Dantzig selector homotopy with dynamic measurements

M. Salman Asif[a] and Justin Romberg[a]

[a] School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA, 30332

ABSTRACT

The Dantzig selector is a near ideal estimator for recovery of sparse signals from linear measurements in the presence of noise. It is a convex optimization problem which can be recast into a linear program (LP) for real data, and solved using some LP solver. In this paper we present an alternative approach to solve the Dantzig selector which we call "Primal Dual pursuit" or "PD pursuit". It is a homotopy continuation based algorithm, which iteratively computes the solution of Dantzig selector for a series of relaxed problems. At each step the previous solution is updated using the optimality conditions defined by the Dantzig selector. We will also discuss an extension of PD pursuit which can quickly update the solution for Dantzig selector when new measurements are added to the system. We will present the derivation and working details of these algorithms.

Keywords: compressive sensing, ℓ_1 norm minimization, sparse signal recovery, statistical estimation, online observations

1. INTRODUCTION

The theory of compressive sensing provides us with a general framework under which a *sparse* signal can be recovered from a small number of linear measurements.[1–4] The general setup is as follows: Suppose there is an unknown signal $x \in \mathbb{R}^n$ and we have its m *measurements* given as $y = Ax$, where A is an $m \times n$ matrix and $m \ll n$. If x is *sparse* i.e., it has a small number of non-zero elements, and the columns of A obey some *incoherence* property,[5] then we can recover x exactly, even though in general it is not possible. We do so by solving the following convex optimization problem

$$\underset{\tilde{x}}{\text{minimize}}\ \|\tilde{x}\|_1 \quad \text{subject to} \quad A\tilde{x} = y. \tag{1}$$

In real world systems the measurements will never be free of noise. To account for such inaccuracies, consider the following model with noisy measurements: $y = Ax + e$, where $e \in \mathbb{R}^m$ is the stochastic or deterministic noise. To accommodate this noise we can replace the equality constraint in (1) with a relaxed *data fidelity* constraint. The Dantzig selector (DS)[6] is a robust estimator for this purpose. The approximation error for DS is within a logarithmic factor of the error achieved by an ideal estimator (for details see Ref. 6). It replaces equality constraint in (1) with the *bounded residual error correlation* constraint, given as

$$\underset{\tilde{x}}{\text{minimize}}\ \|\tilde{x}\|_1 \quad \text{subject to} \quad \|A^T(A\tilde{x} - y)\|_\infty \leq \tau, \tag{2}$$

for some suitable $\tau > 0$. It is also a convex program which can be recast as an LP, and is essentially as easy to solve as (1). A typical way to solve such problems is via interior point methods. The main computational cost of such optimization methods comes from solving a complete $n \times n$ system of linear equations several times, each of which costs $O(n^3)$ flops.

In this paper we will present a fast and efficient scheme to solve (2), which we call "Primal Dual pursuit" or "PD pursuit".[7] It is a homotopy continuation based algorithm which successively builds the solution of (2) in a fashion of "one element at a time". So instead of solving a complete system of equations, we just need a simple rank one update at each step. In particular, if we use explicit matrices then this method is very fast compared to the conventional LP solvers. Another advantage of the homotopy algorithm is that it provides

E-mail: {sasif,jrom}@ece.gatech.edu. This work was partially funded by an ONR Young Investigator Award.

us with a series of solutions to (2) for a range of τ. Whereas, in LP framework one has to solve (2) for each value of τ separately. In addition to solving (2), our algorithm illuminates how DS works; it is essentially like a *soft thresholding* for under-determined systems.[8] Our proposed algorithm is similar in nature to the homotopy method for LASSO[9,10] and its approximation LARS;[11] a relation which we will explore later. There also exists another homotopy algorithm for (2) called DASSO,[12] which is based on sequential simplex-like algorithm.

In the second part of this paper we will discuss another homotopy scheme, based on PD pursuit, which can quickly update the solution of DS as new measurements are added to the system. The problem setup is as follows: Assume that we have solved (2) for a given value of τ and then we receive a new measurement $w = bx + d$, where b is a new row in the measurement matrix and d represents error in the new measurement. This gives us the following updated system

$$\begin{bmatrix} y \\ w \end{bmatrix} = \begin{bmatrix} A \\ b \end{bmatrix} x + \begin{bmatrix} e \\ d \end{bmatrix}. \tag{3}$$

In order to estimate x from this new system we want to solve the following updated version of (2)

$$\underset{\tilde{x}}{\text{minimize}}\ \|\tilde{x}\|_1 \ \text{ subject to } \ \|A^T(A\tilde{x} - y) + b^T(b\tilde{x} - w)\|_\infty \le \tau, \tag{4}$$

for the same value of τ. Our goal here is to find the solution of (4) without having to solve this new optimization problem from scratch. We will present a homotopy algorithm which serves this purpose; instead of solving (4) from scratch it builds a homotopy path from the already computed solution of (2) to that of (4).

The organization of this paper is as follows, in section 2 we will present the derivation and working details of PD pursuit, along with the optimality conditions which are used to derive the homotopy algorithm. Section 3 discusses the homotopy algorithm for dynamic update of measurements. We will conclude with a brief summary of our findings in section 4.

2. PRIMAL DUAL PURSUIT

2.1 Problem Formulation

In the formulation of this homotopy algorithm we use strong duality between primal and dual objectives of DS. We will first present the dual form of (2) and then derive the *optimality* conditions needed to be satisfied by any of its primal and dual solution for a particular value of τ. Then we will see that by changing τ we can construct the homotopy continuation for (2).

The dual problem to (2) can be written as

$$\underset{\lambda}{\text{maximize}}\ -(\tau\|\lambda\|_1 + \langle \lambda, A^T y\rangle) \quad \text{subject to} \quad \|A^T A\lambda\|_\infty \le 1, \tag{5}$$

where $\lambda \in \mathbb{R}^n$ is the dual vector. Using strong duality between the primal and dual objectives in (2) and (5) respectively at any optimal primal-dual solution pair (x^*, λ^*) we get the following equality

$$\|x^*\|_1 = -\tau\|\lambda^*\|_1 - \langle \lambda^*, A^T y\rangle,$$

which can equivalently be written as

$$\|x^*\|_1 + \tau\|\lambda^*\|_1 = -\langle x^*, A^T A\lambda^*\rangle + \langle \lambda^*, A^T(Ax^* - y)\rangle. \tag{6}$$

The complementary slackness condition implies that at any optimal solution point only those elements in the dual vector will be non-zero for which the corresponding primal inequality constraints are active (i.e., hold with equality), similarly for the dual constraints and elements in the primal vector.[13] So using (6) and the feasibility conditions in (2) and (5):

$$\|A^T(Ax - y)\|_\infty \le \tau \tag{7a}$$

$$\|A^T A\lambda\|_\infty \le 1, \tag{7b}$$

we get the following four optimality conditions which must be obeyed by the solution pair (x^*, λ^*) to (2) and (5) at any given τ:

K1. $A^T_{\Gamma_\lambda}(Ax^* - y) = \tau z_\lambda$

K2. $A^T_{\Gamma_x} A\lambda^* = -z_x$

K3. $|a^T_\gamma (Ax^* - y)| < \tau \quad$ for all $\quad \gamma \in \Gamma^c_\lambda$

K4. $|a^T_\gamma A\lambda^*| < 1 \quad$ for all $\gamma \in \Gamma^c_x$,

where Γ_x and Γ_λ is the support of x^* and λ^*, z_x and z_λ are the sign sequences of x^* and λ^* on their respective supports. It can be shown that (K1-K4) are necessary and sufficient conditions for any primal-dual pair $(\hat{x}, \hat{\lambda})$ to be the unique solution of (2) and (5), with respective supports and sign sequences at any given value of τ. So solving (2) is simply finding a primal-dual pair which satisfies (K1-K4). The homotopy method for the Dantzig selector can now be derived using these optimality conditions.

The working principle of homotopy methods (in general) is to trace the solution path of an optimization problem parameterized by some homotopy parameter. We start from an easily computable initial solution and iteratively move towards the desired solution by gradually changing the homotopy parameter. In PD pursuit we select τ as the homotopy parameter. Let $k \in \{1, 2, \ldots\}$ to be the homotopy iteration index. In our method this implies following the path traced by a sequence of primal-dual pairs (x_k, λ_k) towards the final solution (x^*, λ^*), while changing τ. We start from a very large value of τ (e.g., $\tau_1 > \|A^T y\|_\infty$ where solution x_1 is a zero vector) and reduce it towards the final desired value, while updating the primal-dual solution pair (x_k, λ_k). It is important to note from (K1-K4) that at any value of τ_k, the active primal constraints determine the support (Γ_λ) and sign sequence (z_λ) for the dual vector (λ_k) and the active dual constraints determine the support (Γ_x) and sign sequence (z_x) for primal vector (x_k). Moreover, $(\Gamma_\lambda, \Gamma_x, z_\lambda, z_x)$ for every τ_k completely define the homotopy path for the Dantzig selector, as depicted by the following equations

$$x_k = \tau_k (A^T_{\Gamma_\lambda} A_{\Gamma_x})^{-1} z_\lambda + (A^T_{\Gamma_\lambda} A_{\Gamma_x})^{-1} A^T_{\Gamma_\lambda} y \tag{8a}$$

$$\lambda_k = -(A^T_{\Gamma_x} A_{\Gamma_\lambda})^{-1} z_x. \tag{8b}$$

In PD pursuit we use (K1-K4) to continuously update the supports and sign sequences for primal and dual vectors, while reducing τ towards the desired value. Along this homotopy path there are some *critical* values of τ where the support of primal and/or dual vectors change. So our algorithm essentially traverses through this homotopy path while updating the supports of primal-dual vectors on these critical values of τ.

2.2 Algorithm

This algorithm can be divided into two main parts: Primal update and Dual update. In primal update phase we update the primal vector using the primal feasibility conditions in (7a). In dual update phase we use the information from primal update phase to update the dual vector using the dual feasibility conditions in (7b); hence the name "Primal Dual pursuit". A pseudo-code for this algorithm is given in Algorithm 1.

Suppose at the beginning of kth step we have the solution (x_k, λ_k) at a particular value of τ_k with corresponding supports and sign sequences $(\Gamma_\lambda, \Gamma_x, z_\lambda, z_x)$.

2.2.1 Primal update

Compute the update direction ∂x, which minimizes τ by most, as defined in (11) and set $x_{k+1} = x_k + \delta \partial x$, where $\delta > 0$. As we increase δ from zero, primal constraints will change and all the active constraints will *shrink* exactly by a factor of δ in magnitude. This can be thought of as "shrinkage" of the primal constraints.[8] In this way we can encounter two possibilities; either a new element enters Γ_λ (i.e., if an inactive primal constraint becomes active) or a non-zero element of x_k from within Γ_x shrinks to zero. This happens due to the fact that the homotopy path for x_k is piecewise linear[14] w.r.t. τ_k, as shown in (8a). We choose δ depending on which case appears first, as described in (9)

$$
\begin{aligned}
|a_\gamma^T(Ax_{k+1}-y)| = \tau_{k+1} \quad \forall\, \gamma \in \Gamma_\lambda, \qquad |a_\gamma^T(Ax_{k+1}-y)| \le \tau_{k+1} \quad \forall\, \gamma \in \Gamma_\lambda^c \\
|\underbrace{a_\gamma^T(Ax_k-y)}_{p_k(\gamma)} + \delta \underbrace{a_\gamma^T A\partial x}_{d_k(\gamma)}| \le \tau_k - \delta \quad \forall\, \gamma \in \Gamma_\lambda^c \\
|p_k(\gamma) + \delta d_k(\gamma)| \le \tau_k - \delta \quad \forall\, \gamma \in \Gamma_\lambda^c \\
\delta^+ = \min_{i\in\Gamma_\lambda^c}\left(\frac{\tau_k - p_k(i)}{1+d_k(i)}, \frac{\tau_k + p_k(i)}{1-d_k(i)}\right)_+, \qquad \delta^- = \min_{i\in\Gamma_x}\left(\frac{-x_k(i)}{\partial x(i)}\right)_+ \\
\delta = \min(\delta^+, \delta^-),
\end{aligned}
\tag{9}
$$

where $\min(\cdot)_+$ denotes that minimum is taken over positive arguments only. Let us call the indices corresponding to δ^+ and δ^- as i^+ and i^- respectively. So either i^+ enters Γ_λ (if $\delta^+ < \delta^-$) or i^- leaves Γ_x (if $\delta^+ > \delta^-$) and signs are updated accordingly. The new value of homotopy parameter becomes $\tau_{k+1} = \tau_k - \delta$.

2.2.2 Dual update

The dual update works in almost same way as primal update except that here we use the additional information from primal update phase. So similarly, compute update direction $\partial\lambda$ in (12) and set $\lambda_{k+1} = \lambda_k + \theta\partial\lambda$, where $\theta > 0$ is the step size. In contrast to the primal constraints, the dual constraints do not shrink and the path taken by λ_k is piecewise constant. However, we can still use similar procedure to update the supports with some extra checks while computing $\partial\lambda$ (see sec. 2.2.3). So as θ increases from zero, either a new element enters Γ_x (i.e., if an inactive dual constraint becomes active) or an element of λ_k from within Γ_λ shrinks to zero. And we select step size accordingly as described in (10).

$$
\begin{aligned}
|a_\nu^T A\lambda_{k+1}| = 1 \quad \forall\, \nu \in \Gamma_x, \qquad |a_\nu^T A\lambda_{k+1}| \le 1 \quad \forall\, \nu \in \Gamma_x^c \\
|\underbrace{a_\nu^T A\lambda_k}_{a_k(\nu)} + \theta \underbrace{a_\nu^T A\partial\lambda}_{b_k(\nu)}| \le 1 \qquad \forall\, \nu \in \Gamma_x^c \\
|a_k(\nu) + \theta b_k(\nu))| \le 1 \quad \forall\, \nu \in \Gamma_x^c \\
\theta^+ = \min_{j\in\Gamma_x^c}\left(\frac{1-a_k(j)}{b_k(j)}, \frac{1+a_k(j)}{-b_k(j)}\right)_+, \qquad \theta^- = \min_{j\in\Gamma_\lambda}\left(\frac{-\lambda(j)}{\partial\lambda(j)}\right)_+ \\
\theta = \min(\theta^+, \theta^-).
\end{aligned}
\tag{10}
$$

Let us call the indices corresponding to θ^+ and θ^- as j^+ and j^- respectively. So either j^+ enters Γ_x (if $\theta^+ < \theta^-$) or j^- leaves Γ_λ (if $\theta^+ > \theta^-$) and signs are updated accordingly.

2.2.3 Update directions

The update directions for primal and dual vectors can be derived using the optimality conditions (K1-K4). Suppose we are at the vertex (x_k, λ_k) corresponding to τ_k with supports and signs $(\Gamma_\lambda, \Gamma_x, z_\lambda, z_x)$. For primal vector; we want to update x_k in the direction ∂x which causes maximum decrease in τ_k. So using (K1) we get the following update direction for x_k:

$$
\partial x = \begin{cases} -(A_{\Gamma_\lambda}^T A_{\Gamma_x})^{-1} z_\lambda & \text{on } \Gamma_x \\ 0 & \text{elsewhere} \end{cases}. \tag{11}
$$

In order to derive the dual update direction $\partial\lambda$, we will use (K2) with the additional information from primal update. Let us assume that a new element γ enters Γ_λ in the primal update phase. Then the update direction for λ_k can be defined as

$$
\partial\lambda = \begin{cases} -z_\gamma (A_{\Gamma_x}^T A_{\Gamma_\lambda})^{-1} A_{\Gamma_x}^T a_\gamma & \text{on } \Gamma_\lambda \\ z_\gamma & \text{on } \gamma \\ 0 & \text{elsewhere} \end{cases}, \tag{12}
$$

where a_γ is the γth column of A, z_γ is the sign of γth primal active constraint which in fact is sign of the new element in λ_{k+1}.[7] To see why this is true, assume that $\lambda_{k+1} = \lambda_k + \tilde{\theta}\tilde{\partial\lambda}$, with the update direction $\tilde{\partial\lambda}$ and step size $\tilde{\theta}$. Since the dual constraints are active on Γ_x along the homotopy path between τ_k and τ_{k+1}, for a small step size $\tilde{\theta} > 0$ we can write those active constraints as

$$\begin{aligned} A^T_{\Gamma_x} A\tilde{\lambda} &= -z_x \\ A^T_{\Gamma_x} A\lambda_k + \tilde{\theta} A^T_{\Gamma_x} A\tilde{\partial\lambda} &= -z_x \\ A^T_{\Gamma_x} A_{\Gamma_\lambda} u + A^T_{\Gamma_x} a_\gamma v &= 0, \end{aligned}$$

where u is the restriction of $\tilde{\partial\lambda}$ on Γ_λ and v is the value of $\tilde{\partial\lambda}$ on γth index. Since we already know the sign of v from the primal update phase, we can write $v = cz_\gamma$, where c is some positive scalar. This gives

$$u = -cz_\gamma (A^T_{\Gamma_x} A_{\Gamma_\lambda})^{-1} A^T_{\Gamma_x} a_\gamma, \quad v = cz_\gamma, \tag{13}$$

which is precisely what is given in (12) with $c = 1$ (true value of c will be adjusted by the step size θ).

Now consider the case when an element of x_k at index γ_x leaves Γ_x during primal update phase i.e., $\Gamma_x^{k+1} = \Gamma_x^k \backslash \gamma_x$. Here we can pick an element $\gamma_\lambda \in \Gamma_\lambda$ such that the matrix $A^T_{\Gamma_x} A_{\tilde{\Gamma}_\lambda}$ is invertible, where $\tilde{\Gamma}_\lambda := \Gamma_\lambda \backslash \gamma_\lambda$. Invertibility of the new matrix can be easily checked by looking at inverse of its Schur complement; which must be nonzero.[7] Using (12) with $\gamma = \gamma_\lambda, z_\gamma = 1$ and $\Gamma_\lambda = \tilde{\Gamma}_\lambda$ we can compute $\partial\lambda$. Since here we do not know the *actual* sign of λ_{k+1} at γth index, i.e., c can be negative in (13), so we may need to flip sign of $\partial\lambda$ while computing θ in (10) if $\text{sign}[a_k(\gamma_x)] = \text{sign}[b_k(\gamma_x)]$. Also note that $\lambda_k(\gamma_\lambda)$ is not necessarily zero here, so it can also shrink.

2.2.4 Initialization

Start with $x_0 = 0, \lambda_0 = 0, \Gamma_x = [\ ], \Gamma_\lambda = [\ ], z_x = [\ ], z_\lambda = [\ ]$. For the first step pick $\tau_1 = \|A^T\lambda\|_\infty$, because for any large value $x = 0$. This gives $\Gamma_\lambda = \{\gamma\}$, where γ corresponds to the only active primal constraint. After this perform the dual update* as described in Algorithm 1.

2.3 Numerical Implementation

The main computational cost of this algorithm comes from computing the update directions ∂x and $\partial\lambda$, and respective step sizes δ and θ. We have to solve a system of equations at every step involving inversion of the square matrix $G_k := A^T_{\Gamma_x} A_{\Gamma_\lambda}$ or its transpose. Since our algorithm constructs the solution by adding or removing "one element at a time", so we do not need to solve a complete system of equations at every step; each of which will cost $O(S_k^3 + S_k^2 m)$ flops, where S_k is the size of support at kth step. Instead we can easily update the inverse of G_k, whenever Γ_x and/or Γ_λ change, using matrix inversion lemma with a cost of about $O(S_k m)$ flops. The cost associated to compute step sizes is about $O(mn)$. So the computational cost at each step is essentially same as a few matrix-vector multiplications, which cannot be reduced if we use matrices explicitly. The total cost of Algorithm 1 is bounded above by $O(dmn)$, where d is total number of homotopy iterations.

MATLAB files for this implementation are available on this webpage: http://users.ece.gatech.edu/∼sasif

2.4 S-step Solution Property

In[7] we presented some conditions under which any S-sparse signal x can be recovered from its noiseless measurements in exactly S-steps of Algorithm 1; also called S-step solution property. The conditions essentially require that any element which once enters the support never leaves it and only correct elements enter the support at each step. We also showed that in case of random matrix A (Gaussian or Bernoulli), S-step solution property holds with high probability if

$$m \gtrsim S^2 \log n.$$

However, in simulations we have observed that S-step solution property holds even for about $S \log n$ measurements. If S-step solution property holds then the computational cost of Algorithm 1 becomes $O(Smn)$, whereas we do not have any such guarantee if we solve (2) using linear programming.

*support of x and λ will be same for the first step

Algorithm 1 Primal Dual Pursuit Algorithm

Initialize $x_k, \lambda_k, \Gamma_x, \Gamma_\lambda, z_x, z_\lambda$ and τ_k for $k = 1$ as described in sec. 2.2.4
repeat
  $k \leftarrow k + 1$
  Primal update:
  compute the primal update direction ∂x as in (11)
  compute p_k, d_k and δ as in (9)
  $x_{k+1} = x_k + \delta\partial x$
  $\epsilon_{k+1} = \epsilon_k - \delta$
  if $\delta = \delta^-$ **then**
    $\Gamma_x \leftarrow \Gamma_x \setminus i^-$ {remove i^- from supp(x) and update Γ_x}
    $\tilde{\Gamma}_\lambda = \Gamma_\lambda$ {store the current Γ_λ in a dummy variable}
    $\Gamma_\lambda \leftarrow \Gamma_\lambda \setminus \gamma$ {select an index γ from supp(λ) and remove it from Γ_λ}
    $z_\gamma = z_\lambda(\gamma)$ {treat γ as the new element to supp(λ)}
    update z_x, z_λ {update sign sequences on updated supports}
  else
    $\tilde{\Gamma}_\lambda = \Gamma_\lambda \cup \{i^+\}$ {store i^+ but do not update Γ_λ}
    $z_\lambda = \text{sign}[A^T_{\tilde{\Gamma}_\lambda}(Ax_{k+1} - y)]$ {update z_λ}
    $\gamma = i^+$
    $z_\gamma = z_\lambda(\gamma)$
  end if
  Dual update:
  compute the dual update direction $\partial\lambda$ as in (12)
  compute a_k and b_k as in (10)
  if $\delta = \delta^-$ & $\text{sign}[a_k(i^-)] = \text{sign}[b_k(i^-)]$ **then**
    $\partial\lambda \leftarrow -\partial\lambda$ {a check needed due to uncertainty in sign}
    $b_k \leftarrow -b_k$ {flip the sign of $\partial\lambda$ and in turn b_k}
  end if
  compute θ as in (10)
  $\lambda_{k+1} = \lambda_k + \theta\partial\lambda$
  if $\theta = \theta^-$ **then**
    $\Gamma_\lambda \leftarrow \tilde{\Gamma}_\lambda \setminus j^-$ {remove j^- from supp(λ) and update Γ_λ}
    update z_λ {update sign sequence on updated support}
  else
    $\Gamma_x \leftarrow \Gamma_x \cup \{j^+\}$ {add j^+ to supp(x) and update Γ_x}
    $\Gamma_\lambda \leftarrow \tilde{\Gamma}_\lambda$ {set Γ_λ to supp(λ) determined in Primal update}
    $z_x = \text{sign}[A^T_{\Gamma_x} A\lambda_{k+1}]$ {update z_x}
  end if
until $\tau_{k+1} \leq \tau$

2.5 Comparison with LARS and LASSO

The presented homotopy algorithm for DS is very similar to the homotopy method for LASSO. An equivalent formulation for LASSO (also known as *Basis pursuit Denoising*[15]) can be written as the following unconstrained minimization problem

$$\underset{\tilde{x}}{\text{minimize}} \quad \tau\|\tilde{x}\|_1 + \frac{1}{2}\|A\tilde{x} - y\|_2^2. \tag{14}$$

In order to derive homotopy method for (14), we can write its optimality conditions[16] for any solution x^* at a particular value of τ as follows

L1. $A^T_\Gamma(Ax^* - y) = -\tau z$

L2. $|a^T_\gamma(Ax^* - y)| < \tau \quad$ for all $\quad \gamma \in \Gamma^c$

where Γ is the support of x^* and z is its sign sequence on Γ. This gives the following update direction

$$\partial x = \begin{cases} (A_\Gamma^T A_\Gamma)^{-1} z & \text{on } \Gamma \\ 0 & \text{on } \Gamma^c \end{cases}. \tag{15}$$

Its homotopy steps will be same as the primal update phase in Algorithm 1. Note that Lasso optimality conditions (L1-L2) are exactly same as the Dantzig selector optimality conditions (K1-K4) when $\Gamma_\lambda = \Gamma_x$ and $z_\lambda = -z_x$. So in Algorithm 1 if we perform only the primal update, set $\Gamma_x = \Gamma_\lambda$ and $z_\lambda = -z_x$, then it is same as the homotopy method for LASSO.[17] In addition, if we omit the step which removes an element from the support as well, then it is same as LARS.[11] So Algorithm 1, actually a homotopy for the Dantzig selector, can be considered as a generalization of homotopy for LASSO and LARS. This similarity is also discussed by James et al. in Ref. 12 for DASSO. In addition to this, it can also be shown that if S-step solution property holds then Dantzig selector, LASSO and LARS all have exactly same homotopy path[7] and vice versa.

3. DYNAMIC MUEASUREMENT UPDATE

3.1 Problem Formulation

In this section we will present the homotopy algorithm which can quickly update the solution of DS when new measurements are added, as described in (3). In order to solve (4), by using the already computed solution of (2), we propose the following homotopy formulation

$$\underset{\tilde{x}}{\text{minimize}} \ \|\tilde{x}\|_1 \ \text{ subject to } \ \|A^T(A\tilde{x} - y) + \epsilon b^T(b\tilde{x} - w)\|_\infty \leq \tau, \tag{16}$$

where τ is fixed and $\epsilon \in [0, 1]$ is the homotopy parameter here. Note that at $\epsilon = 0$ the solution to (16) is same as that for (2). As ϵ changes from 0 to 1 the solution to (16) traces a homotopy path towards the solution of (4).

In order to derive the homotopy scheme we need the optimality conditions for (16) as well. Similarly, we can write the required optimality conditions for any solution pair (x^*, λ^*) at any given value of ϵ as follows:

D1. $A_{\Gamma_\lambda}^T(Ax^* - y) + \epsilon b_{\Gamma_\lambda}^T(bx^* - w) = \tau z_\lambda$

D2. $A_{\Gamma_x}^T A\lambda^* + \epsilon b_{\Gamma_x}^T b\lambda^* = -z_x$

D3. $|a_\gamma^T(Ax^* - y) + \epsilon b_\gamma^T(bx^* - w)| < \tau \quad$ for all $\quad \gamma \in \Gamma_\lambda^c$

D4. $|a_\gamma^T A\lambda^* + \epsilon b_\gamma^T b\lambda^*| < 1 \quad$ for all $\ \gamma \in \Gamma_x^c$,

where Γ_x and Γ_λ is the support of x^* and λ^*, z_x and z_λ are the sign sequences of x^* and λ^* on their respective supports. Similarly, homotopy path for this problem is also completely defined by the primal-dual supports and sign sequences $(\Gamma_\lambda, \Gamma_x, z_\lambda, z_x)$ for every value of ϵ. Along this homotopy path there are some critical values of ϵ where the support of primal and/or dual vectors change, and our algorithm essentially finds those critical values and updates supports at those points along the homotopy path.

3.2 Algorithm

The algorithm we discuss here for dynamic measurement update in DS is (in principle) similar to the dynamic update for Lasso presented in Ref. 18, along with the fact that here we need to update both primal and dual vectors at every homotopy step and take care of both primal and dual constraints. This algorithm can also be divided into two main parts: In the first phase we change both primal and dual vectors (x, λ) in their respective update directions $(\partial x, \partial\lambda)$, which increase ϵ by most, until there is some change in the support of primal or dual vector; this determines the new critical value of ϵ. Then depending on the outcome of this phase i.e., whether the support is changed by the primal or dual vector, we fix the vector which caused the change and update the other one.

3.2.1 Phase 1

Let us assume that we already have the primal-dual solution pair (x_k, λ_k) to (16) at some critical value of $\epsilon = \epsilon_k$, with respective supports and sign sequences $(\Gamma_\lambda, \Gamma_x, z_\lambda, z_x)$. The corresponding optimality conditions in (D1-D4) can be written as

$$A^T_{\Gamma_\lambda}(Ax_k - y) + \epsilon_k b^T_{\Gamma_\lambda}(bx_k - w) = \tau z_\lambda \tag{17a}$$

$$A^T_{\Gamma_x} A\lambda_k + \epsilon_k b^T_{\Gamma_x} b\lambda_k = -z_x \tag{17b}$$

$$\|A^T_{\Gamma^c_\lambda}(Ax_k - y) + \epsilon_k b^T_{\Gamma^c_\lambda}(bx_k - w)\|_\infty < \tau \tag{17c}$$

$$\|A^T_{\Gamma^c_x} A\lambda_k + \epsilon_k b^T_{\Gamma^c_x} b\lambda_k\|_\infty < 1. \tag{17d}$$

Now we need to find update directions $(\partial x, \partial\lambda)$ for primal-dual pair (x_k, λ_k) such that ϵ increases by most. These directions can be derived using (17a) and (17b). The update directions which change ϵ from ϵ_k to ϵ_{k+1} can be written as

$$\widetilde{\partial x} = \begin{cases} -(\epsilon_{k+1} - \epsilon_k)(A^T_{\Gamma_\lambda} A_{\Gamma_x} + \epsilon_{k+1} b^T_{\Gamma_\lambda} b_{\Gamma_x})^{-1} b^T_{\Gamma_\lambda}(bx_k - w) & \text{on } \Gamma_x \\ 0 & \text{otherwise} \end{cases},$$

$$\widetilde{\partial \lambda} = \begin{cases} -(\epsilon_{k+1} - \epsilon_k)(A^T_{\Gamma_x} A_{\Gamma_\lambda} + \epsilon_{k+1} b^T_{\Gamma_x} b_{\Gamma_\lambda})^{-1} b^T_{\Gamma_x} b\lambda_k & \text{on } \Gamma_\lambda \\ 0 & \text{otherwise} \end{cases}.$$

Similarly the primal and dual constraints are changed with these update directions as follows:

$$\|A^T[A(x_k + \widetilde{\partial x}) - y] + \epsilon_{k+1} b^T[b(x_k + \widetilde{\partial x}_1) - w]\|_\infty \le \tau,$$
$$\|A^T A(\lambda_k + \widetilde{\partial \lambda}) + \epsilon_{k+1} b^T b(\lambda_k + \widetilde{\partial \lambda})\|_\infty \le 1.$$

We can further simplify these equations using matrix inversion lemma,[19] to separate the step size from the update directions, which gives us the following update equations for primal-dual vectors and their constraints (let $u := b_{\Gamma_x}(A^T_{\Gamma_\lambda} A_{\Gamma_x} + \epsilon_k b^T_{\Gamma_\lambda} b_{\Gamma_x})^{-1} b^T_{\Gamma_\lambda}$)

$$\partial x_1 = \begin{cases} -(A^T_{\Gamma_\lambda} A_{\Gamma_x} + \epsilon_k b^T_{\Gamma_\lambda} b_{\Gamma_x})^{-1} b^T_{\Gamma_\lambda}(bx_k - w) & \text{on } \Gamma_x \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

$$\delta_x = \frac{\epsilon_{k+1} - \epsilon_k}{1 + (\epsilon_{k+1} - \epsilon_k)u}, \tag{19}$$

$$\|\underbrace{A^T(Ax_k - y) + \epsilon_k b^T(bx_k - w)}_{p_k} + \delta_x \underbrace{(A^T A + \epsilon_k b^T b)\partial x_1 + b^T(bx_k - w)}_{d_k}\|_\infty \le \tau, \tag{20}$$

$$\partial \lambda_1 = \begin{cases} -(A^T_{\Gamma_x} A_{\Gamma_\lambda} + \epsilon_k b^T_{\Gamma_x} b_{\Gamma_\lambda})^{-1} b^T_{\Gamma_x} b\lambda_k & \text{on } \Gamma_\lambda \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

$$\delta_\lambda = \frac{\epsilon_{k+1} - \epsilon_k}{1 + (\epsilon_{k+1} - \epsilon_k)u}, \tag{22}$$

$$\|\underbrace{A^T A\lambda_k + \epsilon_k b^T b\lambda_k}_{a_k} + \delta_\lambda \underbrace{(A^T A + \epsilon_k b^T b)\partial\lambda_1 + b^T b\lambda_k}_{b_k}\|_\infty \le 1, \tag{23}$$

where ∂x_1 and $\partial \lambda_1$ are the new primal-dual update directions, δ_x and δ_λ are their respective step sizes required to change ϵ from ϵ_k to some nearby ϵ_{k+1}. Subscript 1 here is to denote the phase 1.

So if we move in direction ∂x_1 by increasing δ_x (this is like primal update phase in PD pursuit), ϵ increases and at some point either a new element will enter the support Γ_λ (i.e., an inactive primal constraint in (20) will become active) or an existing element in Γ_x will leave (i.e., a non-zero element in x_k will shrink to zero).

$$\delta_x^+ = \min_{i \in \Gamma^c_\lambda} \left(\frac{\tau - p_k(i)}{d_k(i)}, \frac{\tau + p_k(i)}{-d_k(i)} \right)_+, \qquad \delta_x^- = \min_{i \in \Gamma_x} \left(\frac{-x_k(i)}{\partial x_1(i)} \right)_+ \tag{24}$$
$$\delta_x = \min(\delta_x^+, \delta_x^-),$$

This gives us a critical value of ϵ, let us denote it as $\epsilon_x := \epsilon_k + \dfrac{\delta_x}{1-\delta_x u}$.

In exactly same way, for dual vector, as we move in direction $\partial\lambda_1$ by increasing δ_λ, ϵ increases and at some critical value either a new element will enter the primal support Γ_x or an existing element will leave Γ_λ.

$$\delta_\lambda^+ = \min_{i\in\Gamma_x^c}\left(\frac{1-a_k(i)}{b_k(i)}, \frac{1+a_k(i)}{-b_k(i)}\right)_+, \qquad \delta_\lambda^- = \min_{i\in\Gamma_\lambda}\left(\frac{-\lambda_k(i)}{\partial\lambda_1(i)}\right)_+ \\ \delta_\lambda = \min(\delta_\lambda^+, \delta_\lambda^-), \tag{25}$$

This gives us another critical value of ϵ, let us denote it as $\epsilon_\lambda := \epsilon_k + \dfrac{\delta_\lambda}{1-\delta_\lambda u}$.

Now we check whether ϵ_x or ϵ_λ is smaller (equivalent to checking smaller of δ_x and δ_λ), this gives us the actual next critical value of ϵ on the homotopy path, let us denote it as $\epsilon_{k+1} := \min(\epsilon_x, \epsilon_\lambda)$. In addition to this, let us denote γ as the index of element which is either added or removed from primal or dual support in this phase. It is like finding the largest step size and corresponding critical point on homotopy path for support update while obeying all the optimality conditions. So we update primal-dual vectors and constraints using the smaller step size $\delta := \min(\delta_x, \delta_\lambda)$ as

$$\tilde{x}_k = x_k + \delta\partial x_1, \quad \tilde{\lambda}_k = \lambda_k + \delta\partial\lambda_1, \quad \tilde{p}_k = p_k + \delta d_k, \quad \tilde{a}_k = a_k + \delta b_k, \tag{26}$$

which takes both vectors to the critical point where $\epsilon = \epsilon_{k+1}$. Then depending on whether ϵ_x or ϵ_λ is smaller, the corresponding vector and respective constraints are fixed at the critical point and using information from this phase the other vector and its constraints are updated in phase 2.

3.2.2 Phase 2

In this phase we start with primal-dual vectors $(\tilde{x}_k, \tilde{\lambda}_k)$ at critical point and update only the vector (from critical point onwards) which did not cause support change in phase 1; it is analogous to dual update in PD pursuit.

Assume that $\delta = \delta_x$, i.e., support change in phase 1 was caused by primal vector. So here we keep the primal vector $(\tilde{x}_k)$ and constraints $(\tilde{p}_k)$ fixed and change dual vector and constraints. Also assume that a new element entered† the support of dual vector during phase 1 at index γ with sign z_γ. Then using this information we can write the following equations for dual update direction and constraints

$$\partial\lambda_2 = \begin{cases} -(A_{\Gamma_x}^T A_{\Gamma_\lambda} + \epsilon_{k+1} b_{\Gamma_x}^T b_{\Gamma_\lambda})^{-1}(A_{\Gamma_x}^T a_\gamma + \epsilon_{k+1} b_{\Gamma_x}^T b_\gamma) z_\gamma & \text{on } \Gamma_\lambda \\ z_\gamma & \text{on } \gamma \\ 0 & \text{elsewhere} \end{cases}, \tag{27}$$

$$\| \underbrace{A^T A\tilde{\lambda}_k + \epsilon_{k+1} b^T b\tilde{\lambda}_k}_{\tilde{a}_k} + \theta_\lambda \underbrace{(A^T A + \epsilon_{k+1} b^T b)\partial\lambda_2}_{\tilde{b}_k} \|_\infty \le 1. \tag{28}$$

Then we find the smallest value of θ_λ such that either a new element enters the support Γ_x (i.e., an inactive constraint in (28) becomes active) or an existing element in $\tilde{\lambda}_k$ shrinks to zero, as

$$\theta_\lambda^+ = \min_{i\in\Gamma_x^c}\left(\frac{1-\tilde{a}_k(i)}{\tilde{b}_k(i)}, \frac{1+\tilde{a}_k(i)}{-\tilde{b}_k(i)}\right)_+, \qquad \theta_\lambda^- = \min_{i\in\Gamma_\lambda}\left(\frac{-\tilde{\lambda}_k(i)}{\partial\lambda_2(i)}\right)_+, \qquad \theta_\lambda = \min(\theta_\lambda^+, \theta_\lambda^-).$$

Set $p_{k+1} = \tilde{p}_k$, $a_{k+1} = \tilde{a}_k + \theta_\lambda \tilde{b}_k$, $x_{k+1} = \tilde{x}_k$ and $\lambda_{k+1} = \tilde{\lambda}_k + \theta_\lambda\partial\lambda_2$. Update the support and sign sequence accordingly.

†If instead an element was removed from Γ_x, we can use the same trick as explained in sec. 2.2.3.

Similarly, if $\delta = \delta_\lambda$ in phase 1 and assuming that a new element entered the support of primal vector at index γ with sign z_γ. We will keep dual vector $(\tilde{\lambda}_k)$ and dual constraints $(\tilde{a}_k)$ fixed and change primal vector and constraints using the following set of equations

$$\partial x_2 = \begin{cases} -(A_{\Gamma_\lambda}^T A_{\Gamma_x} + \epsilon_{k+1} b_{\Gamma_\lambda}^T b_{\Gamma_x})^{-1}(A_{\Gamma_\lambda}^T a_\gamma + \epsilon_{k+1} b_{\Gamma_\lambda}^T b_\gamma) z_\gamma & \text{on } \Gamma_x \\ z_\gamma & \text{on } \gamma \\ 0 & \text{elsewhere} \end{cases}, \tag{29}$$

$$\| \underbrace{A^T(A\tilde{x}_k - y) + \epsilon_{k+1} b^T(b\tilde{x}_k - w)}_{\tilde{p}_k} + \theta_x \underbrace{(A^T A + \epsilon_{k+1} b^T b)\partial x_2}_{\tilde{d}_k} \|_\infty \leq \tau. \tag{30}$$

Find the smallest value of θ_x such that either one new element enters the support Γ_λ (i.e., an inactive constraint in (30) becomes active) or an existing element in $\tilde{x}_{k+1}$ shrinks to zero, as

$$\theta_x^+ = \min_{i \in \Gamma_\lambda^c} \left(\frac{\tau - \tilde{p}_k(i)}{\tilde{d}_k(i)}, \frac{\tau + \tilde{p}_k(i)}{-\tilde{d}_k(i)} \right)_+, \qquad \theta_x^- = \min_{i \in \Gamma_x} \left(\frac{-\tilde{x}_k(i)}{\partial x_2(i)} \right)_+, \qquad \theta_x = \min(\theta_x^+, \theta_x^-).$$

Set $p_{k+1} = \tilde{p}_k + \theta_x \tilde{d}_k$, $a_{k+1} = \tilde{a}_k$, $x_{k+1} = \tilde{x}_k + \theta_x \partial x_2$ and $\lambda_{k+1} = \tilde{\lambda}_k$. Update the support and sign sequence accordingly.

Repeat this procedure in phase 1 and phase 2 until ϵ becomes equal to 1. If at any step, on the homotopy path (during phase 1), the value ϵ increases from 1, just stop at $\epsilon = 1$ and quit without any support update.

4. CONCLUSION

In this paper we have presented homotopy algorithm called PD pursuit to solve the Dantzig selector (DS). We used primal and dual forms of DS along with strong duality between the primal-dual objectives to derive the optimality conditions which describe the homotopy continuation path. The computational cost at each homotopy step is very small; involving a rank one update to compute update directions and few matrix-vector multiplications to compute step sizes. So each homotopy step costs about $O(mn)$ flops, which is essentially the cost for few matrix-vector multiplications and cannot be reduced if we use explicit matrices. DASSO[12] uses sequential simplex-like algorithm for updating two supports (primal and dual) and finding update direction for primal vector only. Whereas, we have used dual variable explicitly in PD pursuit, which makes this relation between supports more obvious and requires updating both primal and dual vectors, their supports and sign sequences at each homotopy step. In addition to this, using PD pursuit along with the optimality conditions, we can extend the homotopy procedure to add new measurements or remove[‡] existing measurements dynamically. This way we can avoid solving a new optimization problem from scratch whenever some measurements are updated in the system.

REFERENCES

[1] Candes, E., Romberg, J., and Tao, T., "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on* **52**(2), 489–509 (Feb. 2006).

[2] Donoho, D., "Compressed sensing," *Information Theory, IEEE Transactions on* **52**(4), 1289–1306 (April 2006).

[3] Candes, E. J. and Tao, T., "Near-optimal signal recovery from random projections: Universal encoding strategies?," *Information Theory, IEEE Transactions on* **52**(12), 5406–5425 (Dec. 2006).

[4] Candés, E., "Compressive sampling," *Proceedings of the International Congress of Mathematicians, Madrid, Spain* **3**, 1433–1452 (2006).

[5] Candes, E. and Tao, T., "Decoding by linear programming," *Information Theory, IEEE Transactions on* **51**(12), 4203–4215 (Dec. 2005).

[‡]In order to remove any measurement change respective homotopy parameter from 1 to 0.

[6] Candes, E. and Tao, T., "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics* **35**(6), 2313–2351 (2007).

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

[8] Romberg, J., "The Dantzig selector and generalized thresholding," *Information Sciences and Systems, (CISS). 42nd Annual Conference on* , 22–25 (March 2008).

[9] Osborne, M., Presnell, B., and Turlach, B., "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis* **20**(3), 389–403 (2000).

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

[11] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R., "Least angle regression," *Annals of Statistics* **32**(2), 407–499 (2004).

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

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

[14] Rosset, S. and Zhu, J., "Piecewise linear regularized solution paths," *Annals of statistics* **35**(3), 1012 (2007).

[15] Chen, S. S., Donoho, D. L., and Saunders, M. A., "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing* **20**(1), 33–61 (1999).

[16] Fuchs, J., "On sparse representations in arbitrary redundant bases," *Information Theory, IEEE Transactions on* **50**(6), 1341–1344 (2004).

[17] Donoho, D. L. and Tsaig, Y., "Fast solution of ℓ_1 minimization problems when solution may be sparse," Preprint (October 2006).

[18] Asif, M. S. and Romberg, J., "Streaming measurements in compressive sensing: ℓ_1 filtering," *42nd Asilomar conference on Signals, Systems and Computers* (Oct. 2008).

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