DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS SERIES S
Volume **7**, Number **3**, June **2014**

doi:10.3934/dcdss.2014.7.503

pp. **503–523**

LEARNING AND ESTIMATION APPLICATIONS OF AN ONLINE HOMOTOPY ALGORITHM FOR A GENERALIZATION OF THE LASSO

Aude Hofleitner

Electrical Engineering and Computer Science
UC Berkeley, USA

Tarek Rabbani

Mechanical Engineering
UC Berkeley, USA

Mohammad Rafiee

Mechanical Engineering
UC Berkeley, USA

Laurent El Ghaoui

Electrical Engineering and Computer Science
UC Berkeley, USA

Alex Bayen

Electrical Engineering and Computer Science
Civil and Environmental Engineering
UC Berkeley, USA

Abstract. The LASSO is a widely used shrinkage and selection method for linear regression. We propose a generalization of the LASSO in which the l_1 penalty is applied on a linear transformation of the regression parameters, allowing to input prior information on the structure of the problem and to improve interpretability of the results. We also study time varying system with an l_1-penalty on the variations of the state, leading to estimates that exhibit few "jumps". We propose a homotopy algorithm that updates the solution as additional measurements are available. The algorithm takes advantage of the sparsity of the solution for computational efficiency and is promising for mining large datasets. The algorithm is implemented on three experimental data sets representing applications to traffic estimation from sparsely sampled probe vehicles, flow estimation in tidal channels and text analysis of on-line news.

Least-squares regression with l_1-norm regularization is known as the LASSO algorithm [37]. It has generated significant interest in the statistics [37, 10], signal processing [3, 6, 16] and machine learning [20, 33] communities, in particular for estimation problems. Adding a l_1-penalty usually leads to sparse solutions, which is a desirable property used to achieve model selection, data compression, or to obtain interpretable results.

2010 *Mathematics Subject Classification.* Primary: 49M30, 68W27; Secondary: 68U99.
Key words and phrases. Online estimation, LASSO, convex optimization, algorithm.

The LASSO can be solved using interior-point methods [25], iterative thresholding algorithms [9, 15], feature-sign search [27], bound optimization methods [13], incremental methods [5] or gradient projection algorithms [14]. Homotopy algorithms compute the regularization path [34, 12]. They are particularly efficient when the solution is very sparse [11, 31]. Homotopy algorithms are also powerful to compute online updates [36, 17] when the training examples are obtained sequentially (one at a time). This method is particularly efficient when the support of the LASSO solutions at the particular penalty parameter is similar.

At estimation step n, a set I_n of training examples or observations $(y_i, a_i) \in \mathbb{R} \times \mathbb{R}^m$, $i \in I_n$ is available. The article presents how to fit a linear model to estimate the response y_i as a function of $x \in \mathbb{R}^m$. A linear function of the solution, $K_1 x$, with $K_1 \in \mathbb{R}^{k \times m}$, is expected to be sparse. The matrix K_1 represents inherent structure of the problem or trend filtering [2, 24]. To achieve this property, an l_1 penalty on $K_1 x$ is added to the least-square estimation problem. The resulting optimization problem is given by:

$$\textbf{minimize}_{x \in \mathbb{R}^m} \frac{1}{2} \sum_{i \in I_n} (a_i^T x - y_i)^2 + \mu_n ||K_1 x||_1 \tag{1}$$

For other applications, we may be interested in sparse changes between the state vector and a reference vector $\overline{x}^n$. To achieve this property, an l_1 penalty on the difference between the state vector x and the reference $\overline{x}$ is added to the least-square estimation problem. The estimation problem of x^n is defined as:

$$\textbf{minimize}_{x \in \mathbb{R}^m} \frac{1}{2} \sum_{i \in I_n} (a_i^T x - y_i)^2 + \mu_n ||x - \bar{x}^n||_1. \tag{2}$$

The reference $\bar{x}^n$ may change after each estimation. In particular, the model can encourage sparse temporal variations to regularize the estimates when measurements are noisy and the dynamics of the system is slow compared to the sampling rate. This property is achieved by choosing $\bar{x}^n = x^{n-1}$.

In applications, it is useful to add additional regularization to the optimization problems (1) and (2). In particular, for the solution of the least-squares estimation problem to be unique, the matrix $A^T A$ should be non singular, which is not always the case for some applications. Moreover, the regularization term $\mu_n ||K_1 x||_1$ or $\mu_n ||x - \bar{x}^n||_1$ is on the sparse structure of the estimate but there is no regularization to maintain the state estimates close to an a priori value. As done in the *Elastic Net* [38], the article investigates the addition of an l_2 regularization term with weighting parameter λ to Equations (1) and (2) to improve estimation capabilities. This additional term leverages prior information $\hat{x}$ on the value of the state x (from historical data for example) to improve the estimation capabilities.

The regularization parameter μ_n may depend on the number of measurements $|I_n|$. Example choices are $\mu_n = |I_n|\mu_0$ as in [17] or $\mu_n = \sqrt{|I_n|}\mu_0$ as in [26]. The parameter μ_0 is chosen via cross-validation, as a trade-off between the structure imposed by the regularization, and the fit to the data.

The article presents a general data-driven online estimation algorithm which extends existing work [17, 22] in sparse modeling and estimation. In particular, the article provides online updates of the solution of the LASSO with a l_1 penalty on the difference between the estimate and a reference point. The reference point may change at each update. This last property allows to perform estimation in dynamical system with estimates which exhibit few "jumps" over time. In this

case, the penalty is between successive estimate and the reference is updated at each estimation. The article also presents numerical applications which illustrate the generality of the algorithm for estimation and learning.

The article is organized as follows. Section 1 reviews the optimality conditions of the LASSO algorithm and introduces an existing homotopy algorithm [17] to solve the LASSO problem recursively. Section 2 recalls the results of [22] to update the solution of the LASSO to add (or remove) p observations. In Section 3, the algorithm is adapted to produce estimation with the l_1 penalty imposed between the estimate and a reference point which can vary after each estimation. Section 4 illustrates the potential of the algorithm for estimation and learning applications: traffic estimation from sparsely sampled probe vehicles, flow estimation in tidal channels and text analysis of on-line news.

1. The LASSO problem.

The LASSO problem [37] is defined as follows:

$$\textbf{minimize}_{x\in\mathbb{R}^m} \frac{1}{2}\sum_{i=1}^{n}(a_i^T x - y_i)^2 + \mu_n||x||_1. \tag{3}$$

This section summarizes previous work [12, 17] which uses the optimality conditions to solve this problem. The objective function of (3) is convex and non-smooth since the l_1-norm is not differentiable when there exists an index i such that the i^{th} element of x (denoted x_i) equals zero. There is a global minimum at x if and only if the subdifferential of the objective function at x contains the 0-vector. The subdifferential of the l_1-norm at x is the following set

$$\partial\|x\|_1 = \left\{ v\in\mathbb{R}^m \ : \ \left\{ \begin{array}{rcll} v_i & = & \text{sgn}(x_i) & \text{if} \quad |x_i|>0 \\ v_i & \in & [-1,1] & \text{if} \quad x_i = 0 \end{array} \right\} \right\},$$

where $\text{sgn}(\cdot)$ is the sign function. Let $A\in\mathbb{R}^{|I_n|\times m}$ be the matrix whose i^{th} row is equal to a_i^T, and let $y=(y_i)_{i\in I_n}^T$ be the vector of response variables. The optimality conditions for (3) are given by

$$A^T(Ax-y)+\mu_n v = 0,\ v\in\partial\|x\|_1.$$

Definition 1.1 (Active set)**.** The *active set* a is the set of indices representing non-zero elements of x. The matrix A_a is a selection of the columns of A in a. The non-zero coordinates of x are in x_a. The index a_i references the i^{th} coordinate of the active set. Since $v\in\partial\|x\|_1$, $v_{a_i}=\text{sgn}(x_{a_i})$.

Definition 1.2 (Non active set)**.** The *non active set* na is the set of indices representing zero elements of x. The matrix A_{na} is a selection of the columns of A in na. It follows that x_{na} is the 0-vector. The index na_i references the i^{th} coordinate of the non active set. Since $v\in\partial\|x\|_1$, $v_{na_i}\in[-1,1]$.

If the solution is unique, $A_a^TA_a$ is non-singular[1]. The optimality conditions read

$$\begin{array}{rl} x_a & = (A_a^TA_a)^{-1}(A_a^Ty-\mu_n v_a) \\ -\mu_n v_{na} & = A_{na}^T(A_ax_a-y) \end{array}.$$

Given the active set and the signs of the coefficients of the solution (and thus the vector v_a), the solution x is computed in closed form. When observations come

[1]The Elastic Net [38] ensures the uniqueness of the solution without requiring A^TA to be non-singular.

sequentially, a homotopy algorithm [17] solves the LASSO problem recursively by considering the following problem:

$$x(t,\mu) = \arg\min_{x\in\mathbb{R}^m} \frac{1}{2}\left\| \begin{pmatrix} A \\ ta_{n+1}^T \end{pmatrix} x - \begin{pmatrix} y \\ ty_{n+1} \end{pmatrix} \right\|_2^2 + \mu \|x\|_1 .$$

Adding (resp. removing) a point is equivalent to computing the homotopy path from $t = 0$ to $t = 1$ (resp. from $t = 1$ to $t = 0$). Varying the regularization parameter is equivalent to computing the path from $\mu = \mu_n$ to $\mu = \mu_{n+1}$.

2. **Recursive LASSO with p new observations, l_2 and linear l_1 regularizations.** The section recalls the results presented in [22]. It studies a least square estimation problem, for which a linear transform of the solution, $K_1 x$ for $K_1 \in \mathbb{R}^{k\times m}$ is sparse. The estimate is updated as p new observations $(y^{\text{new}}, A^{\text{new}}) \in \mathbb{R}^p \times \mathbb{R}^{p\times m}$ become available[2]. The algorithm updates the solution online without having to fully recompute it at each estimation step. Let $\hat{x}$ represent a priori information on the solution, which is used as additional regularization when the matrix A is not full column rank or is ill conditioned (see the *Elastic Net* [38] for details). The matrix K_1 is assumed to be full row rank, which is the case for numerous applications including *total variation regularization*. Each row of K_1 corresponds to an information on the sparsity structure of the solution. Let $K_2 \in \mathbb{R}^{m-k\times m}$ be such that $K = (K_1^T\ K_2^T)^T$ is non singular. For example, K_2 is such that the columns of K_2^T form a basis for the null-space of K_1. The non-singular matrix K defines a change of variable $z = Kx$. It is also convenient to define new data matrices $B = AK^{-1}$, $B_{\text{new}} = A_{\text{new}}K^{-1}$ and $\hat{z} = K\hat{x}$. The section develops an algorithm which updates the solution z of

$$\underset{z\in\mathbb{R}^m}{\textbf{minimize}}\ \frac{1}{2}\left\| \begin{pmatrix} B \\ tB^{\text{new}} \end{pmatrix} z - \begin{pmatrix} y \\ ty^{\text{new}} \end{pmatrix} \right\|_2^2 + \mu||(I_k\ 0_{k\times m-k})\ z||_1 + \frac{\lambda}{2}||z-\hat{z}||_2^2. \quad (4)$$

(i) as t varies to add (or remove) observations and (ii) as μ varies to change the weight of the l_1 regularization. The l_1 penalization is on the first k coordinates of z, denoted *regularized indices*. The last $m-k$ indices are in the active set and are referred to as the *non-regularized indices*.

2.1. **Add p observations.** At $t = 0$, the solution $z(0,\mu_n)$ is known, and so are the active set and the signs of the regularized indices of z. Let v_{a_i} be the sign of $z_{a_i}(0)$ for the *regularized indices* and define $v_{a_i} = 0$ for the *non-regularized indices*. The data matrices with the new observations are indicated with a tilde: $\tilde{B} = (B^T B^{\text{new}^T})^T$ and $\tilde{y} = (y^T y^{\text{new}^T})^T$. The optimality conditions of (4) read

$$\tilde{B}_a^T(\tilde{B}_a z_a(t) - \tilde{y}) + (t^2-1)B_a^{\text{new}^T}(B_a^{\text{new}} z_a(t) - y^{\text{new}}) + \mu_n v_a + \lambda(z_a(t) - \hat{z}_a) = 0, \quad (5)$$

$$\tilde{B}_{na}^T(\tilde{B}_a\, z_a(t) - \tilde{y}) + (t^2-1)B_{na}^{\text{new}^T}(B_a^{\text{new}} z_a(t) - y^{\text{new}}) + \mu_n w_{na}(t) - \lambda\hat{z}_{na} = 0. \quad (6)$$

where $w_{na}(t)$ is a vector with coordinates in $[-1,1]$. Notice that, at $t = 0$, $z_a(\cdot)$ and $w_{na}(\cdot)$ are continuous in t. Let t^* to be the largest $t \in [0,1]$ such that: (i) for all $t \in [0,t^*)$, for all i in the regularized indices, $\text{sgn}(z_a(t)) = \text{sgn}(z_a(0))$ and (ii) for all

[2]The solution can also be updated when some of the observations become obsolete.

$t \in [0, t^*)$, for all i in the non-active set, $|w_{na_i}(t)| < 1$. On this interval, v_{a_i} is the sign of $z_{a_i}(t)$ and Equations (5-6) are valid.

The matrix $Q = (\tilde{B}_a^T \tilde{B}_a + \lambda I_{|a|})^{-1}$ is computed from its previous value without the p new observations using the Woodbury matrix identity (p rank update). Let $\tilde{z}_a$ and α be defined as $\tilde{z}_a = Q(\tilde{B}_a^T \tilde{y} + \lambda \hat{z}_a - \mu v_a)$ and $\alpha = t^2 - 1$. The singular value decomposition of $B_a^{\text{new}} Q B_a^{\text{new}^T}$ is written $B_a^{\text{new}} Q B_a^{\text{new}^T} = \Gamma^T \Sigma \Gamma$. The *rotated* data is defined by $\overline{B}^{\text{new}} = \Gamma B^{\text{new}}$ and $\overline{y}^{\text{new}} = \Gamma y^{\text{new}}$. Similarly, the rotated error is $\overline{E} = \overline{B}_a^{\text{new}} \tilde{z}_a - \overline{y}^{\text{new}}$ and U is defined as $U = Q \overline{B}_a^{\text{new}^T}$.

Proposition 1 (Solution path to add p observations [22])**.** *For $t \in [0, t^*)$, $z_a(\cdot)$ is continuous in t and given by*

$$z_a(t) = \tilde{z}_a - (t^2 - 1) U \left(I + (t^2 - 1)\Sigma \right)^{-1} \overline{E}. \tag{7}$$

Let t^0 be the smallest[3] $t \in [0, 1]$ such that a coordinate of $z_a(t)$ equals zero, t^+ (resp. t^-) the smallest[3] $t \in [0, 1]$ which sets a coordinate of $w_{na}(t)$ to 1 (resp. to -1). The transition point t^* is defined as $t^* = \min(t^0, t^+, t^-)$ and can be computed by solving p-degree polynomial equations on a bounded interval.

Proof. See [22]. The computation of t^0, t^+ and t^- is given by Lemma 1 and 2. □

Let $U_{i,j}$ denote the element of U on line i and column j and by U_i the i^{th} line of U, σ_i is the i^{th} singular value of Σ and $\overline{E}_i$ is the i^{th} coordinate of $\overline{E}$.

Lemma 1 (Computation of t^0 [22])**.** *Let $t^0_{a_i}$ be the smallest value of $t \in [0, 1]$ which sets the i^{th} coordinate of z_a (in the regularized indices) to zero. It is given by $t^0_{a_i} = \sqrt{\alpha^0_{a_i} + 1}$ where $\alpha^0_{a_i}$ is the smallest real valued solution in the interval [-1,0] of the following p degree polynomial equation in α:*

$$0 = \tilde{z}_{a_i} \prod_{l=1}^{p} (1 + \alpha \sigma_l) - \alpha \sum_{j=1}^{p} U_{i,j} \bar{E}_j \prod_{l \neq j} (1 + \alpha \sigma_l).$$

If the polynomial equation does not have real valued solutions in [-1,0], set $t^0_{a_i} = 1$. It follows that t^0 is the smallest value of $t^0_{a_i}$ in the interval $[0, 1]$.

Proof. See [22]. □

Let c_i denote the i^{th} column of $\tilde{B}_{na}$, d_i denote the i^{th} row of $\overline{B}_{na}^{\text{new}}$ and $d_{i,j}$ denote the element of $\overline{B}_{na}^{\text{new}}$ on the i^{th} row and j^{th} column. Let f_i be the i^{th} element of $\tilde{B}_{na}^T \tilde{e} - \lambda \hat{z}_{na}$ and let $\tilde{e}$ be defined as $\tilde{e} = \tilde{B}_a \tilde{z}_a - \tilde{y}$.

Lemma 2 (Computation of t^+ and t^-)**.** *The smallest value of t that sets the i^{th} coordinate of w_{na} to 1 (resp. to -1) is denoted $t^+_{na_i}$ (resp. $t^-_{na_i}$). It is given by $t^+_{na_i} = \sqrt{\alpha^+_{na_i} + 1}$ (resp. $t^-_{na_i} = \sqrt{\alpha^-_{na_i} + 1}$) where $\alpha^+_{na_i}$ (resp. $\alpha^+_{na_i}$) is the smallest real valued solution in the interval [-1,0] of the p degree polynomial equation in α^+ (resp. in α^-):*

$$(-\mu - f_i) \prod_{l=1}^{p} (1 + \alpha^+ \sigma_l) = \alpha^+ \sum_{j=1}^{p} \overline{E}_j (d_{i,j} - c_i^T \tilde{B}_a U_j) \prod_{l \neq j} (1 + \alpha^+ \sigma_l),$$

$$(\mu - f_i) \prod_{l=1}^{p} (1 + \alpha^- \sigma_l) = \alpha^- \sum_{j=1}^{p} \overline{E}_j (d_{i,j} - c_i^T \tilde{B}_a U_j) \prod_{l \neq j} (1 + \alpha^- \sigma_l).$$

[3] If no such t exists, set t^0 (resp. t^+ and t^-) to 1.

If the polynomial equation does not have real valued solutions in [-1,0], set $t^+_{na_i} = 1$ *(resp.* $t^-_{na_i} = 1$*). It follows that* t^+ *(resp.* t^-*) is the smallest value of* $t^+_{na_i}$ *(resp.* $t^-_{na_i}$*) in the interval* $[0, 1]$.

Proof. See [22]. □

Lemma 3 (Update of the active set)**.** *When* t *reaches a transition point, the active set and signs of the regularized indices are updated as follows: (i) if* $t^* = t^0$*, remove the corresponding coordinate from the active set, (ii) if* $t^* = t^+$ *(resp.* $t^* = t^-$*), add the coordinate to the active set and set its sign to positive (resp. to negative).*

Proof. See [22]. □

Algorithm 1 updates the solution when t varies from $t = 0$ to $t = 1$. The same algorithm is relevant to remove p observations by finding the transition points as t decreases from 1 to 0.

Algorithm 1 Update of the solution to add p observations

Initialize the active set a, non active set na and signs of the regularized indices v_a.
$t = 0$
while $t < 1$ **do**
  Compute t^0, t^+ and t^- as the smallest value of $t^0_{a,i}$, $t^+_{na,i}$ and $t^-_{na,i}$ in $(t, 1]$ (Lemma 1-2).
  $t = \min(t^0, t^+, t^-)$
  if $t > 1$ **then**
    break;
  else if $t = t^0$ **then**
    Add the corresponding index to na and remove it from a and v_a.
  else if $t = t^+$ **then**
    Add the corresponding index to a and remove it from na, set its sign to positive and add it to v_a.
  else
    Add the corresponding index to a and remove it from na, set its sign to negative and add it to v_a
  end if
  Update the matrix Q to account for the updated active set (rank 1 update).
end while
Compute the solution at $t = 1$.

2.2. **Update the regularization parameter.** The computation of the regularization path is detailed in [12] and in [38] for the Elastic Net. As done in the previous step of the algorithm (add p observations), it is necessary to define the *non-regularized indices* and set $v_{a_i} = 0$ for these indices to solve (4). The end of the section details how the algorithms developed in [12] and [38] are adapted to solve (4).

At $\mu = \mu_n$, the solution $z(0, \mu_n)$ is known, and so are the active set, non active set and signs of the coordinates of z which are in the active set. The optimality

conditions read

$$B_a^T(B_a z_a(\mu) - y) + \mu v_a(\mu) + \lambda(z_a(\mu) - \hat{z}_a) = 0, \tag{8}$$

$$B_{na}^T(B_a\, z_a(\mu) - y) + \mu w_{na}(\mu) - \lambda \hat{z}_{na} = 0. \tag{9}$$

where $v_a(\mu)$ is the partial derivative of the l_1 norm for the indices in the set a with entries $v_{a_i}(\mu) = \text{sgn}(z_{a_i}(\mu))$ for the regularized indices, $v_{a_i} = 0$ for the non regularized indices and $w_{na}(\mu)$ is a vector with coordinates in $[-1, 1]$. Let Q be defined by $Q = (B_a^T B_a + \lambda I_{|a|})^{-1}$.

Proposition 2 (Linear dependence in μ)**.** *There exists a transition point $\mu^* \in [\mu_n,\, \mu_{n+1}]$ such that the active set, non active set and signs of the regularized indices of the solution remain constant for $\mu \in [\mu_n, \mu^*)$. Let μ^0 be the smallest*[4] *$\mu \in [\mu_n,\, \mu_{n+1}]$ such that a coordinate of $z_a(\mu)$ equals zero, μ^+ (resp. μ^-) the smallest*[4] *$\mu \in [\mu_n, \mu_{n+1}]$ which sets a coordinate of $w_{na}(\mu)$ to 1 (resp. to -1). The transition point μ^* is defined as $\mu^* = \min(\mu^0, \mu^+, \mu^-)$. On the interval $[\mu_n, \mu^*)$, v_{a_i} denotes the (constant) sign of $z_{a_i}(\mu)$ for the regularized indices. The estimate $z_a(\mu)$ is affine in μ and given by*

$$z_a(\mu) = Q(B_a^T y + \lambda \hat{z}_a) - \mu Q v_a. \tag{10}$$

Proof. See [22]. □

As long as the active set and signs of the regularized indices remain constant, the expression of $z_a(\mu)$ is given by (10).

Lemma 4 (Expression of μ^0)**.** *Let $\mu^0_{a_i}$ denote the value of μ that sets the i^{th} coordinate of z_a (in the regularized indices) to zero. Let $\mu^+_{na_i}$ (resp. $\mu^-_{na_i}$) denote the value of μ that sets the i^{th} coordinate of w_{na} to 1 (resp. to -1). We have*

$$\mu^0_{a_i} = [Q(B_a^T y + \lambda \hat{z}_a)]_i / [Q v_a]_i$$

$$\mu^+_{na_i} = \frac{\left[B_{na}^T\left((B_a Q B_a^T - I_n)y\right) + \lambda(B_{na}^T B_a Q \hat{z}_a - \hat{z}_{na})\right]_i}{-1 + [B_{na}^T B_a Q v_a]_i},$$

$$\mu^-_{na_i} = \frac{\left[B_{na}^T\left((B_a Q B_a^T - I_m)y\right) + \lambda(B_{na}^T B_a Q \hat{z}_a - \hat{z}_{na})\right]_i}{1 + [B_{na}^T B_a Q v_a]_i},$$

where $[V]_i$ denotes the i^{th} coordinate of generic vector V. The first possible transition point μ^0 (resp. μ^+ and μ^-) is the smallest value of $\mu^0_{a_i}$ (resp. $\mu^+_{na_i}$ and $\mu^-_{na_i}$) in the interval $[\mu_n, \mu_{n+1}]$, or μ_{n+1} if, for all indices, $\mu^0_{a_i} \notin [\mu_n, \mu_{n+1}]$ (resp. $\mu^+_{na_i} \notin [\mu_n, \mu_{n+1}]$ and $\mu^-_{na_i} \notin [\mu_n, \mu_{n+1}]$.

Proof. See [22]. □

Leveraging Proposition 2 and Lemma 4, Algorithm 2 updates the solution z when μ varies from $\mu = \mu_n$ to $\mu = \mu_{n+1}$. Note that the derivations assume that $\mu_n \leq \mu_{n+1}$. The same algorithm is relevant if $\mu_n \geq \mu_{n+1}$ by finding the transition point as the regularization parameter decreases (instead of increases).

Remark 1 (Leveraging the sparsity structure)**.** The matrix Q is efficiently updated when the active or non active set change or when observations are added/removed using low rank updates. The numerical implementation updates the Cholesky factorization of Q which provides better numerical stability to the algorithm than updating Q directly [19].

[4] If no such μ exists, set μ^0 (resp. μ^+ and μ^-) to μ_{n+1}.

Algorithm 2 Update of the solution as μ increases from μ_n to μ_{n+1}

Initialize the active set a, non active set na and sign of the regularized indices v_a.
$\mu = \mu_n$
while $\mu < \mu_{n+1}$ **do**
  Compute μ^0, μ^+ and μ^- as the smallest values of $\mu^0_{a,i}$, $\mu^+_{na,i}$ and $\mu^-_{na,i}$ in $(\mu, \mu_{n+1}]$ (Lemma 4).
  $\mu = \min(\mu^0, \mu^+, \mu^-)$
  if $\mu > \mu_{n+1}$ **then**
    break;
  else if $\mu = \mu^0$ **then**
    Add the corresponding index to na and remove it from a and v_a.
  else if $\mu = \mu^+$ **then**
    Add the corresponding index to a and remove it from na, set its sign to positive and add it to v_a.
  else
    Add the corresponding index to a and remove it from na, set its sign to negative and add it to v_a
  end if
  Update the matrix Q to account for the added (or removed) index in the active set (rank 1 update).
end while
Compute the solution at $\mu = \mu_{n+1}$

Remark 2 (Complexity)**.** The complexity of the algorithm depends on the number of transitions. The theoretical bound on the number of transitions is 3^k, where k is the number of rows of K_1. Indeed, each of the first k coordinates of z can be strictly positive, strictly negative or in the non active set. In practice, it is much smaller because successive estimates are expected to have a similar support. Experience with data suggests that the number of transition is linear in the problem size [35]. A theoretical analysis of the number of transitions is performed in [32]. The transition points are computed according to Lemma 1 and 2 when adding (or removing) observations and according to Lemma 4 when updating the regularization parameter μ.

3. **Recursive LASSO with varying reference parameter.** This section considers the linear regression problem introduced in (2). The problem encourages the vector $x^n - \bar{x}^n$ to be sparse. The reference $\bar{x}^n$ may change at each iteration. For example, the choice $\bar{x}^n = x^{n-1}$ leads to sparse variations of the estimate. The estimate is updated when observations are added (or removed), when the l_1 regularization parameter changes or when the reference parameter $\bar{x}^n$ changes. In order to update the solution from previous estimates, the algorithm computes a homotopy regularization path, as done in Section 2. After computing the solution x^n to Equation (2), p new observations $(y^{\text{new}}, A^{\text{new}}) \in \mathbb{R}^p \times \mathbb{R}^{p\times m}$, a new penalty coefficient μ_{n+1} and a new reference parameter $\bar{x}^{n+1}$ (*e.g.* $\bar{x}^{n+1} = x^n$) are received [5]. As for Section 2, an additional l_2 penalization is added to the objective function of

[5]Note that not all parameters are required to change at each iteration.

the LASSO to improve the estimation capabilities [38]. The homotopy algorithm is derived by introducing the following optimization problem:

$$x(t,u,\mu) = \underset{x\in\mathbb{R}^m}{\arg\min} \quad \frac{1}{2}\left\| \begin{pmatrix} A \\ tA^{\text{new}} \end{pmatrix} x - \begin{pmatrix} y \\ ty^{\text{new}} \end{pmatrix} \right\|_2^2 + \mu \left\| x - \Big((1-u)\bar{x}^n + u\bar{x}^{n+1}\Big) \right\|_1 + \frac{\lambda}{2}||x-\hat{x}||_2^2. \tag{11}$$

The definition of Equation (11) leads to $x(0,0,\mu_n) = x^n$ and $x(1,1,\mu_{n+1}) = x^{n+1}$. The section develops an algorithm that computes a path from x^n to x^{n+1} in three steps: (i) vary μ from μ_n to μ_{n+1} to change the weight of the l_1 regularization, (ii) vary t from 0 to 1 to add observations and (iii) vary u from 0 to 1 to update the reference parameter. Note that the different steps of the algorithm (variation of μ, t and u) do not need to be performed in a pre-specified order.

The change of the weight of the l_1 regularization and the variation of t from 0 to 1 are readily adapted from the computations of Section 2. The section succinctly presents the required changes for these steps and details the algorithm to update the reference parameter from $\bar{x}^n$ to $\bar{x}^{n+1}$ (increase u from 0 to 1).

3.1. **Update the regularization parameter and add observations.** During the update of the regularization parameter and the addition of observations, the parameter u remains constant. Assume without loss of generality that the variation of u is chosen to be performed last and thus $u = 0$ as the regularization parameter is updated and the observations added. If the variation of u has started before these steps occur, replace $\bar{x}^n$ by $(1-u)\bar{x}^n + u\bar{x}^{n+1}$ in the following derivations.

To leverage the algorithm developed in Section 2, it is convenient to introduce the following change of variables: $z = x - \bar{x}^n$, $y_r = y - A\bar{x}^n$, $y_r^{\text{new}} = y - A^{\text{new}}\bar{x}^n$ and $\hat{z} = \hat{x} - \bar{x}^n$. For notation consistency, the matrices A and A^{new} are denoted B and B^{new} respectively (same as for Section 2 with K being the identity matrix). With this notation, updating the regularization parameter (vary μ) and adding new observations (vary t) correspond to updating the solution of

$$\underset{z\in\mathbb{R}^m}{\textbf{minimize}} \ \frac{1}{2}\left\| \begin{pmatrix} B \\ tB^{\text{new}} \end{pmatrix} z - \begin{pmatrix} y_r \\ ty_r^{\text{new}} \end{pmatrix} \right\|_2^2 + \mu||I_m z||_1 + \lambda||z-\hat{z}||_2^2, \tag{12}$$

as μ varies from μ_n to μ_{n+1} and t from 0 to 1.

3.2. **Update the reference parameter.** The last step of the algorithm updates the reference parameter from $\bar{x}^n$ to $\bar{x}^{n+1}$. Let $x_r(u)$ be defined by $x_r(u) = x - [(1-u)\bar{x}^n + u\bar{x}^{n+1}]$. It represents the vector which is expected to be sparse because of the l_1-norm penalization. As done in the previous section, assume without loss of generality that the variation of u is chosen to be performed last. At this step of the algorithm, the regularization parameter has been updated and the new observations have been added. In particular, since the observations have been added, the matrix A and the vector y contain the recently added data.

Define $y_r = y - A\bar{x}^n$, $\Delta x = \bar{x}^n - \bar{x}^{n+1}$ and $Q = (A_a^T A_a + \lambda I)^{-1}$. Let c_j denote the vector defined by $c_j = A_j^T y_r + \lambda[\hat{x} - \bar{x}^n]_j$, where j represents the set of indices a or na. With this notation, $x_r(u)$ is the minimizer of the optimization problem

$$\underset{x_r\in\mathbb{R}^m}{\textbf{minimize}} \ \frac{1}{2}||Ax_r - y_r - u\Delta x||_2^2 + \mu_{n+1}||x_r||_1 + \frac{\lambda}{2}||x_r - (\hat{x} - \bar{x}^n) - u\Delta x||_2^2.$$

The optimality conditions read

$$(A_a^T A_a + \lambda I)x_{r,a}(u) - c_a + \mu v_a + u\left(A_a^T(A\Delta x) + \lambda(\Delta x)_a\right) = 0, \tag{13}$$

$$A_{na}^T A_a x_{r,a}(u) - c_{na} + \mu w_{na}(u) + u\left(A_{na}^T(A\Delta x) + \lambda(\Delta x)_{na}\right) = 0. \tag{14}$$

Proposition 3 (Linear dependence in u)**.** *There exists a transition point $u^* \in [0,1]$ such that the active set, non active set and signs of the regularized indices of the solution remain constant for $u \in [0,u^*)$. On this interval, v_{a_i} is the (constant) sign of $x_{r,a_i}(u)$. The estimate $x_{r,a}(u)$ is affine in u and given by $x_{r,a}(u) = \xi + u\chi$, with $\xi = Q(c_a - \mu v_a)$ and $\chi = Q\left(A_a^T(A\Delta x) + \lambda(\Delta x)_a\right)$.*

Proof. From the optimality conditions, it follows that the function $u \mapsto x_{r,a}(u)$ is affine as long as $u \mapsto v_a(u)$ is constant *i.e.* as long as the coordinates of $u \mapsto x_{r,a}(u)$ have constant signs and as long as the active set remains constant. Denote by u^0 the smallest value of $u \in [0,1]$ such that a coordinate of $x_{r,a}(u)$ equals zero in Equation (13). The signs of the entries of $x^{r,a}(u)$, and thus the value of $v_a(u)$, are constant on $[0,u^0]$. The (constant) value of $v_a(u)$ on this interval is denoted v_a. The optimality condition given in Equation (14) also shows that $u \mapsto w_{na}(u)$ is continuous. A coordinate of the non-active set joins the active set when the corresponding coordinate of $u \mapsto w_{na}(u)$ reaches one in absolute value. Let u^+ (resp. u^-) be the smallest value of $u \in [0,1]$ such that a coordinate of $w_{na}(u)$ equals 1 (resp. -1). The non-active set is constant on $[0, \min(u^+,u^-)]$. The active set and signs of the coordinates in the active set remain constant on the interval $[0,u^*]$ where $u^* = \min(u^?u^+, u^-)$. □

Lemma 5 (Expression of u^0)**.** *Let $u_{a_i}^0$ be the value of u that sets the i^{th} coordinate of $x_{r,a}(u)$ to zero. It is given by $u_{a_i}^0 = -\xi_i/\chi_i$. The first possible transition point u^0 is the smallest value of $u_{a_i}^0$ in the interval $[0,1]$, or 1 if, for all i, $u_{a_i}^0 \ni [0,1]$.*

Proof. The proof is derived from the expression of $x_{r,a}(u)$ given by Equation (13) for $u \in [0,u^*]$. □

Lemma 6 (Expression of u^+ and u^-)**.** *Let $u_{na_i}^+$ (resp. $u_{na_i}^+$) be the value of u that sets the i^{th} coordinate of $w_{na}(u)$ to 1 (resp. -1),* i.e. *the value of u for which the i^{th} coordinate of $x_{r,na}$ enters the active set and becomes positive (resp. negative). They are given by:*

$$u_{na_i}^+ = -\frac{A_{na_i}^T A_a \xi - c_{na_i} + \mu}{A_{na_i}^T(A_a\chi + A\Delta x - A_c(\Delta x)_c) + \lambda(\Delta x)_{na_i}},$$

$$u_{na_i}^- = -\frac{A_{na_i}^T A_a \xi - c_{na_i} - \mu}{A_{na_i}^T(A_a\chi + A\Delta x - A_c(\Delta x)_c) + \lambda(\Delta x)_{na_i}}.$$

The first possible transition point u^+ (resp. u^-) is the smallest value of $u_{na_i}^+$ (resp. $u_{na_i}^-$) in the interval $[0,1]$, or 1 if, for all i, $u_{na_i}^+ \ni [0,1]$ (resp. $u_{na_i}^- \ni [0,1]$).

Proof. The proof is derived from the optimality condition given in Equation (14) and the expression of $x_{r,a}(u)$ for $u \in [0,u^*]$. □

A transition occurs for the smallest $u^* \in [0,1]$ such that one component of $x_{r,na}$ enters the active set or one component of $x_{r,a}$ enters the non-active set. At $u = u^*$, update the active and non active sets and search for the next transition point until $u = 1$ and the update of the reference parameter is completed.

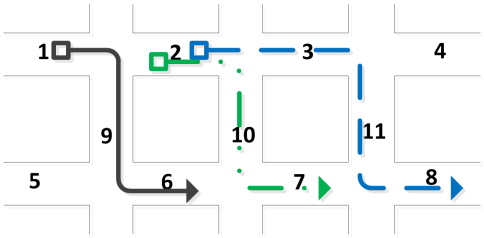

FIGURE 1. Example paths of three probe vehicles on a network. The network has eleven links. The path of a probe is represented as a vector $a_i \in [0,1]^{11}$ where the j^{th} coordinate of a_i represent the fraction of link j traveled by the probe. The path represented with a solid line is represented with a sparse vector with non zero coordinates 1, 6 and 9, respectively equal to 0.4, 0.7 and 1 considering that the probe traveled 40% of link 1 and 70% of link 6. The vector representing the dashed path has non zero coordinates 2, 3, 8 and 11, respectively equal to 0.3, 1, 0.8 and 1 considering that the probe traveled 30% of link 2 and 80% of link 8.

4. **Numerical results.** The potential of the algorithm is illustrated through applications to traffic estimation from sparsely sampled probe vehicles, flow estimation in tidal channels and text analysis of on-line news.

4.1. **Traffic estimation from sparsely sampled probe vehicles.** We first illustrate the algorithm for arterial traffic estimation. At the time when this article is written, traffic data on arterial networks is mainly provided from probe vehicles sending their location at a given sampling frequency (common sampling frequencies are around 1 minute). The proportion of sampled vehicles (penetration rate) rarely exceeds a few percent of the vehicles traveling on the network. Moreover, traffic signals cause important variation on the travel time experienced on a link of the network within very short periods of time (depending on whether the vehicle stopped at the signal or not), while the actual changes in traffic conditions have slower dynamics. The estimate x^n represents the average travel time on each link of the network at time t^n. To filter the travel time fluctuations due to traffic lights and detect when the level of congestion changes, we are interested in sparse variations of the travel time on each link. The estimate x^{n+1} is computed by solving equation (11) with $\bar{x}^n = x^{n-1}$. The algorithm is initialized using a previous estimate of the mean travel times given by least-squares regression. A historical mean value of travel times $\hat{x}$ is used to add a l_2 regularization term $||x - \hat{x}||$. At each estimation time, the regularization parameter is updated (from $|I_n|\mu_0$ to $|I_{n+1}|\mu_0$), the new data is added and the reference parameter is updated (from $\bar{x}^n = x^{n-1}$ to $\bar{x}^{n+1} = x^n$). We use data provided by a fleet of 500 probe vehicles which report their location every minute, representative of the data available in the *Mobile Millennium* system [4]. The estimation is performed in a subnetwork of San Francisco, CA with more than 800 links.

The duration between two successive location reports ξ_1 and ξ_2 is an observation of the travel time y_i on the path from ξ_1 to ξ_2. After using the map-matching and path-inference algorithm to reconstruct the path of each vehicle [23], each trajectory (path) is converted in a vector $a_i \in [0, 1]^m$, where m is the number of links in the network. The j^{th} coordinate of a_i, denoted $a_{i,j}$, is the fraction of the link traveled by the probe vehicle. It is computed as the distance traveled on the link divided by

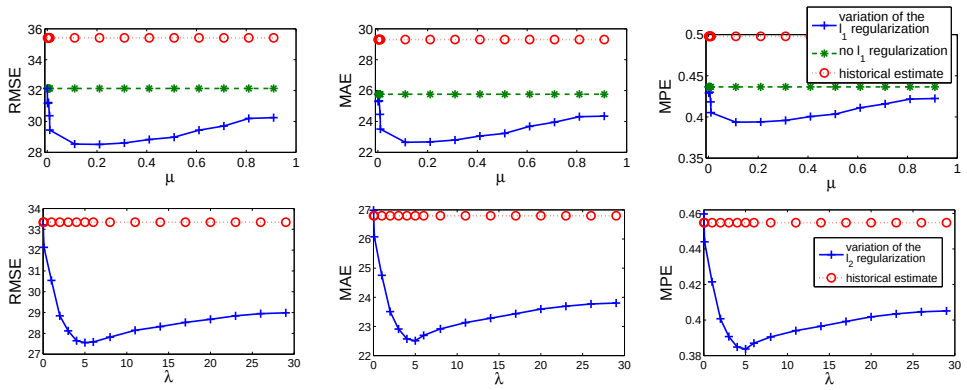

FIGURE 2. Variation of the error metrics in function of the regularization parameters for the l_1 and l_2 penalization when encouraging sparsity on the temporal variations of traffic conditions. Both the l_1 and l_2 regularizations improve the estimation accuracy and the regularization parameters can be chosen optimally. The three top figures represent the effect of the l_1 regularization for the estimation accuracy. The three bottom figures show the importance of the additional l_2 regularization introduced in Section 2 for the robustness of the estimation.

the length of the link[6]. In particular, $a_{i,j} = 0$ if the vehicle did not travel on link j and $a_{i,j} = 1$ if the vehicle fully traversed link j (see Figure 1).

The performance of the model is assessed using cross-validation, randomly splitting the observations sent by the probe vehicles between a training set and a validation set. After learning the travel time estimates on the training set, the validation set is used to compare the estimates to the travel time observations. The performance of the model is compared with a *baseline model*, which uses the historical value of the link travel times $\hat{x}$ as the estimate of the state. Three metrics quantify the quality of the estimation: the root mean squared error (RMSE), the mean absolute error (MAE) and the mean percentage error (MPE)[7]. Note that the variability of arterial travel times (due to traffic signals, pedestians, etc.) leads to important fluctuations of travel times. This inherent variability in the state of the system makes the estimation model robust with sparse variations, but is also responsible for relatively high values of the error metrics.

The numerical analysis assesses the performance of the model and quantifies the effect of the regularization parameters λ and μ_0. The first parameter penalizes solutions which are far (in the l_2-norm sense) from the historical estimate of travel times $\hat{x}$. The second parameter imposes sparsity on the variation of the estimate. The choice of these parameters leads to a compromise between (i) fitting the data, with risks of overfitting and lack of physical interpretation and (ii) putting too much weight on the regularization and not estimating accurately the current state of the system.

The results indicate that both the l_1 and the l_2 regularization (Figure 2) are important to improve the estimation capabilities. For a wide range of parameters,

[6]The coefficients $a_{i,j}$ can account for the fact that travel time on a fraction of the link does not vary proportionally with the distance traveled as vehicles are more likely to experience delays close to signalized intersections as demonstrated in [21].

[7]$\text{RMSE} = \sqrt{\frac{\sum_{o=1}^{O}(y_o - \hat{y}_o)^2}{O}}$, $\text{MAE} = \frac{\sum_{o=1}^{O}|y_o - \hat{y}_o|}{O}$, $\text{MPE} = \frac{1}{O}\sum_{o=1}^{O}\frac{|y_o - \hat{y}_o|}{y_o}$.

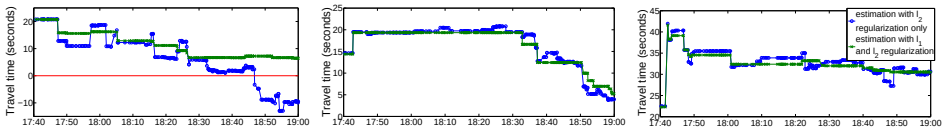

FIGURE 3. Qualitative evolution of the travel time estimates on different links of the network. The l_1 regularization provides more stable estimates that represent the dynamics of traffic more accurately and increase the physical interpretation. The left figure shows that the estimation with l_2 regularization leads to estimates that are not physically possible (negative travel times), while the estimate with l_1 regularization remains within feasible bounds. On all figures, the l_2 estimate is noisy while the additional l_1 regularization remains constant between each temporal transitions in traffic conditions.

the results are significantly better than the baseline model. The results also underline the importance of the additional l_2 regularization to improve the robustness of the algorithm. Figure 3 illustrates that in addition to improving the estimation capabilities, the algorithm produces results that are easier to interpret. Arterial traffic is highly variable and the variability often prevents the interpretation of the results. This model estimates the trends in travel times on the links of the network, while filtering the variability due to the signal dynamics.

4.2. **Water flow estimation in tidal channels.** In this section, we present an application of the LASSO to estimate water flow (discharge) in tidal channels. More precisely, having measurements of the flow at a given location in a channel with tidal forcing, the goal is to estimate the flow at any desired location in the channel. We first derive a transfer function representation of flow in the z domain using the linearized Saint-Venant equations. The derived transfer function corresponds to a *multi-input multi-output* (MIMO) system whose outputs are the available measurements and the inputs are the flow and stage at the location where an estimate is desired. The estimation problem is then posed as an input estimation problem. After parametrizing the inputs using the dominant tidal modes, we use recursive LASSO to estimate the unknown parameters, i.e. the mode amplitudes, recursively. The l_1-norm penalty that we consider in LASSO is the difference between the decision variables and the optimal solution at the previous time step. LASSO enforces a sparse variation in the estimated parameters and it essentially updates the most dominant modes at every time step.

While more traditional state estimation methods such as Kalman filtering could be used to estimate the flow everywhere throughout the channel, the proposed method is particularly useful when estimates of the flow are desired only at a specific location along the channel.

4.2.1. *Flow model: Linearized Saint-Venant equations.* The Saint-Venant model is among the most common models used for modeling the flow in open channels and irrigation systems [7], [8]. In the one dimensional case, Saint-Venant equations are two coupled first-order hyperbolic *partial differential equations* (PDE) derived from conservation of mass and momentum. In cases where a linear model is needed, these equations are linearized around a steady state (backwater curve) [30], [28].

The linearized Saint-Venant equations are as follows:

$$T_0(l)\frac{\partial h}{\partial t} + \frac{\partial q}{\partial l} = 0 \tag{15}$$

$$\frac{\partial q}{\partial t} + 2V_0(l)\frac{\partial q}{\partial l} - \beta_0(l)q + (C_0(l)^2 - V_0(l)^2)T_0(l)\frac{\partial h}{\partial l} - \gamma_0(l)h = 0 \tag{16}$$

with

$$\gamma_0 = V_0^2\frac{dT_0}{dl} + gT_0\left(\kappa S_{f0} + S_b - (1+2F_0^2)\frac{dH_0}{dl}\right) \tag{17}$$

$$\beta_0 = \frac{2g}{V_0}\left(F_0^2\frac{dH_0}{dl} - S_{f0}\right) \tag{18}$$

$$\kappa = 7/3 - 4A_0/(3T_0P_0)\partial P_0/\partial H \tag{19}$$

for $(l,t) \in (0,L)\times\Re^+$, where L is the river reach (m), $Q_0(l)$ is the steady-state discharge or flow ($^{m^3}/_{\mathrm{s}}$), $H_0(l)$ is the steady-state stage or water-depth (m), $V_0(l) = Q_0(l)/A_0(l)$ is the steady-state average velocity across cross section $A_0(l)$, $q(l,t)$ and $h(l,t)$ are the deviation of flow and stage from the steady state, $T(l)$ is the free surface width (m), $D = A/T$ is the hydraulic depth m, $C_0 = \sqrt{gH_0}$ is the wave celerity, $F_0 = V_0/C_0$ is the Froude number, $S_{f0}(l)$ is the steady-state friction slope ($^{\mathrm{m}}/_{\mathrm{m}}$), S_b is the bed slope ($^{\mathrm{m}}/_{\mathrm{m}}$), g is the gravitational acceleration ($^{\mathrm{m}}/_{s^2}$).

The friction slope is empirically modeled by the Manning-Strickler's formula [29]:

$$S_{f0} = \frac{m^2Q_0^2P_0^{4/3}}{A_0^{10/3}} \tag{20}$$

with $P_0(l)$ being the wetted perimeter and m the Manning's roughness coefficient ($sm^{-1/3}$).

4.2.2. *Transfer function derivation.* To obtain the *open-loop transfer matrix* of the system, we follow the same method as introduced in [30]. Applying the z-transform to equations (16) and (15) and rearranging terms, we obtain the following ordinary differential equation

$$\frac{d}{dl}\begin{pmatrix} q_z(l) \\ h_z(l)\end{pmatrix} = \mathcal{A}_z(l)\begin{pmatrix} q_z(l) \\ h_z(l)\end{pmatrix} \tag{21}$$

with

$$\mathcal{A}_z(l) = \begin{pmatrix} 0 & -T_0(l)(\frac{z-1}{\Delta tz}) \\ \dfrac{-(\frac{z-1}{\Delta tz}) + \beta_0(l)}{T_0(l)(C_0(l)^2 - V_0(l)^2)} & \dfrac{2V_0(l)T_0(l)(\frac{z-1}{\Delta tz}) + \gamma_0(l)}{T_0(l)(C_0(l)^2 - V_0(l)^2)}\end{pmatrix} \tag{22}$$

Defining $\zeta(l) = (q_z(l), h_z(l))^T$, the differential equation (21) has a solution of the form

$$\zeta(l) = \Gamma_z(l,0)\zeta_0 \tag{23}$$

For the case of uniform flow, $\mathcal{A}_z$ does not depend on l and consequently the solution to the differential equation can be calculated analytically and we will have

$$\Gamma_z(l,0) = e^{\mathcal{A}_z l} \tag{24}$$

To solve the differential equation for a general case, the method introduced in [30] can be used. Using this method, the interval $[0, l]$ is divided to smaller intervals $0 = l_0 < l_1 < \cdots < l_n = l, \quad l_{k+1} = l_k + L_k$ over which the flow can be approximated by uniform flow and after solving the differential equation over the small intervals, the overall transfer matrix is obtained by multiplying the individual transfer matrices, i.e. we can write

$$\Gamma_z(l, 0) = \prod_{k=1}^{n} e^{\mathcal{A}_z l_k} \tag{25}$$

Approximating the matrix exponentials with the first m terms, we have

$$e^{\mathcal{A}_z l_k} = I + (\mathcal{A}_z l_k) + \frac{1}{2!}(\mathcal{A}_z l_k)^2 + \cdots + \frac{1}{m!}(\mathcal{A}_z l_k)^m \tag{26}$$

This will result in a transfer matrix $\Gamma_z(l, 0)$ whose entries are polynomials of degree nm in z. This transfer matrix relates the upstream discharge and stage with the discharge and stage at any location along the channel.

With the location at which estimates of discharge is desired and the locations of the available measurements fixed, we can carry out the same procedure as above to obtain transfer matrices between the desired discharge, q_d, and each measurement, y_i.

4.2.3. *Estimation set-up.* In channels with tidal forcing, the flow and stage can be considered as the superposition of the dominant modes of the tides and accordingly q_d and h_d can be parameterized as follows

$$q_d(k) = a_0 + \sum_{i=1}^{N_{\text{modes}}} a_i cos(w_i k) + b_i sin(w_i k) \tag{27}$$

$$h_d(k) = c_0 + \sum_{i=1}^{N_{\text{modes}}} c_i cos(w_i k) + d_i sin(w_i k) \tag{28}$$

where N_{modes} is the number of dominant modes considered.

With the above parametrization of Q_d and H_d, the estimation problem boils down to estimation of the coefficients a_i, b_i, c_i, d_i for $i = 1, \cdots, N_{\text{modes}}$.

Let us define

$$u(k) = (q_d(k), h_d(k))^T \tag{29}$$

$$x_q = (a_0, a_1, \cdots, a_{Nmodes}, b_1, \cdots, b_{Nmodes})^T \tag{30}$$

$$x_h = (c_0, c_1, \cdots, c_{Nmodes}, d_1, \cdots, d_{Nmodes})^T \tag{31}$$

$$C(k) = (1, \cos(w_1 k), \cdots, \cos(w_{N_{\text{modes}}} k), \sin(w_1 k), \cdots, \sin(w_{N_{\text{modes}}} k)) \tag{32}$$

and let $y(k) = (y_1(k), \cdots, y_p(k))^T$ be the vector of deviation of p available measurements from their corresponding steady state at time step k. We can now write

$$\begin{aligned} y_i(k) =& q_d * g_i^q + h_d * g_i^h + e(k) \\ =& \sum_{j=1}^{mn} C(k-j) x_q g_i^q(j) + C(k-j) x_h g_i^h(j) + e(k) \end{aligned} \tag{33}$$

where $\{g_i^q(j)\}_{j=1}^{mn}$ and $\{g_i^h(j)\}_{j=1}^{mn}$ are the impulse responses of $G_i^q(z)$ and $G_i^h(z)$, respectively, and $e(k)$ represents the error and $*$ represents convolution.

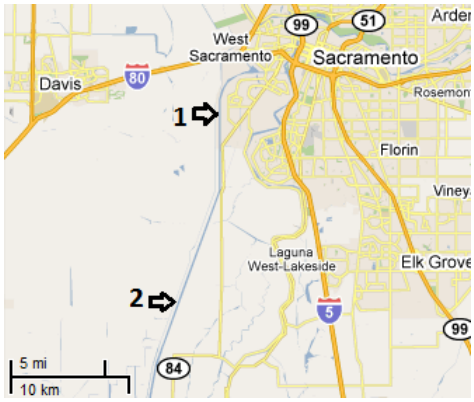

FIGURE 4. Channel used for implementation.

We can write equation (33) in compact form as follows

$$y_i(k) = A_i(k)x + e(k) \tag{34}$$

where

$$A_i(k) = \sum_{j=1}^{mn} \begin{bmatrix} C(k-j)g_i^q(j) & C(k-j)g_i^h(j) \end{bmatrix} \tag{35}$$

We formulate the estimation problem as the following optimization problem

$$\hat{x}^K = \arg\min_x \parallel Ax - y \parallel_2^2 + \mu_K \parallel x - \hat{x}^{K-1} \parallel_1 \tag{36}$$

where

$$A = [A_1(1)^T, \cdots, A_p(1)^T, \cdots, A_1(K)^T, \cdots, A_p(K)^T]^T \tag{37}$$

$$y = (y_1(1), \cdots, y_p(1), \cdots, y_1(K), \cdots, y_p(K))^T \tag{38}$$

The l_1-norm penalty enforces the variations of the estimates to be sparse. In other words, at each time step, the amplitudes corresponding to the more significant modes are updated.

4.2.4. *Implementation.* We implement the method on a 23.4 km long channel in Sacramento-San Joaquin Delta in northern California which is a complex network of over 1150 km of tidally influenced channels and sloughs which cover 738,000 acres of land. The Delta is of great significance in the state of California as it is the main source of drinking water for more than 20 million Californians and it is the source of irrigation of most of California's farmland. The channel chosen for the implementation is located on the southern side of Sacramento as shown in Figure 4.

The *Delta Simulation Model II (DSM2)* is used as the flow model to obtain the measurements of the flow used for performing the estimation and also to evaluate the quality of the estimates. DSM2 is a one-dimensional mathematical model of the flow in Sacramento-San Joaquin Delta which has been developed in the California Department of Water Resources (DWR). DSM2 uses measurements from USGS sensors as boundary conditions and provides discharge and stage at any location within the Delta. More detailed information about DSM2 can be found in [1].

To perform the estimation, we run DSM2 based on historical data starting August 10, 2006 until August 12, 2006. We consider a case in which estimations of the

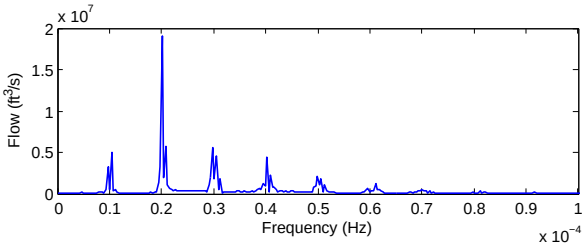

FIGURE 5. The power spectrum of the downstream discharge for the period of study.

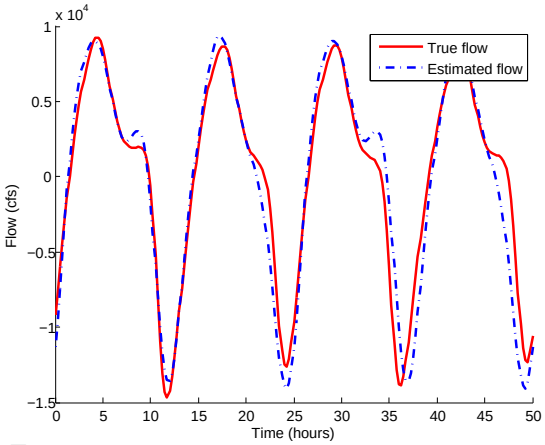

FIGURE 6. The estimated and true flow at location 1.

discharge at location 1 is desired when the flow measurements at location 2 are available. To obtain a parametrization of discharge and stage, we perform a spectral analysis of the downstream flow and we use the first eight dominant modes to parametrize the discharge. Figure 5 shows the power spectrum of the downstream discharge for the month of July 2006. We perform the estimation for 200 time steps with temporal step size of 15 minutes. Figure 6 shows the estimated flow at location 1 and the ground truth, i.e. the flow obtained from DSM2.

4.3. **Statistical analysis of news.** In this application [18], we use the LASSO problem to find an image of a specific topic such as image of countries (Greece, Japan, India etc.) in the news media. We represent the image by k uni-gram words chosen from the dictionary of all used words (around $130,000$) as shown in figure 7. Usually k is between 10 and 50 (in figure 7 we take $k = 14$) so the list of words can be manageable by a human reader. We obtain such list by solving a LASSO problem of the form

$$\min_{x \in \mathbb{R}^m} (1/2) \|Ax - y\|_2^2 + \mu \|x\|_1 ,$$

where A and y are the problem data, and μ is the regularization parameter. Each row of A corresponds to one document, like a headline and/or the first paragraph of a news article. We index all the words in the dictionary from 1 to m, where m is the total number of words used in the entire news corpus. We construct each row

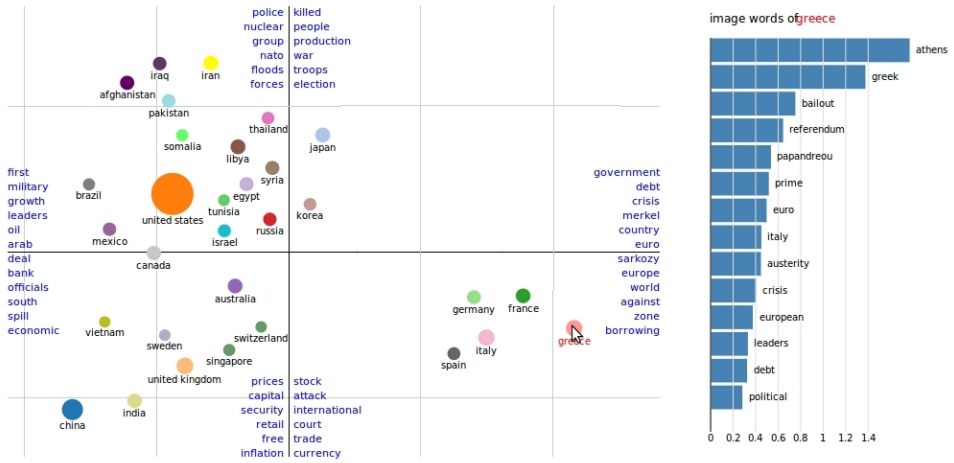

FIGURE 7. The plot shows the associations amongst countries using sparse *Principal Component Analysis* (PCA). The size of the ball of each country is proportional to the frequency of its occurrence in the text corpus. For each country, (in this figure we show Greece) we compute the image words using LASSO as shown in the top-right corner of the figure. The image words are considered "signatures" or unique to the country of interest. The data sources are taken from Headline news from wall Street Journal, Washington Post, Associated Press, Financial Times, Reuters, BBC, USA Today, CNN, CNBC, CBA, Vancouver Sun, and The Australian between October 01, 2011 and December 03, 2011. In total, there are 540,759 news feeds (observations) and 133,512 words used (features).

of A by assigning the number of times the word appears or the words frequencies in each document to the index of each vocabulary term. For example, if document i mentions the word "China" 12 times, and the index of China in our dictionary is j, then we assign the j^{th} element of row i the value $A(i, j) = 12$. We construct the label vector y based on the topic we are interested in finding its image. For example, if we are interested in finding the words that are associated with Greece, we assign to the i^{th} element of y the frequency of the term Greece in document i, i.e. if Greece was mentioned 5 times in the i^{th} document then $y(i) = 5$. By this construction we have $A \in \mathbb{R}^{n \times m}$, and $y \in \mathbb{R}^n$. The LASSO problem can be thought of as a surrogate that allows us to predict the frequency of our topic-word by using only the frequencies of k other words in the entire dictionary. This LASSO regression tells us that our topic-word can be associated to those k words, as these words are good predictors of the appearance of our topic-word. The number k of words representing our topic are obtained by controlling the regularization parameter, which in turn effects the number of non-zeros in the solution. Once the LASSO problem is solved we display the results as shown in the top-right corner of figure 7.

We use the on-line LASSO algorithm described in equation (2) with $\bar{x}^n = 0$ and $\lambda = 0$ to update the LASSO solution when a news headline becomes available. We typically assign $\mu = 0.3\mu_{\max}$ with $\mu_{\max} = ||A^T y||_\infty$. Table 1 summarizes the LASSO results for the countries United States, China, Egypt, Japan, Iraq, in addition to Greece during the period between October and December of 2011. The list of words for each country summarizes the issues that each country was dealing with during

Country:	United States	China	Egypt	Japan	Iraq	Greece
word 1:	iraq	beijing	mubarak	tokyo	iraqi	athens
word 2:	united	chinese	cairo	fukushima	troops	greek
word 3:	american	yuan	rulers	tsunami	baghdad	bailout
word 4:	obama	trade	brotherhood	yen	withdrawal	referendum
word 5:	china	growth	egyptian	japanese	kurdish	papandreou
word 6:	pakistan	currency	military	olympus	war	prime
word 7:	iran	us	elections	noda	veteran	euro
word 8:	military	shanghai	clashes	reactor	biden	italy
word 9:	clinton	manufacturing	civilian	plant	us	austerity
word 10:	reuters	wal-mart	parlimentary	nuclear	militar	crisis
word 11:	economy	inflation	israel	toyota	end	european
word 12:	stocks	hong kong	protesters	intervention	iran	leaders

TABLE 1. Summary of the LASSO results for the countries: United States, China, Egypt, Japan, Iraq, and Greece during the period between October and December of 2011.

that time period. For example, the exit of US troops from Iraq had effected the LASSO solution of both countries by selecting the terms: troops, withdrawal, end, war, veteran and others. The aftermath of Japan's tsunami also had its effect on the words selected by the LASSO like reactor, plan, nuclear, and Fukushima. The Arab-spring and political changes in Egypt were also communicated by the LASSO solution through the words: elections, civilians, clashes, protesters, and military. China's image words selected by the LASSO were more concerned with the economy like trade, growth, currency, manufacturing, and inflation. Greek's image words were mostly related to the bailout and the economic crisis is facing.

The construction of the matrix A affects the LASSO solution shown in table 1. Better results can be obtained by scaling this matrix, for example we can discourage the word "greek" from appearing as an image word of Greece by multiplying the column or feature corresponding to "greek" by a small number. This method can also be used to discourage other uninformative words (*e.g.* articles, pronouns) from appearing in the final LASSO solution. Using the on-line LASSO algorithm suits our recursive nature of updates obtained from new headlines, regardless of the scaling needed to construct the problem data. Constructing the problem data with proper scaling and defining *uninformative* words for the purpose of improving the results of the LASSO is out of the scope of this article and a subject of future research.

5. **Conclusion and discussion.** The article extends existing online-algorithm to update the solution of linear regression problems with a large class of l_1 and l_2 regularizations as new observations become available. The l_1-norm regularization improves the estimation capabilities and the interpretability of the results by exhibiting and exploiting the underlying sparsity structure of the problem. The additional l_2-norm regularization increases the robustness of the estimator and limits numerical issues. The algorithm provides the ability to (i) impose sparsity on a linear function of the estimate, (ii) update the solution online by computing a homotopy as new measurements become available (or as old measurements become obsolete) and (iii) impose sparsity on the variations of the state with respect to a reference parameter which can be updated at any time, for example to impose sparsity on successive estimates.

The homotopy algorithm leverages the sparsity of the solution to reduce the computational complexity and is thus particularly efficient when the solution is

sparse. The computational costs at each transition point is kept low by updating the matrix inverses with low-rank updates. The number of transition points and active indices varies with the parameter μ. As μ increases, the number of transition points and active indices decreases, improving the computational efficiency of the algorithm.

The generality of this algorithm is illustrated through its application for diverse real-world problems. In particular, we apply this generalized LASSO algorithm to real-time traffic estimation from streaming probe vehicle data in large urban networks, providing significant improvement of the estimation capabilities and an automatic detection of congestion changes across the network. We also applied the algorithm to estimate flow in a tidal channel. Given flow measurements at a location in the channel, we used the linearized Saint-Venant equations to obtain the transfer matrix between the flow at any desired location and the measurements. After parametrizing the flow considering the dominant tidal modes, we applied LASSO to estimate the unknown parameters. In the statistical analysis of online-news application, the LASSO algorithm was used to uncover statistical associations of words in a text corpus obtained from headlines of news articles. We presented the formulation of the LASSO problem, construction of the data matrices and illustrated how the online-homotopy algorithm is suited for this application.

REFERENCES

[1] J. Anderson and M. Mierzwa, An introduction to the Delta Simulation Model II (DSM2) for simulation of hydrodynamics and water quality of the Sacramento-San Joaquin Delta, in *Office of State Water Project Planning, California Department of Water Resources*, 2002.

[2] F. Bach, R. Jenatton, J. Mairal and G. Obozinski, Convex optimization with sparsity-inducing norms, in *Optimization for Machine Learning*, 2011, 19–53.

[3] R. Baraniuk, Compressive sensing, *IEEE Signal. Proc. Mag.*, **24** (2007), 118.

[4] A. Bayen, J. Butler and A. Patire, et al., *Mobile Millennium*, Final Report, Tech. rep., University of California, Berkeley, UCB-ITS-CWP-2011-6, 2011.

[5] D. Bertsekas, Incremental gradient, subgradient, and proximal methods for convex optimization: A survey, *Optimization for Machine Learning*, (2011), 85 pp.

[6] E. Candès, Compressive sampling, *Congress of Mathematicians*, **3** (2006), 1433–1452.

[7] V. Chow, *Open-channel Hydraulics*, McGraw-Hill Book Company, New York, NY, 1988.

[8] J. Cunge, F. Holly and A. Verwey, *Practical Aspects of Computational River Hydraulics*, Pitman, 1980.

[9] I. Daubechies, M. Defrise and C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Communications on Pure and Applied Mathematics*, **57** (2004), 1413–1457.

[10] Y. Dodge, *Statistical Data Analysis Based on the l_1-Norm and Related Methods*, Birkhäuser Verlag, Basel, 2002.

[11] I. Drori and D. Donoho, Solution of l_1 minimization problems by LARS/homotopy methods, in *International Conference on Acoustics, Speech and Signal Processing*, Vol. 3, IEEE, 2006.

[12] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, Least angle regression, *Ann. Statist.*, **32** (2004), 407–451.

[13] M. Figueiredo and R. Nowak, A bound optimization approach to wavelet-based image deconvolution, in *International Conference on Image Processing*, Vol. 2, IEEE, 2005.

[14] M. Figueiredo, R. Nowak and S. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE Journal of selected topics in signal processing*, **1** (2008), 586–597.

[15] J. Friedman, T. Hastie, H. Höfling and R. Tibshirani, Pathwise coordinate optimization, *Ann. Appl. Stat.*, **1** (2007), 302–332.

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

[17] P. Garrigues and L. El Ghaoui, An homotopy algorithm for the Lasso with online observations, in *Neural Information Processing Systems*, **21**, 2008.

[18] B. Gawalt, J. Jia, L. Miratrix, L. El Ghaoui, B. Yu and S. Clavier, Discovering word associations in news media via feature selection and sparse classification, in *Proceedings of the International Conference on Multimedia Information Retrieval*, ACM, New York, NY, 2010, 211–220.
[19] G. Golub and C. Van Loan, *Matrix Computations*, Third edition, Johns Hopkins University Press, Baltimore, MD, 1996.
[20] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *The Journal of Machine Learning Research*, **3** (2003), 1157–1182.
[21] A. Hofleitner, R. Herring, P. Abbeel and A. Bayen, Learning the dynamics of arterial traffic from probe data using a Dynamic Bayesian Network, *IEEE Transactions on Intelligent Transportation Systems*, **13** (2012), 1679–1693.
[22] A. Hofleitner, T. Rabbani, L. El Ghaoui and A. Bayen, Online homotopy algorithm for a generalization of the LASSO, *IEEE Transactions on Automatic Control*, **58** (2013), 3175–3179.
[23] T. Hunter, T. Moldovan, M. Zaharia, S. Merzgui, J. Ma, M. J. Franklin, P. Abbeel and A. M. Bayen, Scaling the Mobile Millennium system in the cloud, in *Proceedings of the 2nd ACM Symposium on Cloud Computing*, Article No. 28, ACM, New York, NY, 2011, 1–8.
[24] S. Kim, K. Koh, S. Boyd and D. Gorinevsky, l_1 trend filtering, *SIAM Rev.*, **51** (2009), 339–360.
[25] S. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, An Interior-Point Method for Large-Scale l_1-Regularized Least Squares, *IEEE Journal of Selected Topics in Signal Processing*, **1** (2008), 606–617.
[26] K. Knight and W. Fu, Asymptotics for lasso-type estimators, *Annals of Statistics*, **28** (2000), 1356–1378.
[27] H. Lee, A. Battle, R. Raina and A. Ng, Efficient sparse coding algorithms, *Advances in Neural Information Processing Systems*, **19** (2007), 801 pp.
[28] X. Litrico and V. Fromion, Boundary control of linearized Saint-Venant equations oscillating modes, *Automatica*, **42** (2006), 967–972.
[29] X. Litrico and V. Fromion, *Modeling and Control of Hydrosystems*, Springer, 2009.
[30] X. Litrico and V. Fromion, Frequency modeling of open-channel flow, *ASCE Journal of Journal of Hydraulic Engineering*, **130** (2004), 806–815.
[31] I. Loris, On the performance of algorithms for the minimization of l_1-penalized functionals, *Inverse Problems*, **25** (2009), 035008, 16 pp.
[32] J. Mairal and B. Yu, Complexity analysis of the Lasso regularization path, in *International Conference on Machine Learning*, 2012.
[33] A. Ng, Feature selection, l_1 vs. l_2 regularization, and rotational invariance, in *21st International Conference on Machine learning*, ACM, 2004, 78 pp.
[34] M. Osborne, B. Presnell and B. Turlach, A new approach to variable selection in least squares problems, *IMA journal of numerical analysis*, **20** (2000), 389–403.
[35] S. Rosset and J. Zhu, Piecewise linear regularized solution paths, *The Annals of Statistics*, **35** (2007), 1012–1030.
[36] M. Salman Asif and J. Romberg, Dynamic Updating for l_1 regularization, *IEEE Journal of Selected Topics in Signal Processing*, **4** (2010), 421–434.
[37] R. Tibshirani, Regression shrinkage and selection via the Lasso, *Journal of the Royal Statistical Society. Series B (Methodological)*, **58** (1996), 267–288.
[38] H. Zou and T. Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67** (2005), 301–320.

Received May 2013; revised August 2013.

E-mail address: aude.hofleitner@polytechnique.edu
E-mail address: trabbani@berkeley.edu
E-mail address: rafiee@berkeley.edu
E-mail address: elghaoui@berkeley.edu
E-mail address: bayen@berkeley.edu