Fast interior-point inference in high-dimensional sparse, penalized state-space models

Eftychios A. Pnevmatikakis
Columbia University
eftychios@stat.columbia.edu

Liam Paninski
Columbia University
liam@stat.columbia.edu

Abstract

We present an algorithm for fast posterior inference in penalized high-dimensional state-space models, suitable in the case where a few measurements are taken in each time step. We assume that the state prior and observation likelihoods are log-concave and have a special structure that allows fast matrix-vector operations. We derive a second-order algorithm for computing the maximum a posteriori state path estimate, where the cost per iteration scales linearly both in time and memory. This is done by computing an approximate Newton direction using an efficient forward-backward scheme based on a sequence of low rank updates. We formalize the conditions under which our algorithm is applicable and prove its stability and convergence. We show that the state vector can be drawn from a large class of prior distributions without affecting the linear complexity of our algorithm. This class includes both Gaussian and nonsmooth sparse and group sparse priors for which we employ an interior point modification of our algorithm. We discuss applications in text modeling and neuroscience.

1 Introduction

State-space models have been established as a fundamental tool for the statistical analysis of time series data, providing an online and computationally tractable tool for many real-world applications. However, their applicability is often limited in practice to low-dimensional state spaces, since the computational complexity of inference in these models scales cubically in time and quadratically in space with the dimensionality d of the state vector.

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

This computational burden can be reduced if certain structure is present that allows for fast matrix-vector operations. Examples include the approximation of covariance matrices as sparse, banded, or low-rank matrices (e.g. (Bickel and Levina, 2008; Cressie and Johannesson, 2008)).

In many problems, only a few measurements are available at each time step. For such a setup, and for the special case of a linear-Gaussian state-space model with a sparse, tree-structured state dynamics matrix, Paninski (2010) presented an approximate Kalman filter algorithm with linear time and space complexity. The main idea was that the forward covariance can be approximated by a low-rank perturbation of the steady state covariance (i.e., the state covariance when zero measurements are available). Therefore the standard Kalman algorithm was modified such that only the low rank perturbations were updated at every time step, an operation that required just $O(d)$ time and space.

In this paper, we formalize and extend this algorithm to more general penalized state space models, where the measurement noise and the state temporal dynamics obey log-concave distributions and the state vector is further penalized by appropriate norms. If at each time step the Hessian of the log-prior is of special structure that allows for fast multiplications and matrix solvers, then we show that maximum a-posteriori (MAP) estimation using an approximate Newton method can be computed efficiently: a forward-backward algorithm incorporating a sequence of low rank updates requires just $O(dT)$ time and memory, where T is the number of timesteps and therefore dT is the total dimensionality of the full state path. We characterize the computational gain of this approach, and also derive a bound on the error of our approximative Newton direction that guarantees the stability and convergence of our algorithm. Finally, we present a large family of norms that satisfy these requirements, including Gaussian and other smooth priors, l_1 and total variation (TV) norms, as well as group-sparsity norms. For these nonsmooth norms we use an interior point method (Boyd and Vandenberghe, 2004), based on successive smooth approximations of the nonsmooth terms, to perform our posterior inference again in $O(dT)$ time and space.

This decrease in computational requirements from $O(d^3T)$ to $O(dT)$ per iteration, combined with the low number of iterations required from second order methods (as opposed to first order methods, which rely only on gradient information) enables the consideration of systems with much larger dimensionality than is otherwise possible.

2 Problem Setup

Let $\mathbf{X} = [x_1, \ldots, x_T]$ denote the signal that we wish to estimate, where each x_t is a d-dimensional vector and represents the value of $\mathbf{X}$ at time t (or its expansion coefficients on a given fixed basis). We assume a (continuous state) Markovian evolution of the state vector i.e. $p_X(X) = p_0(x_1)\prod_{t=2}^{T} p_x(x_t|x_{t-1})$ where we assume that p_X is log-concave in $\mathbf{X}$. At every point in time we observe a small measurement vector y_t that depends only on the current state vector x_t through its linear projection on a measurement matrix B_t of size $[b_t, d]$. The likelihood of the observations, which we also assume to be log-concave in $\mathbf{X}$, is given by $p_Y(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{T} p_y(y_t|B_t x_t)$. Our goal is to develop fast second order methods for MAP inference, i.e., the computation of

$$\mathbf{X}_{\text{MAP}} = \arg\min_X \{-\log p_X(\mathbf{X}) - \log p_Y(\mathbf{Y}|\mathbf{X})\}. \quad (1)$$

To do so we need to compute the gradient ∇ and the Hessian H of the posterior likelihood with respect to $\mathbf{X}$ and use them to compute the Newton direction $\mathbf{s} = -H^{-1}\nabla$.[1] The Hessian of the posterior is given by

$$H = \begin{bmatrix} G_1 & -E_1 & 0 & \ldots & 0 \\ -E_1^T & G_2 & -E_2 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & -E_{T-1}^T & G_T \end{bmatrix}, \quad (2)$$

with

$$\begin{aligned} E_t &= \frac{\partial^2}{\partial x_t \partial x_{t+1}} \log p_x(x_{t+1}|x_t) \\ G_t &= D_t + B_t^T W_t^{-1} B_t \\ D_t &= -\frac{\partial^2}{\partial x_t^2}\left(\log p_x(x_{t+1}|x_t) + \log p_x(x_t|x_{t-1})\right) \\ W_t^{-1} &= \text{diag}\left\{ -\frac{\partial^2}{\partial x^2} \log p_y(y_t|x)\Big|_{x=B_t x_t} \right\}. \end{aligned} \quad (3)$$

The block-tridiagonal Hessian can be inverted using the Block-Thomas (BT) algorithm (Isaacson and Keller, 1994), which we repeat here for completeness (Alg.1). We also annotate the cost of each operation.

[1] For now we assume that all the likelihoods are smooth and strictly log-concave and therefore the gradient, the Hessian and its inverse are well defined everywhere. This assumption will be relaxed below.

Algorithm 1 Classic BT (computes $\mathbf{s} = -H^{-1}\nabla$)

$M_1 = D_1 + B_1^T W_1^{-1} B_1, \quad \Gamma_1 = M_1^{-1} E_1^T$ $(O(d^3))$
$\boldsymbol{q}_1 = -M_1^{-1}\nabla_1$ $(O(d^2))$
for $i = 2$ to T **do**
  $M_t = D_t + B_t^T W_t^{-1} B_t - E_{t-1} M_{t-1}^{-1} E_{t-1}^T$ $(O(d^3))$
  $\Gamma_t = M_t^{-1} C_t^T$ $(O(d^3))$
  $\boldsymbol{q}_t = -M_t^{-1}(\nabla_t + E_{t-1}\boldsymbol{q}_{t-1})$ $(O(d^2))$
end for
$\mathbf{s}_T = \boldsymbol{q}_T$
for $t = T-1$ to 1 **do**
  $\mathbf{s}_t = \boldsymbol{q}_t - \Gamma_t \mathbf{s}_{t+1}$ $(O(d^2))$
end for

The cost of the algorithm is $O(Td^3)$ in time, and $O(Td^2)$ in space (needed for the storage of the matrices Γ_t). Our goal is to derive conditions and algorithms under which the cost of the Newton direction operation can be reduced to $O(Td)$. As we will see we can derive such algorithms under two general assumptions: i) The number of measurements at each time step is low and ii) for each t, the matrices E_t, D_t have a special "diagonal plus low rank" structure (in some convenient basis) that allows us to store, multiply and invert them with cost $O(k_t d)$, where $k_t \ll d$ is a small constant. We can then efficiently update and invert the matrices M_t, by approximating their inverses as

$$M_t^{-1} \approx \tilde{M}_t^{-1} := \tilde{D}_t^{-1} - L_t \Sigma_t L_t^T, \quad (4)$$

where $\tilde{D}_t^{-1}$ is a matrix that allows fast $O(d)$ matrix-vector operations, and $L_t V_t L_t^T$ is an appropriate k_t-rank matrix.

3 Fast Inference

3.1 Conditions for Fast Inference

We first motivate our algorithm and discuss under what conditions we can expect a significant computational gain. For now, we assume for simplicity that D_t, E_t are diagonal matrices without additional low rank terms. To examine when the approximation of (4) makes sense, we first derive an algebraic equivalent to the matrices M_t of Alg. 1 that are more convenient to work with mathematically (though not computationally; this form is only used in the analysis).

Proposition 3.1. *The matrices M_t^{-1} can be written as*

$$M_t^{-1} = \tilde{D}_t^{-1} - \tilde{D}_t^{-1} U_t^T (F_t^{-1} + U_t \tilde{D}_t^{-1} U_t^T)^{-1} U_t \tilde{D}_t^{-1}, \quad (5)$$

where $\tilde{D}_t, U_t$ and F_t are defined recursively as:

$$\begin{aligned} \tilde{D}_t &= D_t - E_{t-1}\tilde{D}_{t-1}^{-1} E_{t-1}^T, \quad \tilde{D}_1 = D_1 \\ U_t &= \begin{bmatrix} B_t \\ U_{t-1}\tilde{D}_{t-1}^{-1} E_{t-1} \end{bmatrix}, \quad U_1 = B_1 \\ F_t &= \begin{bmatrix} E_t & 0 \\ 0 & (F_{t-1}^{-1} + U_{t-1}\tilde{D}_{t-1}^{-1} U_{t-1}^T)^{-1} \end{bmatrix}, \quad F_1 = E_1. \end{aligned} \quad (6)$$

Proof. Using induction we can show that the matrices M_t can be written as

$$M_t = \tilde{D}_t + U_t^T F_t U_t, \tag{7}$$

where $\tilde{D}_t, U_t$ and F_t are given by (6). Applying the Woodbury lemma on (7), (5) follows. □

From a statistical viewpoint, we can view M_t and $\tilde{D}_t$ as modified versions of $\mathrm{Cov}(x_t|\boldsymbol{Y}_{1:t})^{-1}$ and $\mathrm{Cov}(x_t)^{-1}$, i.e., the inverse of the posterior forward and prior covariance at time t; in fact, this relation is exact for $t = T$. To examine whether the posterior term M_t^{-1} can be approximated by the prior term $\tilde{D}_t^{-1}$ plus a low rank term (4), we look at the matrix $\tilde{D}_t^{-1}U_t^T(F_t^{-1} + U_t\tilde{D}_t^{-1}U_t^T)^{-1/2}$, where the square root denotes the Cholesky factor.

The matrices U_t, F_t have dimensions $(\sum_{l=1}^t b_l) \times d$ and $(\sum_{l=1}^t b_l) \times (\sum_{l=1}^t b_l)$ respectively. However, note that the l-th block of U_t is the measurement matrix B_l multiplied with the product $\prod_{k=t-l+1}^{t-1} \tilde{D}_k^{-1}E_k$ (for $l > 1$). Therefore, if the product $\prod_{l=1}^t \tilde{D}_l^{-1}E_l$ goes to zero exponentially fast (e.g. the spectral norm satisfies $\|\tilde{D}_l^{-1}E_l\| \leq r < 1$, for all l), then the effect of the measurements at time k, although present at time $k+1$, will decrease exponentially and (assuming the information matrices $B_t^T W_t^{-1} B_t$ are suitably bounded) practically vanish after a few steps. As a result, at time t the posterior covariance will only be affected by the measurements taken at times $t - n_t, \ldots, t$, where n_t is a small integer. Consequently, M_t^{-1} (or M_t) can be written as the sum of a diagonal matrix $\tilde{D}_t^{-1}$ ($\tilde{D}_t$) plus a low rank matrix that captures a high fraction θ of the energy. To analyze this low rank approximation we utilize the notion of the *effective rank* of a matrix.

Definition 3.2. *The effective rank of a matrix U at threshold θ ($0 < \theta < 1$), is defined as the minimum integer k, such that there exists a matrix X with* $\mathrm{rank}(X) = k$ *and*

$$\|X - U\|_F^2 \leq (1-\theta)\|U\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The effective rank is equal to the number of the singular values needed to express a θ fraction of the energy. To get a sense of how it scales, consider the matrix U_t and suppose that at each time step we get one measurement with a $\mathcal{N}(0, I)$ Gaussian random vector, and that $\|\tilde{D}_l^{-1}E_l\| \leq r < 1$, for all l. Then we can obtain a crude l-rank approximation of U_t by taking the matrix $U_{t,l}$ that consists of the first l rows of U_t, (i.e., the ones that have the higher expected energy). We can find the number of rows needed to capture a θ fraction of the energy in the average case by solving

$$\begin{aligned} n_t &= \arg\min\{l \in \mathbb{N} : \mathbb{E}\|U_{t,l}\|_F^2 \geq \theta \mathbb{E}\|U_t\|_F^2\} \Rightarrow \\ n_t &= \left\lceil \frac{\log(1-\theta(1-r^{2t}))}{2\log(r)} \right\rceil, \end{aligned} \tag{8}$$

where $\lceil\cdot\rceil$ is the ceil function (the derivation can be found in the appendix). However the best n_t rank approximation to U (in terms of the residual energy) can be made by taking the singular value decomposition of U and keeping the first n_t singular vectors/values. Therefore, this number provides a good rule of thumb to explain how the effective rank scales with the parameters r and θ; when $n_t \ll d$, we should expect the low rank approximation to lead to substantial dimensionality reduction.

3.2 The Low-Rank Block-Thomas Algorithm

We now explain how to perform the low rank approximations in an efficient way. Obviously, performing an SVD on the matrix $\tilde{D}_t^{-1}U_t^T(F_t^{-1} + U_t\tilde{D}_t^{-1}U_t^T)^{-1/2}$ is not efficient since the number of columns of this matrix grows as $O(t)$. Instead we perform a series of successive approximations.

Consider M_t from (7) for $t = 2$. We can write this as

$$M_2 = \tilde{D}_2 + O_2 Q_2 O_2^T, \tag{9}$$

where $O_2 = [B_2^T \;\; E_1 D_1^{-1} B_1^T]$ and Q_2 is a block-diagonal matrix with $Q_2 = \mathrm{blkdiag}\{\mathrm{W}_2^{-1}, (\mathrm{W}_1 + \mathrm{B}_1\mathrm{D}_1^{-1}\mathrm{B}_1^{\mathrm{T}})^{-1}\}$. Now the matrix O_2 is of dimension $d \times (b_1 + b_2)$ and M_2 can again be inverted using the Woodbury lemma as

$$M_2^{-1} = \tilde{D}_2^{-1} - \tilde{D}_2^{-1}O_2(Q_2^{-1} + O_2^T\tilde{D}_2^{-1}O_2)^{-1}O_2^T\tilde{D}_2^{-1}.$$

We can now perform a partial (thin) SVD on the term $\tilde{D}_2^{-1}O_2(Q_2^{-1} + O_2^T\tilde{D}_2^{-1}O_2)^{-1/2}$ and keep only the first k_2 singular values/vectors, where k_2 is the effective rank at threshold θ. Therefore we can write

$$M_2^{-1} \approx \tilde{M}_2^{-1} := \tilde{D}_2^{-1} - L_2^T\Sigma_2 L_2 \tag{10}$$

and repeat this procedure for all t. At every step t, the effective rank, and thus the number of columns of L_t and Σ_t, will satisfy $k_t \leq k_{t-1} + b_t$. If the stability condition $\|\tilde{D}_l^{-1}E_l\| \leq r < 1$ is satisfied, k_t will remain bounded around bn_t (where b is the average number of measurements per time step), much smaller than the dimension d for large d and small b. The resulting Low-Rank Block-Thomas (LRBT) algorithm is summarized in Alg. 2, where we annotate the cost of each operation.

Note that, besides the $O(dT)$ time complexity, the algorithm also requires $O(dT)$ space. All we need to store are the matrices L_t, Σ_t and the vectors $\boldsymbol{q}_t$, each of which takes $O(k_t d), O(k_t), O(d)$ space respectively.

Remark 3.3. *In the derivation of the algorithm we assumed that the matrices E_t, D_t are diagonal (or more generally can be diagonalized by a convenient fast transform). In the case where these matrices are diagonal plus a low rank symmetric matrix then Alg. 2 stays essentially the same. The only difference is that the matrix O_t now also includes the additional low rank term. Such a setup arises in group sparsity priors, as we'll see in section 4.*

Algorithm 2 Low-Rank Block-Thomas Algorithm

$\tilde{D}_1 = D_1, L_1 = D_1^{-1} B_1^T$ $(O(b_1 d),\ k_1 = b_1)$
$\Sigma_1 = (W_1 + B_1 D_1^{-1} B_1^T)^{-1}$ $(O(b_1^3))$
$\tilde{\boldsymbol{q}}_1 = (-D_1^{-1} + L_1 \Sigma_1 L_1^T)\nabla_1$ $(O(b_1 d))$
for $t = 2$ to T **do**
  $\tilde{D}_t = D_t - E_{t-1}\tilde{D}_{t-1}^{-1} E_{t-1}^T$ $(O(d))$
  $O_t = [B_t^T \ \ E_{t-1} L_{t-1}]$
  $Q_t = \mathrm{blkdiag}\{\mathrm{W}_\mathrm{t}^{-1}, \Sigma_{\mathrm{t}-1}\}$
  $[\hat{L}_t, \hat{\Sigma}_t^{1/2}] = \mathrm{svd}(\tilde{D}_t^{-1} O_t (Q_t^{-1} + O_t^T \tilde{D}_t^{-1} O_t)^{-1/2})$
    (thin SVD, cost $O((b_t + k_{t-1})^2 d)$)
  Truncate $\hat{L}_t$ and $\hat{\Sigma}_t$ to derive L_t and Σ_t.
    (effective rank $k_t \leq b_t + k_{t-1} \ll d$)
  $\tilde{\boldsymbol{q}}_t = -(\tilde{D}_t^{-1} - L_t \Sigma_t L_t^T)(\nabla_t + E_{t-1}\tilde{\boldsymbol{q}}_{t-1})$ $(O(k_t d))$
end for
$\tilde{\mathbf{s}}_T = \tilde{\boldsymbol{q}}_T$
for $i = T - 1$ to 1 **do**
  $\tilde{\mathbf{s}}_t = \tilde{\boldsymbol{q}}_t - (\tilde{D}_t^{-1} E_t^T - L_t \Sigma_t L_t^T E_t^T)\tilde{\mathbf{s}}_{t+1}$ $(O(k_t d))$
end for

Now that the algorithm has been established, we turn to a brief stability analysis. The output of the algorithm, $\tilde{\boldsymbol{s}}$, is linear in the input gradient vector ∇, and may therefore be written as $\tilde{\boldsymbol{s}} = -\tilde{H}^{-1}\nabla$ for some $\tilde{H}$ which approximates the true Hessian H. The proof of the following proposition establishes that $\tilde{H}$ is positive definite (under appropriate conditions); therefore $\tilde{\boldsymbol{s}}$ represents a steepest descent direction under the quadratic norm induced from $\tilde{H}$, (with corresponding bounds on the error $\|\tilde{\boldsymbol{s}} - \boldsymbol{s}\|$). Furthermore our algorithm may be used as an effective preconditioner in a conjugate gradient solver.

Theorem 3.4. *Under the condition $\|D_t^{-1} E_t\| \leq r < 1$, the following are true for sufficiently large threshold θ:*

1. $\|\tilde{H} - H\| \leq O(1 - \theta)$,
2. *$\tilde{\boldsymbol{s}}$ is a descent direction and computes the search direction for a convergent inexact Newton's method.*

Proof. We provide a sketch of the proof here; see the appendix for details. Using the forward-backward structure of Alg. 2, we can compute the approximation of the true Hessian as

$$\tilde{H} = \begin{bmatrix} \tilde{G}_1 & -E_1 & 0 & \dots & 0 \\ -E_1^T & \tilde{G}_2 & -E_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -E_{T-1}^T & \tilde{G}_T \end{bmatrix} \tag{11}$$

$\tilde{G}_t = \tilde{M}_t + E_{t-1}\tilde{M}_{t-1}^{-1} E_{t-1}^T$.

From (2) and (11) we see that the approximate Hessian differs only in the main diagonal from the true one. To analyze the difference, we define the matrices $\hat{M}_t$ as $\hat{M}_1 = M_1$ and

$$\hat{M}_t = D_t + B_t^T E_t B_t - E_{t-1}\tilde{M}_{t-1}^{-1} E_{t-1}^T. \tag{12}$$

Using the fact from the BT recursion

$$G_t = M_t - E_{t-1} M_{t-1}^{-1} E_{t-1}^T, \tag{13}$$

and as a result the approximation error is

$$H - \tilde{H} = \mathrm{blkdiag}\{\hat{\mathrm{M}}_1 - \tilde{\mathrm{M}}_1, \dots, \hat{\mathrm{M}}_\mathrm{t} - \tilde{\mathrm{M}}_\mathrm{t}\}. \tag{14}$$

Now $\tilde{M}_t^{-1}$ is obtained by performing a low rank approximation on $\hat{M}_t^{-1}$ and as a result we have $\|\tilde{M}_t^{-1} - \hat{M}_t^{-1}\| = O(1 - \theta)$, which also implies that $\|H - \tilde{H}\| = O(1 - \theta)$.

To prove the second part, it is not hard to show that $\tilde{H}$ is positive definite (proof in the supplement), which guarantees that $\tilde{\mathbf{s}}$ is a descent direction. Moreover, $\tilde{\mathbf{s}}$, solves

$$\begin{aligned} &\tilde{H}\tilde{\mathbf{s}} = -\nabla \Rightarrow \\ &H\tilde{\mathbf{s}} = -\nabla + (\tilde{H} - H)\tilde{H}^{-1}\nabla, \end{aligned} \tag{15}$$

which shows that $\tilde{\mathbf{s}}$ is an inexact Newton's method direction (Dembo et al., 1982; Sun and Yuan, 2006), with remainder $(\tilde{H} - H)\tilde{H}^{-1}\nabla$. Convergence is guaranteed if

$$\|(\tilde{H} - H)\tilde{H}^{-1}\nabla\| \leq r\|\nabla\|, \tag{16}$$

for some $r < 1$ and for all $\mathbf{x}$ (Eisenstat and Walker, 1994). Since $\|H - \tilde{H}\| = O(1 - \theta)$, this is always possible by picking a sufficiently large θ. □

4 Choice of State Prior and Nonsmoothness

In this section we deal with the important question of what log-concave, Markovian priors $p_X(\mathbf{X})$ satisfy the requirements of our algorithm.

4.1 Gaussian Priors

Consider first the simplest case of a Gaussian state transition prior $x_{t+1}|x_t \sim \mathcal{N}(Ax_t, V)$, where A is a stable matrix, i.e., $\|A\| < 1$. In this case we have (assuming that the initial state x_1 has covariance $V_0 = V$, A is normal, and that A and V commute)

$$\tilde{D}_t^{-1} E_t = -A^T \left(I - (A^T A)^t\right)\left(I - (A^T A)^{t+1}\right)^{-1}.$$

The requirements of the algorithm can be met in cases where the noise covariance V is diagonal, and A has a special sparse structure (e.g. diagonal, banded or adjacency matrix in a tree (Paninski, 2010)). Moreover, $\|\tilde{D}_t^{-1} E_t\| \uparrow \|A\|$, so the stability of A also implies the stability of the LRBT algorithm, and the effective rank does not depend in the observations.

Remark 4.1. *Using a similar analysis, the bound of* (8) *times the number of measurements per timestep holds for the effective rank in the Gaussian case (where $r \leq \|A\|$). Moreover, we can also show that the effect of the noise intensity on the effective rank is limited. The reason for this*

is that the matrix $F_t^{-1} + U_t \tilde{D}_t^{-1} U_t^T$ *of* (5) *is a block diagonal where all the blocks have similar structure and energy. Thus the effective rank is primarily determined by the behavior of* U *and scales at most as* $\log(1-\theta)/\log(\|A\|)$.

4.2 Sparse Priors

Of particular interest are priors that promote sparsity, either in the entries of the state vector x_t (e.g., via l_1-type norms on x_t), or in its variations $x_{t+1} - x_t$ (TV norm). The l_1 norm is important since it promotes sparsity but is also convex. However it is not smooth and therefore our method is not directly applicable. To apply our method we can use an interior point method (Boyd and Vandenberghe, 2004), where in each outer iteration we smooth the non-smooth terms at successive levels and apply our method on the smoothed objective functions. For example, the l_1-norm can be smoothed (at the level μ) by using Nesterov's (Nesterov, 2005) method as

$$|x| \approx f_\mu(x) := \sup_{|z|\leq 1} \left(zx - \frac{\mu z^2}{2} \right) = \begin{cases} \frac{x^2}{2\mu}, & |x| \leq \mu \\ |x| - \frac{\mu}{2}, & |x| > \mu \end{cases} \tag{17}$$

By denoting with $\mathcal{L}_\mu$ the negative posterior log-likelihood when we use the Nesterov approximation at level μ, our solution is given by $\mathbf{X}_{\text{MAP}} = \lim_{\mu \to 0^+} \arg\min_{\mathbf{X}} \{\mathcal{L}_\mu(\mathbf{X})\}$. Note that this solution is equal to the true minimizer (see e.g. (Becker et al., 2010) for the LASSO case). Moreover in some cases (e.g. the Dantzig selector (Becker et al., 2010)), the minimizer of $\mathcal{L}_\mu(\mathbf{X})$ is equal to the true one even for small enough but positive μ, reducing the number of outer loops required.

With that in mind, many sparsifying terms can be incorporated in our setup; e.g., in a fused Lasso setup (Tibshirani et al., 2005) we have

$$-\log p_X(\mathbf{X}) \propto \lambda_1 \sum_{t=1}^{T} \|x_t\| + \lambda_2 \sum_{t=2}^{T} \|x_t - x_{t-1}\|. \tag{18}$$

Using the smooth approximation of (17) we have (by abuse of notation let $E_0, E_T = 0$)

$$\begin{aligned} E_t &= \lambda_2 \text{diag}\left(f''_\mu(x_{t+1} - x_t)\right) \\ D_t &= E_t + E_{t-1} + \lambda_1 \text{diag}\left(f''_\mu(x_t)\right), \end{aligned} \tag{19}$$

which allow for fast matrix-vector operations.

More generally, any convex combination of well defined norms of the form $\sum_{t=1}^{T} f(\mathbf{1}^T g(x_t))$ or $\sum_{t=2}^{T} f(\mathbf{1}^T g(x_t - x_{t-1}))$ where $f, g : \mathbb{R} \mapsto \mathbb{R}$ (g is applied separately to each element of x_t) can be incorporated into our model without affecting the linear complexity. For example, the terms D_t in the first case are "diagonal plus rank one" given by

$$\begin{aligned} \frac{\partial^2}{\partial x_t^2} f(\mathbf{1}^T g(x_t)) =& f'(\mathbf{1}^T g(x_t)) \text{diag}\{g''(x_t)\} \\ &+ f''(\mathbf{1}^T g(x_t)) g'(x_t) g'(x_t)^T. \end{aligned} \tag{20}$$

Similarly, for the second class of norms, the contribution to the terms D_t of the Hessian is a "diagonal plus rank two" matrix, whereas the contribution to the terms C_t is a "diagonal plus rank one" matrix. Again our algorithm can be run with linear complexity (see Rem. 3.3). Apart from the l_1 and TV norms considered above, this class of norms includes many other norms of interest. For example, the group l_1-l_2 norm (Yuan and Lin, 2006) (or a group TV-l_2 variant) can be obtained by setting $f(x) = \sqrt{x}$ and $g(x) = x^2$. As before, the Nesterov method can be used to provide a smooth approximation. Our fast interior point method is summarized in Alg. 3.

Algorithm 3 Fast Interior Point Algorithm

Pick $\mu_0 > 0$, $\epsilon > 1$, $\boldsymbol{x_0}$, set $\mu \leftarrow \mu_0$, $\boldsymbol{x} \leftarrow \boldsymbol{x_0}$.
repeat
  Smooth objective function at level μ: $\mathcal{L}_\mu$
  repeat
    Find search direction $\tilde{\boldsymbol{s}}$ using Alg. 2.
    Find stepsize t using back-tracking line search.
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + t\tilde{\boldsymbol{s}}$
  until convergence
  $\mu \leftarrow \mu/\epsilon$.
until convergence
$\boldsymbol{X}_{\text{MAP}} = \boldsymbol{x}$.

Note in the absence of smooth prior terms, the Nesterov smoothing method can lead to numerical instabilities since the smoothed versions of E_t, D_t are not guaranteed to be positive definite. For example in (17), $f''(x) = 0$ for $|x| > \mu$. In this case we can use other smooth approximations, e.g. $\|x\| \approx (x^2 + \mu)^{1/2}$ or $\|x\| \approx \mu \log(\cosh(x/\mu))$.

5 Applications

5.1 Estimation of Non-stationary Receptive Fields

We begin with an example from sensory neuroscience. We present a synthetic but realistic example of estimation of a one-dimensional, time-varying receptive field (RF) from Poisson process observations. The function to be estimated was of the form

$$u(x,t) = h(x - r(t)), \tag{21}$$

i.e, a constant spatial RF function $h(x)$ that is centered around a time varying point that is given by $r(t)$. This function g can represent a drift of the receptive field, e.g. due to eye movement. Such drifts affect the standard analysis of spiking data for receptive field estimation (Read and Cumming, 2003; Tang et al., 2007), and therefore must be estimated prior to estimating the RF. In our case, $r(t)$ was a smooth sine-wave that was randomly jittered at every time step with probability 2%, giving a piecewise smooth u. The time vector was normalized to the interval $[0, 1]$ and was discretized into 1000 bins. At each time step t, n_t spikes

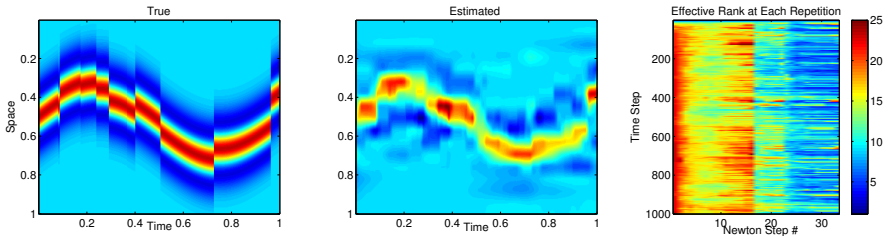

Figure 1: Estimation of a time-varying receptive field from Poisson observations. Left: True RF. Middle estimated. Right: Effective rank for each Newton direction computation.

were observed, where n_t was drawn from a Poisson distribution with rate $\lambda(t) = \exp(\langle u(\cdot, t), z_t \rangle)$, and the stimulus z_t was a normalize white noise random vector. The mean of n_t was 1.5, with roughly one third of the measurements having zero spikes (median 1). The signal was estimated by inferring the Laplacian pyramid expansion coefficients of the signal at every time step (Burt and Adelson, 1983). We used the following prior:

$$\log p_X(\mathbf{X}) \propto -\sum_{t=2}^{T} \frac{1}{2}(x_t - x_{t-1})^T W^{-1}(x_t - x_{t-1}) - \lambda_1 \sum_{t=1}^{T} \|x_t\|_1 - \lambda_2 \sum_{t=2}^{T} \|x_t - x_{t-1}\|_1.$$

The Gaussian term was chosen to capture the smooth parts of the drift, whereas the TV norm is used to capture the discontinuities. Finally, the l_1 norm is used since the signal is expected to be sparse in the Laplacian pyramid basis. Imposing sparse priors on (static) receptive fields has been shown to lead to more accurate estimation from a limited number of measurements (Mineault et al., 2009; Hu and Chklovskii, 2009). The Gaussian prior covariance W for the Laplacian pyramid was chosen to be diagonal.

Fig. 1 shows the correct signal $u(x, t)$ and the estimated signal with $\lambda_1 = 0.01, \lambda_2 = 0.25$. Although the data is very noisy (Poisson observations) and the number of spikes per bin is not unrealistically high, it can be seen that our algorithm captures the main structure of the time-varying RF and some of its discontinuities. The lower right corner of Fig. 1 shows the effective rank k_t of our algorithm for every time step and all the Newton steps required for convergence. The threshold in this setup was set to $\theta = 1 - 10^{-4}$. Setting a lower threshold gives even smaller effective rank, but a less accurate search direction, resulting in a larger number of iterations before convergence. As can be seen, the effective rank k_t grows linearly at the first timesteps but then quickly stabilizes at a low value which was always less than 25, whereas the dimension was $d = 256$ in this case. Moreover, these relative high values for the effective rank were observed only at the first Newton steps. As the algorithm converges the effective rank drops significantly.

To quantify the computational complexity of our algorithm

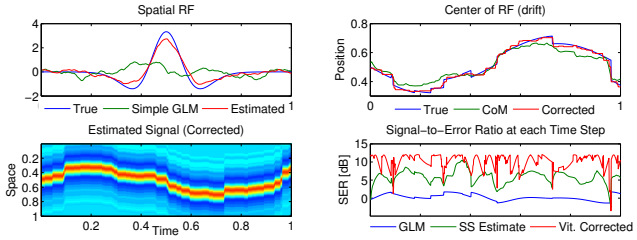

Figure 2: Estimation of spatial RF and drift. Upper Left: True spatial RF (blue), estimation with Poisson regression (green) and with the proposed method (red). Upper Right: True drift (blue), CoM estimate (green) and final estimate (red). Lower Left: Corrected Estimation of the full time varying RF. Lower Right: SER of plain Poisson estimate (blue), initial estimate with our state space model (blue) and corrected estimate using the Viterbi algorithm (green).

we also solved this problem using the TFOCS package (Becker et al., 2010), which can efficiently handle multiple nonsmooth regularizers. For the example used here our algorithm ran in 12.4 sec whereas TFOCS required 42.7. For the same example but with $d = 512$, our algorithm converged in 28 sec (exhibiting approximately linear scaling in d), as opposed to 346 sec required by TFOCS.

Although the estimate of the time-varying RF is not very accurate, since we have not exploited the fact that the RF shape $h(.)$ is constant here, we can use this estimate as an initial guess for the separate estimation of the drift and spatial components. A simple way to do this is as follows: We form an initial estimate of the drift by calculating the center of mass (CoM) of the estimated RF at each time. We use this to fit a purely spatial RF with the above drift, using standard penalized Poisson regression methods. Then we calculate again the drift by finding the most likely path of the fitted spatial RF using the Viterbi algorithm (Forney Jr, 1973). This procedure can be iterated if necessary. The results are shown in Fig. 2.

In the upper left panel of Fig. 2, we see that the corrected estimate of the spatial RF is very close to the true RF. On the other hand, a simple Poisson regression that does not compensate for the drift cannot predict the RF (green

curve). From the upper right corner we see that estimation of the drift using the Viterbi algorithm (red curve) is very accurate and captures both the smooth and the discontinuous transitions. Note also that the initial estimate based on the CoM is also fairly accurate. With these corrected estimates we can form a new estimate for the full time-varying RF (lower left corner of Fig. 2). The corrected estimate is more accurate than the initial one, as shown in the lower right corner of Fig. 2, where the signal-to-error ratio (SER) is plotted for each estimate, at all times. The SER is defined as $20\log_{10}(\|\mathbf{x_t}\|/\|\mathbf{x_t} - \hat{\mathbf{x}}_t\|)$. We are currently pursuing applications of our algorithm to real data.

5.2 Smoothing Multinomial Time Series Data

Next we briefly discuss applications to a simplified version of the influential dynamic topic model introduced in (Blei and Lafferty, 2006). Suppose that the word probabilities at time t within a text are described by the vector x_t:

$$P(w_t = i) = b_{t,i} := \exp(x_t(i))/\sum_{j=1}^{d}\exp(x_t(j)), \quad (22)$$

where the state vector $\mathbf{X} = [x_1, \ldots, x_T]$ follows a suitable log-concave prior distribution, like those presented above. The observation y_t at time t is the count data of each word at this time, drawn from a multinomial distribution with parameters (N_t, b_t), where N_t is the number of words observed at time t, and b_t is the vector of event probabilities at time t, as defined by (22). Then we have

$$\log p(y_t|x_t) \propto y_t^T x_t - N_t \log(\mathbf{1}^T \exp(x_t)), \quad (23)$$

which is concave in x_t. Moreover the first term is linear in x_t, whereas the second is of the form $f(\mathbf{1}^T g(x_t))$, and therefore its contribution to the Hessian is a diagonal plus rank one matrix (20). Thus, although the number of different words d can be very large, the observation at each time has the special structure that allows fast posterior inference of the dynamic mixture proportions.

We present an example where word count data are observed over $T = 100$ steps. The prior was chosen as $x_t|x_{t-1} \sim \mathcal{N}(0, 0.25I)$ and $x_1 \sim \mathcal{N}(0, I)$. We run the smoother for 6 different values of N_t and five different values of d.

In Fig. 3 we see that the mean effective rank stays very low even for large values of d (e.g. d=1000). More interestingly, it drops as the number of observations N_t per time step increases and remains approximately constant for a fixed ratio N_t/d. This can be explained by observing that the diagonal term of $-\partial^2 \log p(y_t|x_t)/\partial x_t^2$ is equal to $N_t\text{diag}\{b_t\}$. In our algorithm, this acts additively to the matrix $\tilde{D}_t$ and shifts its spectrum. Similarly to (8), the effective rank is expected to scale roughly as $k_t \propto 1/\log(\|N_t\text{diag}\{b_t\}\|)$. Now if each entry of b_t has the same marginal distribution, its value will be roughly of the order $O(1/d)$ and thus $\|N_t\text{diag}\{b_t\}\| = O(N_t/d)$.

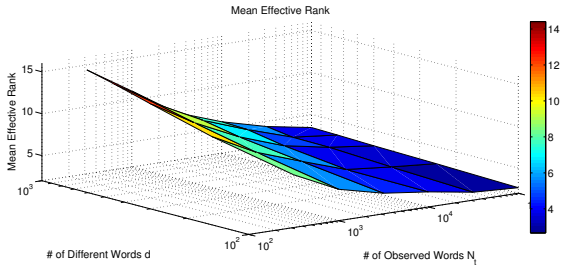

Figure 3: Variation of the mean effective rank with the number of words d and observations N_t. The effective rank scales roughly as $O((\log(N_t/d))^{-1})$ and agrees with our theoretical predictions.

Table 1: Comparison of our LRBT method with the low memory BFGS method. Our method scales linearly and is significantly faster for medium and large problem sizes.

	LRBT			BFGS		
d	time	iter	h_L	time	iter	h_B
200	**4.5**	8	.0029	40	137	.0015
400	**9**	14	.0016	126	170	.0019
800	**11**	13	.0011	591	320	.0035
1600	**16**	13	.00078	2116	367	.0036
5000	**73**	23	.00063	-	-	-

We also compared the speed and accuracy of our fast second order method with the limited memory BFGS method (Liu and Nocedal, 1989), using its optimized implementation in the matlab function `minFunc` (Schmidt, 2011). (Other gradient methods such as conjugate gradients performed similarly.) We tested different values of d and $N = 10d$, $T = 500$. The state dynamics were the same as before. As an accuracy criterion we used the KL-divergence between the true and inferred word probabilities.

From Table 1 we see that for the same or even better accuracy (not shown), our method is faster than the first order approaches, and the difference in times scales with the dimension d. The main reason is that the number of iterations required by our LRBT algorithm remains fairly constant in d, while the cost per iteration scales linearly with d. The indexes h_L and h_B are the time cost per iteration, normalized by the dimensions d, for our LRBT method and the BFGS implementation respectively. As we see h_L and the number of LRBT repetitions remain approximately constant, indicating the linear complexity of our algorithm. On the contrary for the BFGS method, the number of iterations grows with d, resulting in superlinear time complexity. Moreover, by trying multiple realizations of the same inference problems we observed that our algorithm is robust and always requires a similar number of iterations for convergence. The required iteration count is highly variable in the case of first order methods, which in our experience often become

very slow near the convergence criterion. Thus the LRBT approach is preferred here.

5.3 Smoothing of Spatiotemporal Data with Nuclear Norm Penalties

We briefly sketch the case where each vector x_t represents the coefficients of a time varying matrix (e.g., in the neural setting, a time varying spatial receptive field) and we want to control the rank of this matrix parameter at each time. The rank function is not convex, but we can penalize the nuclear norm (NN) to control the rank. The NN of a matrix equals the sum of its singular values and is the convex envelope for the rank function (Fazel et al., 2001). Due to the recent advances in the matrix completion problem (Candes and Recht, 2009), many algorithms have been developed for NN minimization. In our case a fast alternating minimization method (FALM) (Goldfarb et al., 2009) is applicable. We can write our cost functional as

$$L(\mathbf{X}) \propto -\sum_{t=1}^{T} \log p_y(y_t|B_t x_t) - \log p_x(\mathbf{X}) + \rho \sum_{t=1}^{T} \|x_t\|_*$$

where p_y, p_x are log-concave densities that meet the requirements of our fast algorithm and $\|x_t\|_*$ represents the NN of x_t (when the latter is written in matrix form). In a simplified form, the backbone of the ALM methods consists of the iterative alternating minimizations

$$\min_{\mathbf{X}} \left(-\sum_{t=1}^{T} \log p_y(y_t|B_t x_t) - \log p_x(\mathbf{X}) + \frac{\|\mathbf{X}-\mathbf{Z}\|^2}{2\mu} \right)$$

$$\min_{\mathbf{Z}} \left(\rho \sum_{t=1}^{T} \|z_t\|_* + \frac{1}{2\mu}\|\mathbf{X}-\mathbf{Z}\|^2 \right)$$

where μ is an appropriate constant. For the details of a FALM method see (Goldfarb et al., 2009). For this setup, we can minimize the first function efficiently using our method with cost $O(dT)$. The minimizer of the second function can be found in closed form using the singular value thresholding (SVT) operator (Cai et al., 2010). In general the minimization cost of that function would be $O(d^{3/2}T)$ because of the cubic cost of each SVD (the dimension of the matrix is $\sqrt{d} \times \sqrt{d}$). However, this can be improved, since in smooth time varying problems we don't expect the SVD of x_t to change rapidly with t. As a result, iterative methods for SVT with warm starts can be used that do not require a full SVD (e.g. (Cai and Osher, 2010)). These methods converge relatively quickly and are of cubic complexity only at points with discontinuities.

6 Discussion

We presented a fast interior point algorithm for performing inference in high dimensional penalized state space models. Our algorithm is applicable in state space models where only a few measurements are observed per time step and the prior is of a special structure that allows fast computations. We showed that in this case a good approximation to the Newton direction can be efficiently computed by using a forward backward algorithm in the block-tridiagonal Hessian, based on provably low rank updates. We characterized the computational gain of the algorithm and showed that the error of the approximate Newton direction remains appropriately bounded.

Although they require a low number of iterations (typically between 10 and 50) to achieve good accuracy, interior point methods are rarely used in general medium to large scale problems because of the large complexity per iteration. As an alternative, first order methods typically exhibit a relatively low cost per iteration. In a typical smooth setup the number of repetitions to achieve ϵ-accuracy is $O(\log(1/\epsilon))$ (Boyd and Vandenberghe, 2004). Similar convergence rates can be established in some certain nonsmooth cases: examples include the message passing algorithm of (Donoho et al., 2009) or the projected gradient descent algorithm for restricted strongly convex functions (Agarwal et al., 2011). The problem of sparse signal estimation in the context of state-space models has also received some attention (Vaswani, 2008; Carmi et al., 2010; Asif and Romberg, 2010; Ziniel et al., 2010), although these studies do not focus on fast computation methods.

Our algorithm exploits the special structure of the state-space MAP estimation problem to combine the fast convergence methods of interior point methods with the low cost per iteration ($O(dT)$ scaling) of first order methods. Moreover, it can incorporate multiple priors, including non-smooth and sparsity priors, without affecting its linear convergence characteristics. As a result, the proposed methods provide a flexible, efficient framework for tractable exact inference in this high dimensional state space setting.

We believe that our methods will be useful in a number of applied settings. In the future we also plan to pursue some open theoretical questions; for example, we would like to further examine the rate of convergence compared to the exact Newton method; to develop good guidelines for choosing the optimal threshold value θ; and to develop rigorous a priori estimates of the effective rank in non-Gaussian and nonnsmooth settings.

Acknowledgements

We thank J. Huggins and K. Rahnama Rad for helpful discussions. This work was supported by an NSF CAREER award, a McKnight Scholar award, and DARPA contract N66001-11-1-4205.

References

Agarwal, A., Negahban, S., and Wainwright, M. (2011). Fast global convergence of gradient methods for high-dimensional statistical recovery. *Arxiv preprint arXiv:1104.4824*.

Asif, S. and Romberg, J. (2010). Dynamic updating for l_1 minimization. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):421–434.

Becker, S., Candès, E., and Grant, M. (2010). Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):1–54.

Bickel, J. and Levina, E. (2008). Regularized estimation of large covariance matrices. *Annals of Statistics*, 36:199–227.

Blei, D. and Lafferty, J. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Oxford University Press.

Burt, P. and Adelson, E. (1983). The Laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4):532–540.

Cai, J., Candes, E., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Cai, J. and Osher, S. (2010). Fast singular value thresholding without singular value decomposition. *UCLA CAM Report*, pages 10–24.

Candes, E. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772.

Carmi, A., Gurfil, P., and Kanevsky, D. (2010). Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms. *Signal Processing, IEEE Transactions on*, 58(4):2405–2409.

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal Of The Royal Statistical Society Series B*, 70:209–226.

Dembo, R., Eisenstat, S., and Steihaug, T. (1982). Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408.

Donoho, D., Maleki, A., and Montanari, A. (2009). Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914.

Eisenstat, S. and Walker, H. (1994). Globally convergent inexact Newton methods. *SIAM Journal on Optimization*, 4(2):393–422.

Fazel, M., Hindi, H., and Boyd, S. (2001). A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference, 2001. Proceedings of the 2001*, volume 6, pages 4734–4739. IEEE.

Forney Jr, G. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Goldfarb, D., Ma, S., and Scheinberg, K. (2009). Fast alternating linearization methods for minimizing the sum of two convex functions. *Arxiv preprint arXiv:0912.4571*.

Hu, T. and Chklovskii, D. (2009). Reconstruction of sparse circuits using multi-neuronal excitation (rescume). *Advances in Neural Information Processing Systems*, 22:790–798.

Isaacson, E. and Keller, H. (1994). *Analysis of numerical methods*. Dover Publications.

Liu, D. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.

Mineault, P., Barthelmé, S., and Pack, C. (2009). Improved classification images with sparse priors in a smooth basis. *Journal of Vision*, 9(10).

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

Paninski, L. (2010). Fast Kalman filtering on quasilinear dendritic trees. *Journal of Computational Neuroscience*, 28:211–28.

Read, J. and Cumming, B. (2003). Measuring V1 receptive fields despite eye movements in awake monkeys. *J Neurophysiol*, 90:946–960.

Schmidt, M. (2011). minFunc. http://www.di.ens.fr/ mschmidt/Software/minFunc.html.

Sun, W. and Yuan, Y. (2006). *Optimization Theory and Methods: Nonlinear Programming*. Springer Verlag.

Tang, Y., Saul, A., Gur, M., Goei, S., Wong, E., Ersoy, B., and Snodderly, D. M. (2007). Eye position compensation improves estimates or response magnitude and receptive field geometry in alert monkeys. *J Neurophysiol*, 97:3439–3448.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Vaswani, N. (2008). Kalman filtered compressed sensing. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 893–896. IEEE.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Ziniel, J., Potter, L., and Schniter, P. (2010). Tracking and smoothing of time-varying sparse signals via approximate belief propagation. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pages 808–812. IEEE.