

EARTH-MOVER’S DISTANCE AS A TRACKING REGULARIZER

Adam S. Charles* Nicholas P. Bertrand† John Lee† Christopher J. Rozell†

*Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544

†School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332

*adamsc@princeton.edu, †{nbertrand, john.lee, crozell}@gatech.edu

ABSTRACT

Tracking time-varying signals is an important part of many engineering systems. Recently, signal processing techniques have been developed to improve tracking performance when the signal of interest is known *a-priori* to be sparse. Leveraging sparsity, however, depends heavily on gridding the space, treating the signal as a collection of active or inactive pixels in an image, rather than traditional methods which track the continuous spatial coordinates. Using the dynamics constraint in this setting is challenging, as a model which approximately predicts target location may result in seemingly large errors, as measured by the ℓ_p -norm typically used in such algorithms. To take advantage of approximate spatial priors without introducing unnecessary penalties, we present a tracking algorithm using the earth-mover’s distance (EMD) as an alternate dynamics regularization term. We note that while requiring a higher computational burden, the EMD can more effectively utilize target location prediction when the space is gridded.

Index Terms—Dynamic Filtering, Earth-mover’s Distance, Compressive Sensing, Kalman Filtering

1. INTRODUCTION

One of the classical problems in signal processing concerns the efficient tracking of temporally changing signals over time. Often termed *dynamic filtering*, these algorithms use a current set of measurements and a prediction of the underlying signal, given by a dynamical model, to infer the current signal as accurately as possible. Mathematically, we assume that for an unknown temporally-evolving signal $\mathbf{x}_n \in \mathbb{R}^N$, our measurements are taken as

$$\mathbf{y}_n = \mathbf{G}_n \mathbf{x}_n + \boldsymbol{\epsilon}_n, \quad (1)$$

where $\mathbf{y}_n \in \mathbb{R}^M$ are linear measurements taken at time n via the measurement matrix $\mathbf{G}_n \in \mathbb{R}^{M \times N}$, and $\boldsymbol{\epsilon}_n \in \mathbb{R}^M$ is potential measurement noise. Additionally, we use what knowledge is available of the temporal evolution of \mathbf{x}_n by modeling the dynamics as

$$\mathbf{x}_n = f_n(\mathbf{x}_{n-1}) + \boldsymbol{\nu}_n, \quad (2)$$

where $f_n : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a function that portrays our knowledge of the signal dynamics and $\boldsymbol{\nu}_n \in \mathbb{R}^N$ is the model error, often termed the *innovations*. Classically, the Kalman filter [1, 2] has provided an optimal and efficient inference methodology for problems where the measurement and dynamics model errors are Gaussian, and the measurements and dynamics function are linear.

In addition to the signal structure imbued by the dynamics process, modern signal processing also seeks to leverage additional *a-priori* known signal structure inherent to many signal classes to regularize inverse problems and improve the performance of tracking algorithms. Specifically, the concept of *sparsity*, which models the signal as having few non-zero elements in a linear representation, has gained traction in the tracking community [3–10]. Algorithms designed to leverage sparsity in tracking applications have typically focused on problems where the dynamics are accurate in terms of an ℓ_p metric. In some applications, however, such innovations models can actually mislead inference procedures. In particular, we consider in this work the case where the innovations are inherently spatial errors, i.e., dynamics models that can predict the location of targets in an image to within a few pixels. In this scenario, ℓ_p -norm based error metrics may unjustly penalize even very small spatial deviations. For example, in an image consisting of a single active target, an estimate that places the active pixel very close to the true location is penalized as much as one that places it far away.

Thus we focus here on time-varying signals where the dynamics model essentially represents small movement of the data with respect to a distance function. Instead of the ℓ_p dynamics inducing norms used in the literature, we move towards using a more natural regularizer: the earth mover’s distance (EMD) [11]. The EMD provides a natural way to describe small spatial errors. Specifically, in this work we describe a tracking algorithm that uses the EMD as a regularization term that conveys dynamic information over time. Additionally, we discuss some of the computational benefits of using the EMD distance for tracking sparse signals. Finally, we empirically explore the improvement of EMD dynamics regularization over current methods via target tracking simulations.

2. BACKGROUND

2.1. Dynamic Signal Tracking

Traditional tracking algorithms are predominantly based on the Kalman filter [1]. Succinctly, the Kalman filter can be described as a least-squares solution at each time step

$$\hat{\mathbf{x}}_n = \arg \min_{\mathbf{x}} \left[\|\mathbf{y}_n - \mathbf{G}_n \mathbf{x}\|_{2, \mathbf{R}_n}^2 + \|\mathbf{x} - \mathbf{F}_n \hat{\mathbf{x}}_{n-1}\|_{2, (\mathbf{Q}_n + \mathbf{F}_n \mathbf{P}_{n-1} \mathbf{F}_n^T)}^2 \right] \quad (3)$$

where $\hat{\mathbf{x}}_{n-1}$ and \mathbf{P}_{n-1} are the estimate of the previous time step and its covariance, and \mathbf{Q}_n and \mathbf{R}_n are the covariances for the innovations and measurement noise. The Kalman filter has been particularly popular because of its optimality guarantees under linear assumptions on the measurements and dynamics and Gaussian assumptions on the signal, innovations and measurement noise.

This work was supported in part by NSF grant CCF-1409422 and the James S. McDonnell Foundation.

Many current popular signal models, however, differ greatly from Gaussian assumptions. One particularly popular model that has found use in many applications is *sparsity*. In the sparsity model we describe our signal as the linear combination of few elements from a large dictionary

$$\mathbf{x}_n = \mathbf{\Psi}\mathbf{a}, \quad (4)$$

where $\mathbf{\Psi} \in \mathbb{R}^{N \times P}$ is the dictionary of atoms and \mathbf{a} is the sparse coefficient vector containing mostly zeros. This model has increased the performance of many signal and image processing tasks [12].

One of the most widely used algorithms for recovering the sparse coefficients from noisy measurements is basis pursuit denoising (BPDN) [13] (also known as the LASSO [14]) which involves solving the optimization problem

$$\hat{\mathbf{a}}_n = \arg \min_{\mathbf{a}} \|\mathbf{y}_n - \mathbf{G}_n \mathbf{\Psi}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1. \quad (5)$$

Here, the first term requires the solution to agree with the measurements, and the second term encourages sparse solutions. The parameter λ represents the trade-off between fidelity and sparsity. BPDN is appropriate for static problems, but does not utilize information provided by the dynamics model in tracking scenarios (e.g., in [3–10]). A natural extension to BPDN, basis pursuit denoising with dynamic filtering (BPDN-DF) [10, 15] involves a similar optimization problem with the addition of a tracking regularizer:

$$\hat{\mathbf{a}}_n = \arg \min_{\mathbf{a}} \|\mathbf{y}_n - \mathbf{G}_n \mathbf{\Psi}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1 + \gamma \|\mathbf{\Psi}\mathbf{a} - f(\hat{\mathbf{x}}_{n-1})\|_p^p. \quad (6)$$

The additional third term requires our estimated coefficients to agree with those predicted by the dynamics function f . The parameter γ is adjusted according to the accuracy of our dynamics model. We note here that the dynamics regularizer penalizes according to the ℓ_p -norm. Thus BPDN-DF incurs problems associated with insensitivity to the spatial location of the coefficients. More recent methods have sought to use variance information native to sparsity to induce additional robustness [15]. One such example, re-weighted ℓ_1 dynamic filtering (RWL1-DF) accomplishes this task by iteratively solving a weighted BPDN estimate

$$\hat{\mathbf{a}}_n = \arg \min_{\mathbf{a}} \|\mathbf{y}_n - \mathbf{G}_n \mathbf{\Psi}\mathbf{a}\|_2^2 + \sum_i \lambda_i |a_i|, \quad (7)$$

and updating the weights based on a combination of dynamics and measurement information

$$\lambda_i = \frac{\alpha}{\beta + |\hat{a}_i| + \gamma |[\mathbf{\Psi}^T f(\hat{\mathbf{x}}_{n-1})]_i|}, \quad (8)$$

for parameters α , β and γ . RWL1-DF allows for sparsity preserving deviations from an erroneous dynamics model [15], however RWL1-DF is still incurs the same spatial location insensitivity problems as BPDN-DF, as the spatial relationships between coefficients remain unaccounted for.

2.2. Earth-Mover's Distance

The earth-mover's distance (EMD), denoted here by $d_{\text{EMD}}(\mathbf{x}, \tilde{\mathbf{x}})$, is a metric that was originally devised as a method to compare probability distributions, however in recent years has been applied more generally to inverse problems [11, 16, 17]. If the elements of a vector are considered as “mass”, the EMD is the minimum work (i.e., mass

times displacement) needed to obtain one vector from the other. In one dimension, the EMD is trivial to calculate, as one can simply start at the beginning of a vector and move the difference between the two vectors' elements down to the next index until the last index is reached. In higher dimensions, however, the EMD is more difficult to calculate, and instead must be written as an optimization program. Specifically, $d_{\text{EMD}}(\mathbf{x}, \tilde{\mathbf{x}})$ can be written as the minimum value obtained by solving the linear program

$$\begin{aligned} \min_{\mathbf{F}} \sum_{i,j} F_{i,j} r_{i,j} \quad \text{s.t.} \quad & F_{i,j} \geq 0 \\ & \sum_j F_{i,j} \leq x_i \\ & \sum_i F_{i,j} \leq \tilde{x}_j \\ & \sum_{i,j} F_{i,j} = \min(\sum_i x_i, \sum_j \tilde{x}_j), \end{aligned} \quad (9)$$

where $F_{i,j}$ are flow variables which represent the mass moved from x_i to \tilde{x}_j , and $r_{i,j}$ is the associated “displacement” cost. In this linear program, the first line represents the minimization of the total work (sum of mass times displacement) subject to non-negativity constraints on the flow, the second line indicates the constraint that no more mass can come from a pixel than the mass already there, the third line indicates the constraint that no more mass can be placed in a pixel than the mass required, and the final line indicates the constraint that the total mass moved must be the minimum mass of the two vectors. The final constraint here is particularly important, as that constraint prevents the trivial solution of all zero flows. While the EMD has been used in linear inverse problems (e.g., in lieu of the ℓ_2 -norm in BPDN [11] or to regularize differences between columns of a sparse matrix [16]), the utility of the EMD for tracking applications has yet to be explored.

3. DYNAMIC FILTERING WITH THE EMD

Using the EMD as a dynamics regularizer follows the same overall strategy as in BPDN-DF. In particular we wish to solve

$$\hat{\mathbf{x}}_n = \arg \min_{\mathbf{x}} \|\mathbf{y}_n - \mathbf{G}_n \mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 + \gamma d_{\text{EMD}}(\mathbf{x}, \tilde{\mathbf{x}}), \quad (10)$$

where λ and γ are again pre-set trade-off parameters and $\tilde{\mathbf{x}} = f(\hat{\mathbf{x}}_{n-1})$ is the predicted signal using the dynamics model. The idea here is that the EMD distance will be more forgiving to innovations that are often encountered in spatial tracking scenarios.

3.1. Optimization Routine

Since the computation of the EMD is itself an optimization program, solving (10) is more involved than solving the BPDN-DF optimization (6). By substituting the EMD distance in (10) with the corresponding optimization program, we can see that to solve EMD-regularized BPDN we need to optimize over two sets of variables: the signal of interest \mathbf{x} and the EMD flow variables \mathbf{F} ,

$$\begin{aligned} \hat{\mathbf{x}}_n = \arg \min_{\mathbf{x}} \|\mathbf{y}_n - \mathbf{G}_n \mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 + \gamma \min_{\mathbf{F}} \sum_{i,j} F_{i,j} r_{i,j} \\ \text{s.t.} \quad & F_{i,j} \geq 0 \\ & \sum_j F_{i,j} \leq x_i \end{aligned}$$

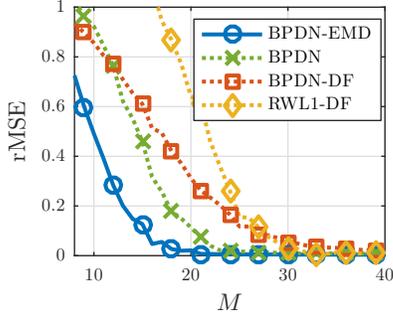


Fig. 1. One-step recovery results for EMD-regularized BPDN. When given a prediction of the true signal that is erroneous by small spatial extents (i.e., predicted pixels are close, but not precisely at their true location in the image), the EMD-regularized BPDN estimates the signal more accurately from many fewer measurements.

$$\sum_i F_{i,j} \leq \tilde{x}_j$$

$$\sum_{i,j} F_{i,j} = \min(\|\mathbf{x}\|_1, \|\tilde{\mathbf{x}}\|_1). \quad (11)$$

The objective here remains convex, and all the constraints, save the equality constraint, are linear. To avoid the non-convex term $\min(\|\mathbf{x}\|_1, \|\tilde{\mathbf{x}}\|_1)$, we note that at the solution, either $\sum_{i,j} F_{i,j} = \|\tilde{\mathbf{x}}\|_1$ or $\sum_{i,j} F_{i,j} = \|\mathbf{x}\|_1$ holds. We can thus solve the optimization (11) twice, replacing the final constraint with the two possible equality constraints, and take the solution with the smaller objective. While requiring two optimizations, this solution allows for EMD regularization in N -dimensional settings.

3.2. Computational Considerations

Solving the optimization in (11) can be computationally cumbersome, as we must solve for both the N signal variables, and the N^2 flow variables. When tracking sparse signals, however, we can reduce the overall number of variables in the optimization. Consider the second constraint in (11). On the left-hand side of this condition is the summation along the rows of the flow matrix \mathbf{F} , and on the right is the signal prediction $\tilde{\mathbf{x}}$. Since all flow variables are non-negative, and few elements of $\tilde{\mathbf{x}}$ are non-zero, we have that the sum of a large number of the flow variables of \mathbf{F} must equal zero: indicating that any row not corresponding to an active element of $\tilde{\mathbf{x}}$ must be populated entirely of zeros. This means that the actual number of variables to be optimized is $N + N\tilde{K}$, where \tilde{K} is the sparsity of $\tilde{\mathbf{x}}$. Additionally, we can mitigate the cost of having to run the optimization twice by seeding the second optimization program with the solution to the first. As a final note, while the optimization (11) can be implemented in general solver environments, (e.g., CVX [18,19]), we have also implemented our method using the efficient alternating directions method of multipliers (ADMM) [20].

4. RESULTS

We first test our EMD-based tracking scheme by focusing on a single time step tracking simulation with generated 10×10 pixel images containing five targets (i.e., pixels with value 1 in an otherwise all-zero image). We then generate a prediction of the target locations

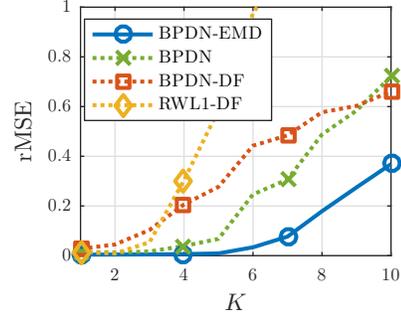


Fig. 2. One-step recovery error as a function of the sparsity level K . For any given value of K , EMD-regularized BPDN recovers the original signal with higher fidelity than competing algorithms.

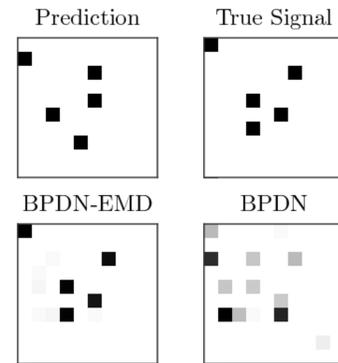


Fig. 3. Example recovery using EMD-regularized BPDN. The prediction of the true pixel locations is shown (top left) as well as the true pixel locations (top right). The recovered pixel locations via the EMD-regularized BPDN (bottom left) clearly portrays the true pixel locations much more accurately than the nearest competitor which we simulated, BPDN (bottom right).

consisting of an image wherein the targets locations are predicted to within one pixel of the true location, providing approximate location information. We recover the target locations using BPDN with EMD regularization as well as BPDN, BPDN-DF and RWL1-DF for comparison. In this and the following experiments, the parameters λ and γ are found using a grid search, and are chosen to yield the best possible performance for each algorithm. The relative mean-squared error between the image of inferred targets and the true target distribution, as calculated by $\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 / \|\mathbf{x}\|_2^2$, is shown in Figure 1. We notice that as we reduce the number of measurements of our signal, the EMD-regularized BPDN recovers the target locations with much higher fidelity. It is also interesting to note that standard BPDN actually outperforms the other two tracking algorithms, BPDN-DF and RWL1-DF. This is because the tracking regularizer in each of these algorithms uses the ℓ_p -norm which assigns the same penalty regardless of the location of the coefficients. Thus, the tracking term in fact misguides the recovery. In addition to varying the number of measurements, we also investigate how the recovery error varies with the sparsity level by varying the number of targets for a fixed number of measurements. Specifically, we fix the number of measurements to $M = 20$ and vary the number of targets from $K = 1$ to $K = 10$. The resulting rMSE values in Figure 2 demonstrate that for a given number of measurements, BPDN-EMD successfully tracks more tar-

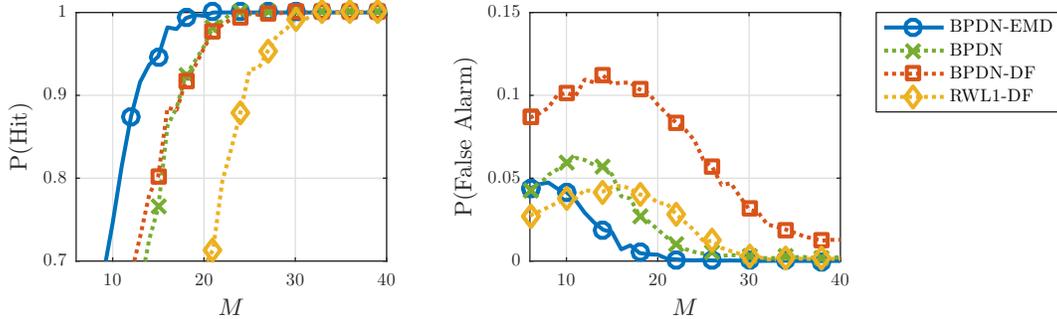


Fig. 4. Comparison of EMD-regularized BPDN in a detection task. A hit occurs if the support of the estimate contains the true support and the estimated coefficients on the true support exceed a given threshold. Similarly, we declare a false alarm when coefficients outside of the true support exceed this threshold. Similar to the rMSE calculation, the EMD-BPDN is better able to detect the support at lower measurement numbers than BPDN, BPDN-DF and RWL1-DF. Interestingly, we note that despite BPDN-DF having a worse rMSE than BPDN, the detection performance is much more comparable for the two algorithms, indicating that the model mismatch through the dynamics term is consistently contaminating BPDN-DF with many small erroneous values.

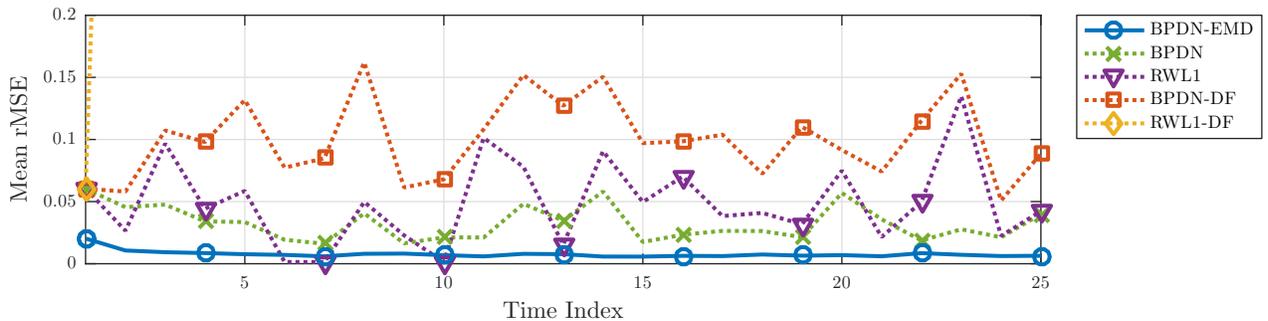


Fig. 5. Recovery of simulated moving pixels. We simulate a sequence of targets moving with pseudo-brownian-motion (active pixels move randomly at each time step to a neighboring location) and recover the sequence with various algorithms. The lowest relative mean-squared error is achieved by the EMD-regularized BPDN, followed by BPDN and RWL1 (which do not use any dynamics information.) BPDN-DF and RWL1-DF are given dynamics information that incurs a high penalty given the assumed innovations model, and thus their performance is worse than even the non-dynamics aware inference schemes.

gets than BPDN-DF or RWL1-DF.

As an example of the resulting inferred images, Figure 3 shows the image of true target locations, the image of target prediction, the EMD-based recovery and the BPDN estimate. Clearly using the EMD as a regularizer results in an estimate that better captures the target distribution in the image over the next-best estimator that we implemented. To further explore the utility of the EMD, we also consider this optimization scheme as a detection algorithm. We look at both the probability of false alarms and the probability of correct detection. As the number of measurements gets smaller, the EMD correctly detects all the correct target locations with minimal false alarms, as is shown in Figure 4.

Finally, we test the EMD-based dynamic filtering on target tracking over a longer sequence. We simulate targets that move around randomly, with the constraint that the targets do not move more than one pixel in each direction at each time-step. Again we use 10×10 images with 5 targets and we fix the number of measurements to $M = 20$. As shown in Figure 5, under these conditions, the EMD-regularized BPDN outperforms both static algorithms (BPDN and RWL1) as well as dynamic algorithms incapable of handling such innovations (BPDN-DF and RWL1-DF).

5. CONCLUSIONS

In this work we explore the utility of the EMD in the context of tracking algorithms. Specifically, we describe an optimization program that leverages the EMD as a regularizer and empirically explore its performance on signal estimation when approximate spatial data is available. We conclude that the EMD shows promise in increasing the performance for a number of image processing applications where tracking objects through a scene is desired. Thus it a worthwhile goal to further explore EMD-regularized trackers. To mitigate the inefficiency of solvers including the EMD, we briefly discuss the benefits of sparsity in reducing the computational burden. Other avenues, however, should be explored as well, such as continued work on our ADMM solver. Additionally, the EMD assumes non-negative signals, a restriction that should be addressed to make EMD-based tracking more widely applicable. As a final note, our experiments used a grid-search to set the optimization program parameters. Many alternatives exist for parameter selection (e.g., Bayesian type-II maximum likelihood,) which can be explored as well. Further inquiry into the most efficient parameter selection technique for EMD-based tracking will be particularly important given the increased computational cost of the EMD.

6. REFERENCES

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. D, pp. 35–45, 1960.
- [2] S. Haykin, "Kalman filters," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. John Wiley & Sons, Inc., 2001, pp. 1–22.
- [3] A. S. Charles and C. J. Rozell, "Spectral superresolution of hyperspectral imagery using reweighted ℓ_1 spatial filtering," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 3, pp. 602–606, March 2014.
- [4] N. Vaswani, "Kalman filtered compressed sensing," *Proc of ICIP 2008*, pp. 893–896, 2008.
- [5] A. S. Charles and C. J. Rozell, "Convergence of basis pursuit de-noising with dynamic filtering," *Proceedings the IEEE GlobalSIP*, Dec 2014.
- [6] Z. Zhang and B. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, no. 99, pp. 1–1, 2011.
- [7] M. S. Asif, L. Hamilton, M. Brummer, and J. Romberg, "Motion-adaptive spatio-temporal regularization (master) for accelerated dynamic mri," *Magnetic Resonance in Medicine*, vol. 70, no. 3, pp. 800–812, 2013.
- [8] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Transaction on Signal Processing*, vol. 61, no. 21, pp. 5270–5284, 2013.
- [9] E. Hall and R. Willett, "Dynamical models and tracking regret in online convex programming," in *Proceedings of the International Conference on Machine Learning*, 2013, pp. 579–587.
- [10] A. S. Charles, M. S. Asif, J. Romberg, and C. J. Rozell, "Sparsity penalties in dynamical system estimation," *Proc of the CISS*, March 2011.
- [11] R. Gupta, P. Indyk, and E. Price, "Sparse recovery for earth mover distance," in *48th Allerton Conference on Com., Cont., and Comp.*, 2010, pp. 1742–1744.
- [12] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *IEEE Proc. - Spec. Issue on App. of Compressive Sensing & Sparse Rep.*, Oct 2008.
- [13] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, Jul 2007.
- [14] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [15] A. Charles, A. Balavoine, and C. Rozell, "Dynamic filtering of time-varying sparse signals via ℓ_1 minimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5644–5656, 2016.
- [16] L. Schmidt, C. Hegde, and P. Indyk, "The constrained earth mover distance model, with applications to compressive sensing," in *10th Intl. Conf. on Sampling Theory and Appl. (SAMPTA)*, 2013.
- [17] D. Mo and M. F. Duarte, "Compressive parameter estimation with earth mover's distance via k-median clustering," in *SPIE Optical Engineering+ Applications*, 2013, pp. 88 581P–88 581P.
- [18] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar 2014.
- [19] —, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008—, pp. 95–110, <http://stanford.edu/~boyd/graph-dcp.html>.
- [20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.