DYNAMIC SPARSE CODING WITH SMOOTHING PROXIMAL GRADIENT METHOD

Rakesh Chalasani and Jose C. Principe

Department of Electrical and Computer Engineering, University of Florida
Gainesville, FL, USA - 32611.

ABSTRACT

In this work we focus on the problem of estimating time-varying sparse signals from a sequence of under-sampled observations. We formulate this problem as estimating hidden states in a dynamic model and exploit the underlying temporal structure to find a more accurate solution, particularly when the information in the observations is at scarce. We propose an optimization procedure based on smoothing proximal gradient method to estimate these hidden states. We show that the proposed model is efficient and more robust to the noise in the system.

Index Terms— State-space, Sparse coding, Dynamics, Proximal methods.

1. INTRODUCTION

Recovering a sparse signal in a linear inverse problem has attracted a lot of attention in the recent past and has improved performance tremendously in several applications. Such problems are now widely studied; both in statistics [1][2] and computational neuroscience [3]. Although these methods are very successful, in most cases they assume that the environment is static. There are several applications where there is temporal information; like video, audio, electroencephalogram (EEG), etc, and extending the sparse methods to these could be useful. This will be the main focus of this work [1].

We consider the problem of recovering the sparse signal $\mathbf{x}_t \in \mathbb{R}^k$ as *causal* estimation (or *filtering*) and assume that the observations at any time t ($\mathbf{y}_t \in \mathbb{R}^p$) are synthesized as a linear combination of a set of basis functions $\mathbf{C}_t \in \mathbb{R}^{p\times k}$ (possibly also time-varying, but known *apriori*) with some sparse coefficients, called as *states* ($\mathbf{x}_t$), that are to be estimated. Mathematically, we write this generative model as:

$$\mathbf{y}_t \quad = \quad \mathbf{C}_t\mathbf{x}_t + \mathbf{n}_t \tag{1}$$

where $\mathbf{n}_t \in \mathbb{R}^p$ is the observation noise. We are here interested in the case when the system is over-complete (i.e., $k \gg p$) and recovering the hidden states require some regularization to find a unique solution. Since we assumed the states to be sparse, we regularize the solution with an ℓ_1-norm over the state vector $\mathbf{x}_t$.

Now, to keep track of the changes in the states over time, we assume that these dynamics are modeled through a *state transition* equation. For simplicity and without losing any generality, we assume the states follow some linear state transitions, taking the following form:

$$\mathbf{x}_t \quad = \quad \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{v}_t \tag{2}$$

where $\mathbf{A}_t$ is a called state-transition matrix (also known *apriori*) and $\mathbf{v}_t \in \mathbb{R}^k$ is the noise in the state predictions over time, called as *innovations*. The knowledge about the state transitions over time is very informative and can provide some *contextual* information to accurately infer the sparse states from the observations. We refer to this method of using state-space to recover sparse causal signals as *dynamic sparse coding* (DSC).

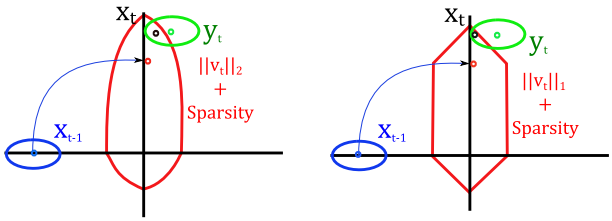
Previously, many methods were proposed to explore the possibility of using dynamics to recover the sparse, time-varying signals. Notably, some modifications to the Kalman filter are proposed based on selecting a constrained subset of basis [4] or using hierarchical Bayesian sparse coding [5]. Others addressed the problem as dynamic programming [6], using homotopy [7] or modeling the state innovations as Gauss-Bernoulli signal while using sampling methods [8]. To our knowledge, the closest method to our approach is re-weighted ℓ_1 dynamic filtering (RWL1-DF) [9], where dynamics over time are used to model the weighted sparsity prior on the states. However, it requires performing an ℓ_1 optimization multiple times for each time instance. Also, as we will show in our experiments (Section 3), it becomes unstable when the noise in the observations becomes large.

2. DYNAMIC SPARSE CODING

In contrast to the methods discussed above, we directly utilize the state-space equation with appropriate constraints on the solution. Specifically, we consider the innovations over states $\mathbf{v}_t$ also to be sparse along with the sparsity constraint on the states $\mathbf{x}_t$ itself. Such constraint on the innovations is not only consistent with the sparsity on the states, but also leads to a more stable and sparser solution. Fig. 1 shows the

This work is supported by ONR grant #N000141010375.

[1] Part of this work appeared previously in "Deep Predictive Coding Networks", Workshop at ICLR, 2013



(a) Gaussian Innovation + Sparsity (b) Sparse Innovation + Sparsity

Fig. 1: Comparison between Gaussian and sparse innovations along with sparsity constraint on the states in a 2D space. Shape of the combined priors on the states when (a) Gaussian innovations and ℓ_1 constraint and (b) sparse innovations and ℓ_1 constraint are applied. Note that the shape of the combined prior with Gaussian innovations is rounded around the solution $\mathbf{x}_t$ (in black), indicating that it does not promote sparsity. On the other hand, with the sparse innovations, the combined prior is sharp around the solutions, which is known to promote a sparse solution. (Best viewed in color)

comparison between modeling the innovations as sparse versus "dense" (or Gaussian distribution) (similar to Kalman filtering but without updating the covariance matrix over time). Notice that the shape of the combined regularizer over the states around the solution is sharper with sparse innovations, indicating that it promotes better sparsity than when the innovations are modeled as a Gaussian distribution.

Now, the cost function (or the negative log-likelihood) for the state-space model with the above described constraints can be written as:

$$\mathcal{L}(\mathbf{x}_t) = \frac{1}{2}\|\mathbf{y}_t - \mathbf{C}_t\mathbf{x}_t\|_2^2 + \lambda\|\mathbf{x}_t - \mathbf{A}_t\mathbf{x}_{t-1}\|_1 + \gamma\|\mathbf{x}_t\|_1 \tag{3}$$

Though this cost function is convex in $\mathbf{x}_t$, it has two non-smooth terms, making the optimization hard. While there are many methods proposed to solve smooth convex functions with ℓ_1 constraint, it is not straight forward to use them here. In this work, inspired by the method proposed in [10] for structured sparsity, we propose a smoothing proximal gradient method that first finds a smooth approximation to the non-smooth innovation term (second term in (3)) using Nesterov's smoothness property [11]. This allows us to re-formulate the cost function $\mathcal{L}(\mathbf{x}_t)$ into a regular ℓ_1 optimization problem and can be solved with proximal methods like fast iterative shrinkage thresholding algorithm (FISTA) [12].

2.1. Smooth approximation of sparse innovations

Firstly, we discuss the procedure to approximate the non-smooth sparse structured innovations term in (3). To begin, let $\Omega(\mathbf{x}_t) = \|\mathbf{e}_t\|_1$ where $\mathbf{e}_t = (\mathbf{x}_t - \mathbf{A}_t\mathbf{x}_{t-1})$. The idea is to first find a smooth approximation to this function $\Omega(\mathbf{x}_t)$ in $\mathbf{e}_t$. Since $\mathbf{e}_t$ is a linear function on $\mathbf{x}_t$, the approximation would also be smooth with respect to $\mathbf{x}_t$ and finding the gradient becomes straightforward.

Now, we can re-write $\Omega(\mathbf{x}_t)$ using the dual of ℓ_1-norm as

$$\Omega(\mathbf{x}_t) = \underset{\|\boldsymbol{\alpha}\|_\infty \leq 1}{\arg\max}\ \boldsymbol{\alpha}^T\mathbf{e}_t$$

where $\boldsymbol{\alpha} \in \mathbb{R}^k$. Using the Nesterov's smoothness property [11], $\Omega(\mathbf{x}_t)$ can be approximated with a smooth function $f_\mu(\mathbf{e}_t)$, which takes the following form:

$$\Omega(\mathbf{x}_t) \approx f_\mu(\mathbf{e}_t) = \underset{\|\boldsymbol{\alpha}\|_\infty \leq 1}{\arg\max}\ [\boldsymbol{\alpha}^T\mathbf{e}_t - \mu d(\boldsymbol{\alpha})] \tag{4}$$

where $d(\cdot) = \frac{1}{2}\|\boldsymbol{\alpha}\|_2^2$ is called a smoothing function and $\mu > 0$ is a smoothness parameter. From Nesterov's theorem [11], it can be shown that $f_\mu(\mathbf{e}_t)$ is convex and continuously differentiable in $\mathbf{e}_t$ and the gradient of $f_\mu(\mathbf{e}_t)$ with respect to $\mathbf{e}_t$ is given by:

$$\nabla_{\mathbf{e}_t} f_\mu(\mathbf{e}_t) = \boldsymbol{\alpha}^* \tag{5}$$

where $\boldsymbol{\alpha}^*$ is the optimal solution to $f_\mu(\mathbf{e}_t) = \underset{\|\boldsymbol{\alpha}\|_\infty \leq 1}{\arg\max}[\boldsymbol{\alpha}^T\mathbf{e}_t - \mu d(\boldsymbol{\alpha})]$. This optimal solution of $\boldsymbol{\alpha}$ in (4) can be analytically computed as follows:

$$\begin{aligned}\boldsymbol{\alpha}^* &= \underset{\|\boldsymbol{\alpha}\|_\infty \leq 1}{\arg\max}\ [\boldsymbol{\alpha}^T\mathbf{e}_t - \frac{\mu}{2}\|\boldsymbol{\alpha}\|^2] \\ &= \underset{\|\boldsymbol{\alpha}\|_\infty \leq 1}{\arg\min}\ \left\|\boldsymbol{\alpha} - \frac{\mathbf{e}_t}{\mu}\right\|^2 \\ &= S\left(\frac{\mathbf{e}_t}{\mu}\right)\end{aligned} \tag{6}$$

where $S(.)$ is a function projecting $\left(\frac{\mathbf{e}_t}{\mu}\right)$ onto an ℓ_∞-ball. This is of the form:

$$S(x) = \begin{cases} x, & -1 \leq x \leq 1 \\ 1, & x > 1 \\ -1, & x < -1 \end{cases}$$

Now, by using the chain rule, $f_\mu(\mathbf{e}_t)$ is also convex and continuously differentiable in $\mathbf{x}_t$ and the gradient of $f_\mu(\mathbf{e}_t)$ with respect to $\mathbf{x}_t$ also turns out to be the same.

2.2. Effect of smoothing

To visualize the effect of the above described smoothing operation, we plot the function $f_\mu(\mathbf{e}_t)$ for a one-dimensional error signal $\mathbf{e}_t \in \mathbb{R}$ for various values of μ. Note that μ determines the maximum value of $\boldsymbol{\alpha}$ in (4) ($\boldsymbol{\alpha}^*$) corresponding to each error value. Fig. 2 shows the resulting plots. As it indicates, the sharp point in ℓ_1-norm around the origin is smoothed in the approximated function $f_\mu(\mathbf{e}_t)$. Also note that, as the value of μ increases, the approximation, though smoother, starts to deviate more from the ℓ_1-norm. In fact, one can show that, given the desired accuracy ϵ of the solution, following convergence results from Theorem 2 in [10] suggests $\mu = \frac{\epsilon}{k}$, where k is the dimensions of the states, leads to the best convergence rate. We refer the reader to [10] for details.

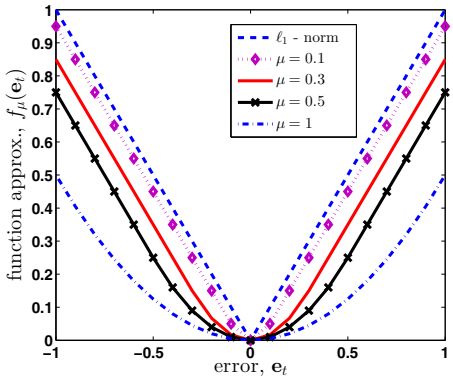


Fig. 2: Effect of smoothing on the cost function. Plot shows the smooth function $f_\mu(\mathbf{e}_t)$ versus a one dimensional error signal $\mathbf{e}_t$ for various values of the smoothness parameter μ.

2.3. Smoothing proximal gradient descent for DSC

Now, by substituting the smooth approximation of $\Omega(\mathbf{x}_t)$ in (4) into (3), we can re-write the cost function as follows:

$$\widetilde{\mathcal{L}}(\mathbf{x}_t) = \frac{1}{2}\|\mathbf{y}_t - \mathbf{C}\mathbf{x}_t\|_2^2 + \lambda f_\mu(\mathbf{e}_t) + \gamma\|\mathbf{x}_t\|_1 \tag{7}$$

Notice now that the new cost function $\widetilde{\mathcal{L}}(\mathbf{x}_t)$ is a regular ℓ_1 optimization problem, a smooth convex function:

$$h(\mathbf{x}_t) = \frac{1}{2}\|\mathbf{y}_t - \mathbf{C}\mathbf{x}_t\|_2^2 + \lambda f_\mu(\mathbf{e}_t)$$

regularized with an ℓ_1-norm. Also, the gradient of $h(\mathbf{x}_t)$ with respect to $\mathbf{x}_t$ is given by:

$$\nabla h(\mathbf{x}_t) = \mathbf{C}^T(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t) + \lambda\boldsymbol{\alpha}^* \tag{8}$$

Using this gradient information of $h(\mathbf{x}_t)$ in (8), we can now infer $\mathbf{x}_t$ that minimizes $\widetilde{\mathcal{L}}(\mathbf{x}_t)$ using the fast iterative shrinkage thresholding algorithm (FISTA) [12]. Algorithm 1 summarizes the steps involved, where L is the step-size parameter set using line-search method, as discussed by Beck and Teboulle [12], and $\mathcal{T}(.)$ is a soft-thresholding operation on each element, defined as:

$$\mathcal{T}_\lambda(x_i) = \text{sgn}(x_i)\max(|x_i| - \lambda)$$

3. EXPERIMENTS

We consider an experimental set-up similar to one used by Charles et al. [9] with synthetic data and compare the performance of the proposed dynamic sparse coding (DSC) with other methods—sparse coding using FISTA (SC) [12], Kalman filter [13], re-weighted ℓ_1 dynamic filtering (RWL1-DF) [9]. We also compare our method while considering

Algorithm 1 Smoothing proximal gradient descent for dynamic sparse coding

Require: Inputs - $\mathbf{y}_t$, $\mathbf{x}_{t-1}$, $\mathbf{C}_t$, $\mathbf{A}_t$, λ and γ
Desired accuracy - ϵ.
Initialize - $\boldsymbol{\beta}^0 = \mathbf{0}$, $\theta_0 = 1$, $\mathbf{z}^0 = \boldsymbol{\beta}^0$
Compute: $\mu = \frac{\epsilon}{k}$
for n = 0,1,2 ... until convergence **do**
  Compute $\nabla h(\mathbf{z}^n)$ using (8).
  Find step-size parameter L using line search [12].
  Update states: $\boldsymbol{\beta}^{n+1} = \mathcal{T}_{\lambda/L}(\mathbf{z}^n - \frac{1}{L}\nabla h(\mathbf{z}^n))$
  Set $\theta_{n+1} = \left(1 + \sqrt{1 + 4{\theta_n}^2}\right)/2$.
  Update $\mathbf{z}^{n+1} = \boldsymbol{\beta}^{n+1} + \left(\frac{\theta_n - 1}{\theta_{n+1}}\right)(\boldsymbol{\beta}^{n+1} - \boldsymbol{\beta}^n)$
end for
return $\mathbf{x}_t = \boldsymbol{\beta}^{n+1}$

the states innovations in (3) as Gaussian (SC-L2 Innov.), as depicted in Fig. 1[2].

Specifically, the experimental set-up is a follows: we simulate a state sequence with only 20 non-zero elements in a 500-dimensional state vector evolving with a permutation matrix (note that this keeps the number of non-zero elements same over time), which is different for every time instant. This state sequence is then passed through a Gaussian scaling matrix to generate a sequence of observations. We vary observation dimensions (p) depending on the experiment, which will be specified later. We consider that both the permutation and the scaling matrices are known *apriori*. The observation noise is Gaussian with zero mean and variance $\sigma^2 = 0.001$. We consider sparse state-transition noise, which is simulated by choosing a subset of active elements (n) in the state vector chosen randomly and switching each of them with a randomly chosen element (with uniform probability over the state vector). This resemble a sparse innovation in the states with $2n$ number of wrongly placed elements, one "missing" element and one "additional" element. We use these generated observation sequences as inputs and use the *apriori* known parameters to infer the states $\mathbf{x}_t$. To set the hyper-parameters, we perform a parameter sweep to find the best configuration for each method. We compare the inferred states from different methods with the true states in terms of relative mean squared error (rMSE); defined as

$$\frac{\|\mathbf{x}_t^{est} - \mathbf{x}_t^{true}\|}{\|\mathbf{x}_t^{true}\|}$$

Fig. 3 shows the tracking performance of different methods — see caption for details about the model used. Also, Table 1 shows the computation time (per time instance) for all the methods[3]. We observe that the dynamic sparse coding (DSC) is able to track the states over time more accurately than sparse coding (SC), which does not use any dynamics.

[2]We perform inference on this model using FISTA [12]
[3]All computations are done on 8-core Intel Xeon, 2.4 GHz processor.

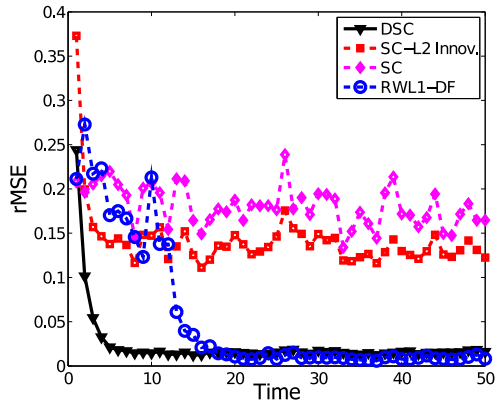


Fig. 3: Performance of dynamic sparse coding with sparse innovation while tracking sparse states. The plot shows the relative mean square error (rMSE) versus time and each plot is an average over 40 runs. Experiment is performed with $p = 70$, $n = 3$ and we set the parameters $\lambda = 10$ and $\gamma = 10$ in (3). Kalman filter completely failed to track the states and is not shown here.

Table 1: Computational time of different methods on synthetic data (per time sample).

Methods	DSC	SC-L2 Innov.	SC	RWL1-DF
Time (in seconds)	0.17	0.16	0.27	0.54

The dynamic model with Gaussian innovations (SC- L2 Innov.), though performs better than the sparse coding model at times, is not able to track the state accurately, re-asserting our argument that considering sparse innovations make the model more stable and consistent. Finally, RWL1-DF is able to track the states as accurately as our model, but requires several observations before reaching a steady state and is computationally more expensive. In fact, we observed that RWL1-DF becomes very unstable when the observations have "inadequate" information, as a result of very noisy observation or when the number of observation dimensions are less. We discuss more about this in the following experiments.

Fig. 4 shows the "steady" state error (rMSE) after 50 time instances versus with the dimensionality of the observation sequence (p). Each point is obtained after averaging over 50 runs. We observe that DSC is able to track the true states even for low dimensional observations, when other methods fail. This shows that the temporal relations adopted in the model provide contextual information necessary to track the changes in the observation, even when the information provided by the instantaneous observations is not sufficient. Notice also that RWL1-DF becomes very unstable when the dimensions of the observations are small.

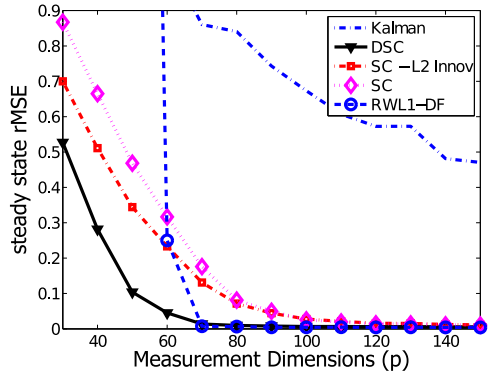


Fig. 4: Performance of the dynamic sparse coding with varying number of observation dimensions. We use similar set of parameters as before, $\lambda = 10$, $\gamma = 10$ and $n = 3$.

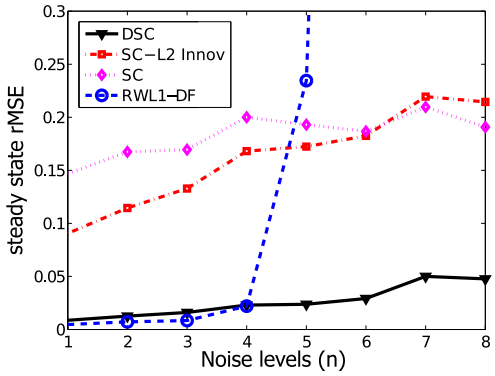


Fig. 5: Performance of the dynamic sparse coding with varying (sparse) noise levels (n). We use similar set of parameters as before, $\lambda = 10$, $\gamma = 10$ and $p = 70$.

Same can be extrapolated in case of noisy observation sequences, where the essential information in the time sequence is at scarce. Fig. 5 shows the performance of all the methods versus varying sparse noise levels (n). Again, we observe that DSC outperforms other methods, particularly when the noise levels are high. Also, notice that the RWL1-DF becomes very unstable when the noise levels are high.

4. CONCLUSION

In this work, we proposed a smoothing proximal gradient method for solving dynamic sparse coding problem. The key idea of the proposed method is to approximate the non-smooth state-innovation term using Nesterov's smoothness. This allowed us to re-formulate a difficult optimization problem into a simple regular ℓ_1 optimization problem, which we solve using FISTA. We show that the proposed model is able to efficiently and accurately estimate the states from time-varying signals.

5. REFERENCES

[1] Robert Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, , no. 1, pp. 267–288.

[2] Emmanuel J. Candés, Justin K. Romberg, and Terence Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[3] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images.," *Nature*, vol. 381, no. 6583, pp. 607–609, June 1996.

[4] N. Vaswani, "Kalman filtered Compressed Sensing," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, oct. 2008, pp. 893 –896.

[5] Evripidis Karseras, Kin Leung, and Wei Dai, "Tracking dynamic sparse signals using hierarchical bayesian kalman filters," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[6] D. Angelosante, G.B. Giannakis, and E. Grossi, "Compressed sensing of time-varying signals," in *Digital Signal Processing, 2009 16th International Conference on*, july 2009, pp. 1 –8.

[7] A. Charles, M.S. Asif, J. Romberg, and C. Rozell, "Sparsity penalties in dynamical system estimation," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, march 2011, pp. 1 –6.

[8] D. Sejdinovic, C. Andrieu, and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 29 2010-oct. 1 2010, pp. 1730 –1736.

[9] Adam S Charles and Christopher J Rozell, "Reweighted l1 dynamic filtering for time-varying sparse signal estimation," *arXiv preprint arXiv:1208.0325*, 2012.

[10] X. Chen, Q. Lin, S. Kim, J.G. Carbonell, and E.P. Xing, "Smoothing proximal gradient method for general structured sparse regression," *The Annals of Applied Statistics*, vol. 6, no. 2, pp. 719–752, 2012.

[11] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[12] Amir Beck and Marc Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, , no. 1, pp. 183–202, Mar.

[13] Rudolph Emil Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.