we have solved (2) for some given value of τ. Then we get a new set of measurements $\breve{y}$ as given in (4) and we want to solve (5) for the *same* value of τ.

We will develop the algorithm for updating the solution to (5) following three steps. First, we construct the homotopy transformation between (2) and (5). Second, we derive the optimality conditions that the solution must obey for any given value of the homotopy parameter. Finally, using these optimality conditions we trace the homotopy path towards the new solution.

Our proposed homotopy formulation is as follows:

$$\underset{x}{\text{minimize}} \quad \frac{1}{2}\|Ax - (1-\epsilon)y - \epsilon\breve{y}\|_2^2 + \tau\|x\|_1, \tag{8}$$

where ϵ is the homotopy parameter. As we increase ϵ from 0 to 1, the solution to (8) traces a homotopy path from the solution of (2) to the solution of (5). The optimality conditions for any solution x^* to (8) at a given value of ϵ can be written as:

$$\|A^T(Ax^* - (1-\epsilon)y - \epsilon\breve{y})\|_\infty \le \tau, \tag{9}$$

or more precisely,

$$A_\Gamma^T(Ax^* - (1-\epsilon)y - \epsilon\breve{y}) = -\tau z \tag{9a}$$

$$\|A_{\Gamma^c}^T(Ax^* - (1-\epsilon)y - \epsilon\breve{y})\|_\infty < \tau, , \tag{9b}$$

where Γ is the support of x^* and z is its sign sequence on Γ. We can see from (9a) that again the solution to (8) follows a piecewise linear path as ϵ varies; the critical points in this path occur when an element is either added or removed from the solution x^*.

Suppose that we are at a solution x_k (with support Γ and signs z) to (8) at some critical value of $\epsilon = \epsilon_k$. To find the direction to move, we will examine how the optimality conditions behave as ϵ increases by an infinitesimal amount from ϵ_k to ϵ_k^+. The solution x_k^+ at $\epsilon = \epsilon_k^+$ must obey

$$A_\Gamma^T(Ax_k^+ - (1-\epsilon_k^+)y - \epsilon_k^+\breve{y}) = -\tau z. \tag{11}$$

Subtracting (9a) from (11), the difference between the solutions $\widetilde{\partial x} = x_k^+ - x_k$ will be

$$\widetilde{\partial x} = \begin{cases} \Delta\epsilon \cdot (A_\Gamma^T A_\Gamma)^{-1} A_\Gamma^T(\breve{y} - y) & \text{on } \Gamma \\ 0 & \text{otherwise,} \end{cases}$$

where $\Delta\epsilon = \epsilon_k^+ - \epsilon_k$. So as ϵ increases from ϵ_k, the direction the solution moves is given by

$$\partial x = \begin{cases} (A_\Gamma^T A_\Gamma)^{-1} A_\Gamma^T(\breve{y} - y) & \text{on } \Gamma \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

With the update direction given by (12), we need to find the step-size θ that will take us to the next critical value of ϵ. We increase ϵ from ϵ_k, moving the solution away from x_k in the direction ∂x, until one of the two things happens: one of the entries in the solution shrinks to zero or one of the constraints in (9b) becomes active (equal to τ). The smallest amount we can move ϵ so that the former is true is simply

$$\theta^- = \min_{i\in\Gamma}\left(\frac{-x_k(i)}{\partial x(i)}\right)_+, \tag{13}$$

where $\min(\cdot)_+$ denotes that the minimum is taken over positive arguments only. For the latter, set

$$p_k = A^T(Ax_k - y + \epsilon_k(y - \breve{y})) \tag{14a}$$

$$d_k = A^T(A\partial x + y - \breve{y}). \tag{14b}$$

We are now looking for the smallest stepsize $\Delta\epsilon$ so that $p_k(i) + \Delta\epsilon \cdot d_k(i) = \pm\tau$ for some $i \in \Gamma^c$. This is given by

$$\theta^+ = \min_{i\in\Gamma^c}\left(\frac{\tau - p_k(i)}{d_k(i)}, \frac{\tau + p_k(i)}{-d_k(i)}\right)_+. \tag{15}$$

So the stepsize to the next critical point is

$$\theta = \min(\theta^+, \theta^-). \tag{16}$$

With the direction ∂x and stepsize θ chosen, the next critical value of ϵ and new solution at that point will be

$$\epsilon_{k+1} = \epsilon_k + \theta, \qquad x_{k+1} = x_k + \theta\partial x.$$

The support for new solution x_{k+1} differs from Γ by one element. Let i^- and i^+ be the indices for the minimizers in (13) and (15) respectively. So either we remove i^- from the support Γ (if $\theta = \theta^-$) or we add i^+ to the support (if $\theta == \theta^+$), and update the sign vector accordingly.

This procedure is repeated until $\epsilon = 1$. A precise outline of the algorithm is given in Algorithm 1.

The main computational cost of this algorithm comes from finding the update direction ∂x and d_k for stepsize. For update direction, since the support changes by a single element

Algorithm 1 Dynamic solution update for the LASSO

Start with $\epsilon_0 = 0$ at solution x_0 to (2) with support Γ and sign sequence z on the Γ for $k = 0$.
repeat
  compute ∂x as in (12)
  compute p_k, d_k as in (14) and θ as in (16)
  $x_{k+1} = x_k + \theta \partial x$
  $\epsilon_{k+1} = \epsilon_k + \theta$
  if $\epsilon_{k+1} \geq 1$ **then**
    $\theta = 1 - \epsilon_k, \quad x_{k+1} = x_k + \theta \partial x$
    `break;` {Quit without any further update}
  end if
  if $\theta = \theta^-$ **then**
    $\Gamma \leftarrow \Gamma \setminus \{i^-\}$
    update z
  else
    $\Gamma \leftarrow \Gamma \cup \{i^+\}$
    $z(i^+) = \text{sign}\ [p_k(i^+) + \theta d_k(i^+)]$
  end if
  $k \leftarrow k + 1$
until `stopping criterion is satisfied`

from step to step, $(A_\Gamma^T A_\Gamma)^{-1}$ can be computed by a rank-1 update using matrix inversion lemma [17]. Computing step size involves two matrix-vector multiplications. As such, the computational cost of each step is $O(mn)$.

IV. Dantzig selector dynamic update

The homotopy algorithm for dynamic update of the Dantzig selector is very similar to the update of LASSO discussed in Section III, with the additional requirement of updating both primal and dual solutions at every step. Assume we have already solved (3) for some given value of τ and data y. Then we receive a new set of measurements $\breve{y}$ as described in (4). We are interested in solving the following updated optimization problem

$$\text{minimize } \|\tilde{x}\|_1 \ \text{ subject to } \ \|A^T(A\tilde{x} - \breve{y})\|_\infty \leq \tau, \tag{17}$$

for which we can write the homotopy formulation as

$$\text{minimize } \ \|x\|_1 \quad \text{subject to} \quad \|A^T(Ax - (1-\epsilon)y - \epsilon\breve{y})\|_\infty \leq \tau. \tag{18}$$

By adapting the optimality conditions D1-D4, we can write the optimality conditions for any primal-dual solution pair (x^*, λ^*) to (18) at a given value of ϵ as

$$A_{\Gamma_\lambda}^T (Ax^* - (1-\epsilon)y - \epsilon\breve{y}) = \tau z_\lambda, \tag{19a}$$

$$A_{\Gamma_x}^T A\lambda^* = -z_x \tag{19b}$$

$$\|A_{\Gamma_\lambda^c}^T (Ax^* - (1-\epsilon)y - \epsilon\breve{y})\|_\infty < \tau, \tag{19c}$$

$$\|A_{\Gamma_x^c}^T A\lambda^*\|_\infty < 1, \tag{19d}$$

where Γ_x and Γ_λ are supports of x^* and λ^* respectively, and z_x and z_λ are the sign sequences on their respective supports. We can see from (19a) that again solution x^* to (18) follows a piecewise linear path w.r.t. ϵ, and there will be some critical points (corresponding to the critical values of ϵ) where the support of x^* and/or λ^* changes.

The homotopy steps for dynamic update of DS also consist of two parts: 1) primal update and 2) dual update.

Primal update: Suppose we are at some critical value of $\epsilon = \epsilon_k$, with primal-dual solution (x_k, λ_k). As we change ϵ from ϵ_k to ϵ_k^+, the solution changes to $x_k^+ = x_k + \Delta\epsilon \partial x$, where ∂x is given as

$$\partial x = \begin{cases} (A_{\Gamma_\lambda}^T A_{\Gamma_x})^{-1} A_{\Gamma_\lambda}^T (\breve{y} - y) & \text{on } \Gamma_x \\ 0 & \text{otherwise} \end{cases}, \tag{20}$$

and $\Delta\epsilon = \epsilon_k^+ - \epsilon_k$. Now if we start to move the solution in the direction ∂x by increasing ϵ from ϵ_k, at some point either a primal constraint will be activated in (19c) (indicating a new element in Γ_λ) or an element in x_k will shrink to zero. We select the smallest step size such that one of these two things happens. The smallest step size we can take such that an entry in x_k shrinks to zero is

$$\theta^- = \min_{i \in \Gamma_x} \left(\frac{-x_k(i)}{\partial x(i)} \right)_+. \tag{21}$$

The smallest step size such that a constraint in (19c) becomes active is given by

$$\theta^+ = \min_{i \in \Gamma_\lambda^c} \left(\frac{\tau - p_k(i)}{d_k(i)}, \frac{\tau + p_k(i)}{-d_k(i)} \right)_+, \tag{22}$$

where p_k and d_k are defined in the same way as in (14). So the stepsize to the next critical point is

$$\theta = \min(\theta^+, \theta^-). \tag{23}$$

The next critical value of ϵ and solution at that point will be

$$\epsilon_{k+1} = \epsilon_k + \theta, \qquad x_{k+1} = x_k + \theta\partial x.$$

Let us denote i^- and i^+ as the indices for the minimizers in (21) and (22) respectively. So either we remove i^- from the primal support Γ_x (if $\theta = \theta^-$) or we add i^+ to the dual support Γ_λ (if $\theta = \theta^+$), and update the sign sequences accordingly. As we mentioned earlier in the case of the standard DS homotopy in Section II-B, we will not have the dual solution at this new critical point.

Dual update: In the dual update we use information about the change of support in the primal update phase to find an update direction $\partial\lambda$ for the dual vector and consequently the dual solution λ_{k+1} at $\epsilon = \epsilon_{k+1}$. Assume that during the primal update, a new element entered[1] the support of λ at index γ. Then using (19b) we get the update direction $\partial\lambda$, which is the same as the one defined in (7). This direction ensures that the dual constraints remain active on Γ_x and the sign of new non-zero element in $\lambda_k^+ = \lambda_k + \delta_k \partial\lambda$ at index γ is z_γ. So as we move our solution λ_k in this direction $\partial\lambda$ by increasing the step size δ_k from 0, one of two things will happen, either a nonzero element from λ_k will shrink to zero or a dual constraint in (19d) will become active (indicating

[1]If instead an element was removed from x_k, we can use the same trick of considering an artificial new element, as discussed in Section II-B.

addition of a new element in Γ_x). The smallest step size such that an entry in λ_k shrinks to zero is simply

$$\delta^- = \min_{j \in \Gamma_\lambda} \left(\frac{-\lambda_k(j)}{\partial\lambda(j)} \right)_+ . \quad (24)$$

The smallest step size such that a constraint in (19d) becomes active is given by

$$\delta^+ = \min_{j \in \Gamma_x^c} \left(\frac{1 - a_k(j)}{b_k(j)}, \frac{1 + a_k(j)}{-b_k(j)} \right)_+ , \quad (25)$$

where

$$a_k = A^T A \lambda_k, \qquad b_k = A^T A \partial\lambda. \quad (26)$$

The stepsize for the update of dual solution is

$$\delta = \min(\delta^+, \delta^-). \quad (27)$$

The new dual solution will be $\lambda_{k+1} = \lambda_k + \delta\partial\lambda$. Let us define j^- and j^+ as the indices corresponding to minimizers of (24) and (25) respectively. So either j^+ is added to Γ_x (if $\delta = \delta^+$) or j^- is removed from Γ_λ (if $\delta = \delta^-$).

This procedure of primal and dual update is repeated until $\epsilon = 1$. A precise outline of the algorithm is given in Algorithm 2.

V. Numerical experiments

In our first simulation, we start with a sparse signal x which contains ± 1 spikes at randomly chosen K locations. The measurement vector y is generated as in (1), with e as a Gaussian noise whose entries are distributed $\text{Normal}(0, 0.01^2)$. We used $m \times n$ Gaussian matrix as our measurement matrix A with all entries independently distributed $\text{Normal}(0, 1/m)$. We solve (2) for a given value of τ. Then we modify the sparse signal x to get $\breve{x}$ as follows. First, we perturb the non-zero entries of x by adding random numbers distributed $\text{Normal}(0, 0.1^2)$. Then K_n new entries are added to x, with the locations chosen uniformly at random, and the values distributed $\text{Normal}(0, 1)$. New measurements $\breve{y} := A\breve{x} + \breve{e}$ are generated, with another realization of the noise vector $\breve{e}$, and (5) is solved using the DynamicX algorithm (Algorithm 1).

The results of 500 simulations with $n = 1024$, $m = 512$, $K = m/5$ are summarized Table I. In each simulation, K_n was selected uniformly from $[0, K/20]$. Several values of τ were tested, $\tau = \lambda\|A^T y\|_\infty$ with $\lambda \in \{0.5,\ 0.1,\ 0.05,\ 0.01\}$. The experiments were run on a standard desktop PC, and two numbers were recorded: the average number of times we needed to apply[2] A^T and A (nProdAtA), and the average CPU time needed to complete the experiment (CPU).

Table I also compares DynamicX to three other methods. The first is "Standard LASSO homotopy", which resolves (5) from scratch using our own implementation of the homotopy algorithm (starting with large τ and gradually reducing it to its desired value). The second is the GPSR-BB algorithm [18], which is "warm started" by using the previously recovered

[2]Each iteration of the DynamicX algorithm requires an application of $A^T A$ along with several much smaller matrix-vector multiplies to perform the rank one update.

Algorithm 2 Dynamic solution update for the Dantzig selector

Start at $\epsilon_0 = 0$ with primal-dual solution (x_0, λ_0), their support sets and sign sequences $(\Gamma_x, \Gamma_\lambda, z_x, z_\lambda)$.
repeat
  Primal update:
  compute ∂x as in (20) and θ as in (23)
  $x_{k+1} = x_k + \theta\partial x$
  $\epsilon_{k+1} = \epsilon_k + \theta$
  if $\epsilon_{k+1} \geq 1$ **then**
    $\theta = 1 - \epsilon_k, \quad x_{k+1} = x_k + \theta\partial x.$
    `break;`   {Quit without any further update}
  end if
  if $\theta = \theta^-$ **then**
    $\Gamma_x \leftarrow \Gamma_x \backslash \{i^-\}$
    $\tilde{\Gamma}_\lambda = \Gamma_\lambda$
    $\Gamma_\lambda \leftarrow \Gamma_\lambda \backslash \{\gamma\}$ {an artificial new element γ}
    update z_x, z_λ, z_γ
  else
    $\tilde{\Gamma}_\lambda = \Gamma_\lambda \cup \{i^+\}$, set $\gamma = i^+$
    update z_λ, z_γ
  end if
  Dual update:
  compute $\partial\lambda$ as in (7), a_k and b_k as in (26)
  if $\theta = \theta^-$ & $\text{sign}[a_k(i^-)] = \text{sign}[b_k(i^-)]$ **then**
    $\partial\lambda \leftarrow -\partial\lambda, \quad b_k \leftarrow -b_k$
  end if
  compute δ as in (27)
  $\lambda_{k+1} = \lambda_k + \delta\partial\lambda$
  if $\delta = \delta^-$ **then**
    $\Gamma_\lambda \leftarrow \tilde{\Gamma}_\lambda \setminus j^-$
    update z_λ
  else
    $\Gamma_x \leftarrow \Gamma_x \cup \{j^+\}$
    $\Gamma_\lambda \leftarrow \tilde{\Gamma}_\lambda$
    $z_x = \text{sign}[A_{\Gamma_x}^T A \lambda_{k+1}]$ {update z_x}
  end if
  $k \leftarrow k + 1$
until `stopping criterion is satisfied`

signal as the starting point. The third algorithm is FPC_AS [3] [19], which is also warm started. The accuracy in GPSR and FPC was chosen so that the relative error between the exact solution and their solution was 10^{-6}. We see that DynamicX compares favorably across a large range of τ.

Table I also contains results for the two experiments below.

Blocks: In this experiment, we recover a series of 200 piecewise constant signals of length $n = 2048$, similar to the *Blocks* signal from WaveLab [20]. We use the Haar wavelet transform to represent the signal, and take $m = 1024$ measurements. Each signal is a slight variation of the last: the discontinuities stay fixed, while the levels of the constant regions are perturbed by multiplying by a random number

[3]The FPC actually uses a different number of applications of A and A^T; these were averaged for the numbers in Table I.

TABLE I

COMPARISON OF THE DYNAMIC UPDATE OF TIME-VARYING SPARSE SIGNALS USING THE STANDARD LASSO HOMOTOPY, GPSR AND FPC. RESULTS ARE GIVEN IN TERMS OF THE NUMBER OF PRODUCTS WITH A^T AND A, AND CPU TIME.

Signal type	λ ($\tau = \lambda \|A^T y\|_\infty$)	DynamicX (nProdAtA, CPU)	Standard Homotopy (nProdAtA, CPU)	GPSR-BB (nProdAtA, CPU)	FPC_AS (nProdAtA, CPU)
$n = 1024$,	0.5	(11.84, 0.031)	(42.05, 0.10)	(15.34, 0.03)	(31.29, 0.055)
$m = 512$,	0.1	(12.9, 0.055)	(154.5, 0.491)	(54.45, 0.095)	(103.38, 0.13)
$K = m/5$,	0.05	(14.56, 0.062)	(162, 0.517)	(58.17, 0.10)	(102.37, 0.14)
values: ± 1 spikes	0.01	(23.72, 0.132)	(235, 0.924)	(104.5, 0.18)	(148.65, 0.177)
Blocks	0.01	(2.7,0.028)	(76.8,0.490)	(17,0.133)	(53.5,0.196)
Pcw. Poly.	0.01	(13.83,0.151)	(150.2,1.096)	(26.05, 0.212)	(66.89, 0.250)

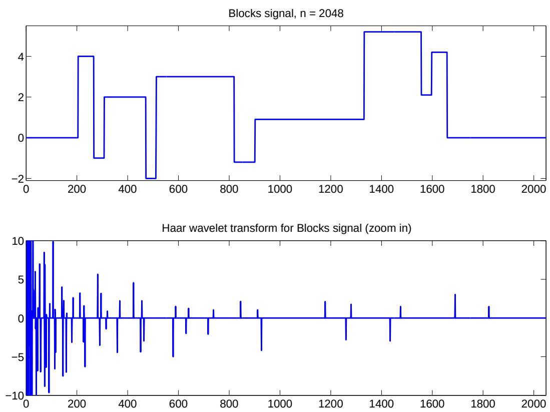

Fig. 1. An example of Piecewise constant signal, sparse in wavelet domain

uniformly distributed between 0.8 and 1.2. A typical signal and its wavelet transform are shown in Figure 1.

Piecewise polynomial: This experiment is similar to the Blocks experiment, except that we use a piecewise polynomial (cubic) signal and represent it using the Daubechies 8 wavelet transform. The polynomial functions are perturbed from signal to signal by adding small Gaussian random variables to the polynomial coefficients.

VI. CONCLUSIONS

We have presented a homotopy based scheme to dynamically update the solution for the LASSO and the Dantzig selector when the signal we are observing changes slightly. There are several applications where we may need to estimate a series of closely related sparse signal. In such situations solving a new optimization problem every time becomes computationally expensive. So whenever signal is changing slowly over time our homotopy algorithm can update its estimate at little computational cost. The algorithm is most effective when the support of the solution does not change too much from instance to instance. The simulation results further show that for the dynamic update in small to medium size problems, these homotopy methods outperform warm started GPSR and FPC methods.

REFERENCES

[1] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[2] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[3] E. Candes, "Compressive sampling," *Proceedings of the International Congress of Mathematicians, Madrid, Spain*, vol. 3, pp. 1433–1452, 2006.

[4] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

[6] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[7] M. Lustig, D. Donoho, J. Santos, and J. Pauly, "Compressed Sensing MRI [A look at how CS can improve on current imaging techniques]," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 72–82, March 2008.

[8] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *Communications, IEEE Transactions on*, vol. 50, no. 3, pp. 374–377, 2002.

[9] M. Osborne, B. Presnell, and B. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[11] D. Malioutov, M. Cetin, and A. Willsky, "Homotopy continuation for sparse signal representation," *IEEE International Conference on Acoustics, Speech, and Signal Processing,*, vol. 5, pp. v/733–v/736, March 2005.

[12] D. Bertsekas, *Nonlinear programming*. Athena Scientific Belmont, Mass, 1999.

[13] J. Fuchs, "On sparse representations in arbitrary redundant bases," *Information Theory, IEEE Transactions on*, vol. 50, no. 6, pp. 1341–1344, 2004.

[14] G. James, P. Radchenko, and J. Lv, "The DASSO algorithm for fitting the Dantzig selector and the Lasso," *Journal of the Royal Statistical Society, Series B*, vol. 71, pp. 127–142, 2009.

[15] M. S. Asif, "Primal Dual Pursuit: A homotopy based algorithm for the Dantzig selector," Master's thesis, Georgia Institute of Technology, August 2008.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.

[17] G. Golub and C. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 1996.

[18] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 586–597, 2007.

[19] Z. Wen and W. Yin, "FPC_AS: A MATLAB Solver for ℓ_1 Regularized Least Squares Problems." http://www.caam.rice.edu/∼optimization/L1/FPC_AS/.

[20] J. Buckheit, S. Chen, D. Donoho, and I. Johnstone, "Wavelab 850, Software toolbox." http://www-stat.stanford.edu/∼wavelab/.