USING BLOCK COORDINATE DESCENT TO LEARN SPARSE CODING DICTIONARIES WITH A MATRIX NORM UPDATE

Bradley M Whitaker and David V Anderson

Georgia Institute of Technology
Electrical and Computer Engineering
Atlanta, GA 30332, USA
b.whitaker@gatech.edu, anderson@gatech.edu

ABSTRACT

Researchers have recently examined a modified approach to sparse coding that encourages dictionaries to learn anomalous features. This is done by incorporating the matrix 1-norm, or $\ell_{1,\infty}$ mixed matrix norm, into the dictionary update portion of a sparse coding algorithm. However, solving a matrix norm minimization problem in each iteration of the algorithm causes it to run more slowly. The purpose of this paper is to introduce block coordinate descent, a subgradient-like approach to minimizing the matrix norm, to the dictionary update. This approach removes the need to solve a convex optimization program in each iteration and dramatically reduces the time required to learn a dictionary. Importantly, the dictionary learned in this manner can still model anomalous features present in a dataset.

Index Terms— Sparse coding, dictionary learning, matrix norms, anomaly detection

1. INTRODUCTION

Sparse coding is an increasingly important field of signal processing having many applications in audio and visual pattern recognition [1, 2, 3, 4]. Recently, sparse coding has also been explored in the context of anomaly detection [5, 6, 7, 8]. One goal of sparse coding is to uncover the underlying structure present in a dataset. Dictionary learning does this by minimizing the average reconstruction error given a set of observations, which requires the atoms that make up the training data to be present fairly uniformly.

This introduces a major difficulty in using sparse coding for anomaly detection. Anomalous data points may lose their abnormal behavior in reconstruction, making them hard to classify. Another problem associated with anomaly detection is the difficulty in producing training data. For example, anomalies can take the form of network intrusion [9], disease [10], or other unfortunate events [11] that are not easily or ethically replicated.

Researchers have explored ways to modify sparse coding to encourage dictionaries to learn anomalous features. In particular, the work presented in [12] demonstrates that incorporating a matrix norm into the dictionary update portion of the algorithm allowed the dictionary to recover a known anomalous feature in a dataset.

The purpose of this paper is to address a significant limitation to the algorithm presented in [12] by introducing block coordinate descent [13, 14] to minimize the matrix norm in the dictionary update. We show that using this approach reduces the time required to learn a dictionary by an order of magnitude while preserving the algorithm's ability to model anomalous features.

2. BACKGROUND

2.1. Dictionary Learning

The goal of sparse coding dictionary learning is to find the dictionary and sparse coefficient vectors that best explain a set of input vectors. Under assumptions of independent coefficient vectors and Gaussian zero-mean noise, this reduces to jointly solving for a dictionary ($\mathcal{D}$) and sparse coefficient vectors ($\{\boldsymbol{\alpha}_n\}_{n=1,\ldots,\mathrm{N}}$) given many training samples, ($\{\boldsymbol{x}_n\}_{n=1,\ldots,\mathrm{N}}$) [15, 16]:

$$\min_{\mathcal{D}\in\mathcal{C},\{\boldsymbol{\alpha}_n\}\in\mathbb{R}^{\mathrm{K}}} \quad \frac{1}{\mathrm{N}}\sum_{n=1}^{\mathrm{N}}\frac{1}{2}\left\|\boldsymbol{x}_n-\mathcal{D}\boldsymbol{\alpha}_n\right\|_2^2+\lambda\left\|\boldsymbol{\alpha}_n\right\|_1. \quad (1)$$

In this formulation, λ is a fidelity-sparsity tradeoff parameter and $\mathcal{C}$ is the set of matrices in $\mathbb{R}^{\mathrm{M}\times\mathrm{K}}$ whose columns have ℓ_2-norm less than one. Without that constraint, the columns of $\mathcal{D}$ could grow arbitrarily large, allowing each $\boldsymbol{\alpha}_n$ to be arbitrarily small and effectively removing the ℓ_1 term in the objective function. We solve Eq. (1) using the alternate minimization algorithm, outlined in Algorithm 1 [17, 18].

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650044. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Algorithm 1 Alternate Minimization

Inputs: Signals $\{\boldsymbol{x}_n \in \mathbb{R}^{\mathrm{M}}\}_{n=1,...,\mathrm{N}}$, initial dictionary $\mathcal{D}_0 \in \mathcal{C}$, regularization term λ, number of iterations I.

1: Initialize $\mathcal{D} \leftarrow \mathcal{D}_0$
2: **for** $i = 1, ..., \mathrm{I}$ **do**
3: **for** $n = 1, ..., \mathrm{N}$ (*in parallel*) **do**
4: Calculate coefficient vectors:
$\boldsymbol{\alpha}_n = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \ \frac{1}{2} \|\boldsymbol{x}_n - \mathcal{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$
5: **end for**
6: Update dictionary:
$\mathcal{D} = \underset{\mathcal{D}\in\mathcal{C}}{\operatorname{argmin}} \ \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \frac{1}{2} \|\boldsymbol{x}_n - \mathcal{D}\boldsymbol{\alpha}_n\|_2^2$
7: **end for**
8: **return** $\mathcal{D}$

2.2. Matrix Norm

The averaged ℓ_2 dictionary update (Step 6 in Algorithm 1) is the most natural update under the assumptions described earlier. However, it has been shown previously that using a matrix norm can force the dictionary to consider minimizing a maximum error rather than the average error [12]. The modified dictionary update (step 6 in Algorithm 1) using the $\ell_{1,\infty}$ norm becomes

$$\mathcal{D} = \underset{\mathcal{D}\in\mathcal{C}}{\operatorname{argmin}} \ \|\mathcal{X} - \mathcal{D}\mathcal{A}\|_{1,\infty}. \tag{2}$$

The matrix 1-norm presented in [12] is equivalent to the $\ell_{1,\infty}$ mixed matrix norm [19, 20]. To avoid confusing the subordinate matrix 1-norm with the vector ℓ_1 norm, we choose to use the mixed norm notation in this paper. It is defined as

$$\|\boldsymbol{A}\|_{1,\infty} \equiv \max_{1\le n\le \mathrm{N}} \sum_{m=1}^{\mathrm{M}} |\boldsymbol{A}_{m,n}| = \big|\big| \|\boldsymbol{a}_1\|_1 \cdots \|\boldsymbol{a}_{\mathrm{N}}\|_1 \big|\big|_\infty, \tag{3}$$

where the vector $\boldsymbol{a}_n$ is the n^{th} column of $\boldsymbol{A}$. In words, $\|\boldsymbol{A}\|_{1,\infty}$ returns the maximum absolute column sum of $\boldsymbol{A}$. Therefore, minimizing $\|\mathcal{X} - \mathcal{D}\mathcal{A}\|_{1,\infty}$ will minimize the maximum absolute deviation of any given training vector, encouraging the features to explain every data point, regardless of how infrequently it appears.

2.3. Block Coordinate Descent

One implementation of the alternating minimization algorithm updates the dictionary by running a few iterations of gradient descent [21]. This reduces the complexity of each main iteration of the algorithm. For the standard average ℓ_2 dictionary update, the gradient is well-defined. The $\ell_{1,\infty}$ update, however, includes a max operator and an absolute value. While both of these are convex, neither has a smooth gradient. In [12], the authors used CVX [22] to solve for the dictionary update. This caused the dictionary to be learned ten times more slowly (in wall time) than a standard dictionary.

In this paper we use block coordinate descent to perform the matrix norm update [13, 14]. In block coordinate descent, the columns with the largest ℓ_1 norms are identified and the dictionary is updated to reduce the ℓ_1 norms of those columns. The dictionary update then becomes

$$\mathcal{D}_{\text{new}} = \mathcal{D}_{\text{cur}} + \delta \operatorname{sign}\left(\mathcal{X}_I - \mathcal{D}_{\text{cur}}\mathcal{A}_I\right)\mathcal{A}_I^\top, \tag{4}$$

where I is the set of indices corresponding to the columns with the largest ℓ_1 norms, and the matrices $\mathcal{X}_I$ and $\mathcal{A}_I$ are formed by extracting only those columns from $\mathcal{X}$ and $\mathcal{A}$. Equivalently, $\mathcal{X}_I$ can be thought of as being formed using the training vectors that correspond to a large error. In our implementation, we selected the step size δ to be 0.0005.

Because block coordinate descent only requires simple operations, we expect the bulk of the computational time to be spent calculating the sparse coefficients rather than updating the dictionary.

3. EXPERIMENT

The goal of this paper is to demonstrate that learning a dictionary using block coordinate descent of the $\ell_{1,\infty}$ mixed matrix norm in the dictionary update step of Algorithm 1 can recover known anomalies in a dataset. In addition, we wish to show that it can be done much more quickly than by minimizing the $\ell_{1,\infty}$-norm completely at each iteration.

3.1. Dataset Description

We created 6 different datasets with different structure to explore the effectiveness of the matrix norm subgradient descent dictionary update. All datasets are 3-sparse linear combinations of 16 different vectors in $\mathbb{R}^{64}$. The 16 underlying vectors associated with the dataset are denoted $\{b_1 ... b_{16}\}$ and are referred to as "basis vectors."

The first four datasets used cosines as the 16 basis vectors: $b_i(t) = \mathrm{A}\cos(2\pi(i)t)$. The variable t consists of 64 linearly-spaced points over [0,1]. The constant A is chosen to normalize the basis vector so $\|b_i\| = 1$. In two of the datasets, random gaussian noise was added so the SNR of the training samples was 30 dB. In one of the clean datasets and one of the noisy datasets, one basis vector chosen to be an anomaly, and was included in a single training vector.

The last two datasets used 16 random vectors as the basis set for the training samples. One of the datasets includes an anomalous basis vector which only appears once in the training data. The other dataset uses all 16 basis vectors with equal probability.

3.2. Dictionary Description

For each dataset, we learned three different dictionaries. Each dictionary had 16 elements to match the number of known basis vectors in the datasets. The first dictionary follows the

Table 1. Summary of results

Dictionary Update	ℓ_2		$\ell_{1,\infty}$-CVX		$\ell_{1,\infty}$-BCD	
Database	Recovers	Error	Recovers	Error	Recovers	Error
Sinusoid						
No Noise, No Anomalies	✓	$5.27 \pm 1.29\%$	✓	$5.96 \pm 1.60\%$	✓	$8.48 \pm 1.55\%$
Noise, No Anomalies	✓	$6.40 \pm 1.87\%$	✓	$6.61 \pm 1.97\%$	✓	$8.18 \pm 1.88\%$
No Noise, One Anomaly	misses b_{11}	$5.56 \pm 3.86\%$	✓	$5.77 \pm 1.60\%$	✓	$7.98 \pm 1.53\%$
Noise, One Anomaly	misses b_{11}	$6.54 \pm 3.63\%$	✓	$6.99 \pm 2.10\%$	✓	$8.22 \pm 1.91\%$
Random						
No Anomalies	misses all	$3.25 \pm 1.17\%$	✓	$1.18 \pm 0.31\%$	✓	$3.05 \pm 0.59\%$
One Anomaly	misses all	$3.12 \pm 1.09\%$	✓	$1.34 \pm 0.39\%$	✓	$2.79 \pm 0.47\%$

format described in Algorithm 1, but the dictionary is updated using several iterations of gradient descent. The second dictionary is learned by solving the $\ell_{1,\infty}$-norm minimization problem at each iteration using CVX [22]. The third dictionary is updated using several iterations of block coordinate descent with respect to the $\ell_{1,\infty}$-norm, as described in Section 2.3. We denote these three dictionaries as the "ℓ_2," "$\ell_{1,\infty}$-CVX," and "$\ell_{1,\infty}$-BCD" dictionaries, respectively.

4. RESULTS

4.1. Dictionary Performance

Table 1 reports a summary of how well each dictionary recovered the original basis vectors and models the training vectors. The 'Recovers' column associated with each dictionary update method contains a check mark (✓) if each dictionary recovers all basis vectors. We consider a basis vector to be recovered if the angle between the basis vector and the closest dictionary element is within 10°. We note that occasionally a pair of basis vectors is 'jointly recovered' by two dictionary elements (e.g. $b_1 = d_1 + d_2$ and $b_2 = d_1 - d_2$). We observed that the $\ell_{1,\infty}$ methods often modeled the anomalous training vector rather than the hidden dictionary element. Because of this, we relaxed the definition for recovery with respect to the anomalous basis vector and designate it as 'recovered' if it appears in combination with other recovered basis vectors.

The 'Error' column of Table 1 reports the average percent error, as well as the standard deviation ($\mu \pm \sigma$), of the reconstructed vectors with respect to the original training vectors. The table shows that the ℓ_2 and $\ell_{1,\infty}$-CVX methods have lower average errors than the $\ell_{1,\infty}$-BCD method for the sinusoidal databases. For the random database, the $\ell_{1,\infty}$-CVX outperforms the other two methods.

Another way to determine the success of a dictionary in modeling the data is by looking at the Euclidean angle between the dictionary elements and the original basis vectors. Fig. 1 shows examples of this for two of the databases: the clean sinusoid-basis dataset with one anomaly and the random-basis dataset with one anomaly. For the sinusoid dataset, all dictionaries recover most of the original basis vectors. We notice, however, that the ℓ_2 method fails to recover the anomalous basis vector, b_{11}. Instead, it recovers what appears to be a linear combination of b_{10} and b_{14}. We also note that the $\ell_{1,\infty}$-CVX method combines b_1 and b_5. Notably, both the $\ell_{1,\infty}$-CVX and the $\ell_{1,\infty}$-BCD methods recover the anomalous basis vector. It is not a problem that both recover b_{11} in combination with another basis vector, since the 'extra' basis vector can be subtracted out in a sparse reconstruction.

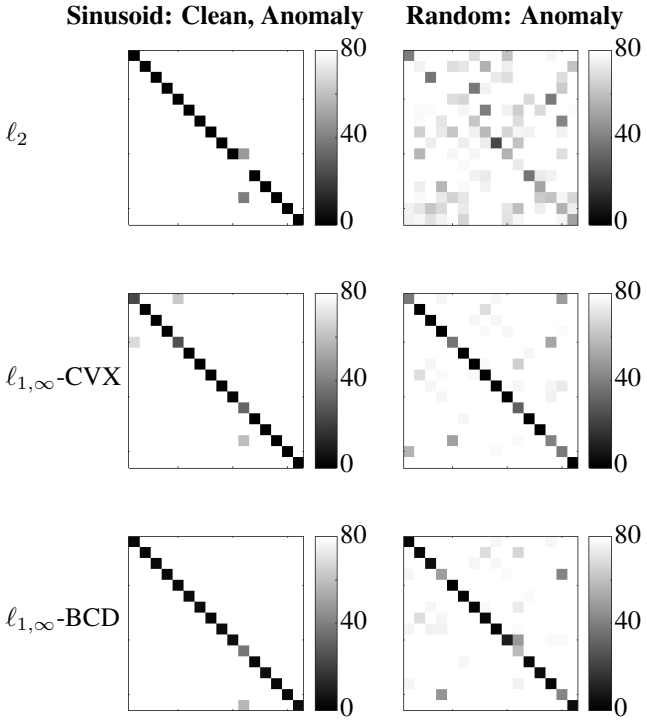

Fig. 1. Angle between original basis vectors and learned dictionary elements. A low angle (black) along the diagonal means that the dictionary was able to recover that basis vector. The angles are shown for all three dictionary update methods on two databases—the sinusoid-basis database with no noise and one anomaly and the random-basis database with one anomaly.

Table 2. Reconstruction Error of Anomalous Training Vector

Update Method	Sinusoid	Noisy Sinusoid	Random
ℓ_2	85.78%	76.50%	5.68%
$\ell_{1,\infty}$-CVX	3.81%	3.79%	0.84%
$\ell_{1,\infty}$-BCD	12.41%	12.77%	3.18%

Table 3. Approximate Iteration Time

Update Method	Single Iteration	500 Iterations
ℓ_2	$\approx$ 6 seconds	$\approx$ 1 hour
$\ell_{1,\infty}$-CVX	$\approx$ 90 seconds	$\approx$ 12.5 hours
$\ell_{1,\infty}$-BCD	$\approx$ 6 seconds	$\approx$ 1 hour

The second column shows the angles between the learned dictionaries and the original basis vectors for the random-basis dataset with one anomaly. Again, we see clear structure along the diagonal in the $\ell_{1,\infty}$-CVX and $\ell_{1,\infty}$-BCD dictionaries. This structure is missing in the ℓ_2 dictionary. Despite this, we see from Table 1 that the average recovery error associated with this dictionary is only 3.12%. Therefore, while the dictionary did not recover the original basis, it was able to learn a dictionary that represents the training data.

4.2. Anomaly Reconstruction

In addition to looking at how well the dictionaries perform in recovering all original basis vectors, it is interesting to analyze how well each dictionary models the anomalous basis vector present in each training set. Fig. 2 shows plots of the training vector containing the anomaly for each dataset, as well as the reconstructed training vector associated with each dictionary. For both sinusoid datasets, the ℓ_2 dictionary fails to reconstruct the training vector; it can only reconstruct the portion of the training vector associated with the non-anomalous basis vector. For the random dataset, all three methods are able to reconstruct the training vector. Recall, that while the ℓ_2 method is able to reconstruct all training vectors in the random dataset, it fails to learn the original training vectors.

Table 2 gives a quantitative analysis of the reconstruction error associated with the anomalous training vector. It is interesting to note that the ℓ_2 has the worst recovery error in all three datasets. The table also shows that the $\ell_{1,\infty}$-BCD method has a fairly large recovery error of about 12%. In contrast, the $\ell_{1,\infty}$-CVX method has an error of about 4%. However, when looking at Fig. 2, the plots show that both of these dictionaries are able to visually reconstruct the training vectors from the sinusoid datasets.

4.3. Computational Efficiency

One large motivation for exploring the subgradient descent version of the matrix norm was that the $\ell_{1,\infty}$-CVX method was very computationally expensive [12]. This method requires solving a convex optimization program at each iteration. Replacing this requirement with computing several iterations of block coordinate descent dramatically reduced the time it took to learn a dictionary, while still producing a dictionary that can model anomalous features of a dataset. Table 3 reports the approximate time it took to run a single iteration of Algorithm 1 using each dictionary update method, as well as the approximate time it took to learn the dictionary (after 500 iterations). For reference, all experiments were run in MATLAB on a desktop computer with a quad-core i7 processor clocked at 3.4 GHz with 16 GB RAM.

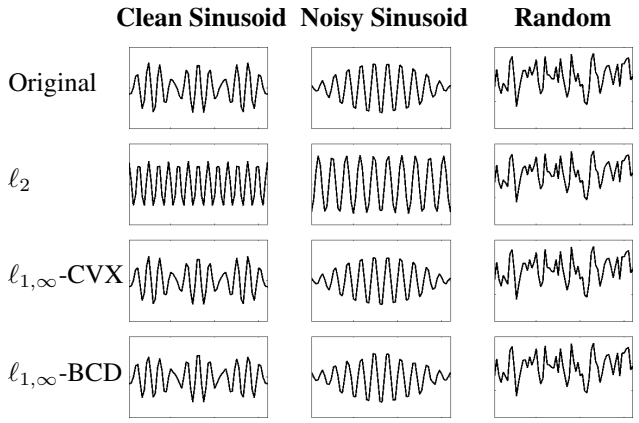

Fig. 2. Original and reconstructed anomalous training vectors. The top row shows the training vector in each dataset that contained the anomalous basis vector. The second row shows the reconstruction of this training vector using the dictionary learned with the standard ℓ_2 update. The third row shows the reconstruction using the dictionary learned with the $\ell_{1,\infty}$-CVX dictionary update. The bottom row shows the reconstruction using the dictionary learned with the $\ell_{1,\infty}$-BCD update. The reconstruction percentage errors corresponding to these vectors are presented in Table 2.

5. CONCLUSION AND FUTURE WORK

The work presented in this paper shows a fast method for learning a sparse coding dictionary that can model known anomalous features in a dataset. The algorithm is based on updating the dictionary using several iterations of block coordinate descent with respect to the $\ell_{1,\infty}$ mixed matrix norm. We have demonstrated successful recovery on two types of datasets: one with a set of sinusoidal features, and another with a set of random features.

While we have shown that our method can quickly learn a dictionary that models anomalous features, we recognize some limitations in this work. Moving forward, it would be interesting to test the usefulness of the learned dictionary in a classification setting. In particular, this method may be a good tool for learning features in an unsupervised manner that can then be used to classify anomalous training samples. It would also be beneficial to investigate matrix norm dictionary learning on imbalanced datasets in general, not focusing solely on anomaly recovery.

6. REFERENCES

[1] Bruno A Olshausen and David 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, 1996.

[2] K. Engan, S.O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, 1999, vol. 5, pp. 2443–2446.

[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[4] Wei Dai, Tao Xu, and Wenwu Wang, "Dictionary learning and update based on simultaneous codeword optimization (SimCO)," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 2037–2040.

[5] T Andrysiak and L Saganowski, "Anomaly detection system based on sparse signal representation," *Image Processing & Communications*, vol. 16, no. 3-4, pp. 37–44, 2012.

[6] D Carrera, G Boracchi, A Foi, and B Wohlberg, "Scale-invariant anomaly detection with multiscale group-sparse models," in *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3892–3896, IEEE.

[7] R. Sarkar, A. Vaccari, and S. T. Acton, "SS-PARED: Saliency and sparse code analysis for rare event detection in video," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, July 2016, pp. 1–5.

[8] B. T. Carroll, B. M. Whitaker, W. Dayley, and D. V. Anderson, "Outlier learning via augmented frozen dictionaries," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1207–1215, June 2017.

[9] Aleksandar Lazarevic, Levent Ertöz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection.," in *SDM*. SIAM, 2003, pp. 25–36.

[10] Weng-Keen Wong, Andrew Moore, Gregory Cooper, and Michael Wagner, "Bayesian network anomaly pattern detection for disease outbreaks," in *ICML*, 2003, pp. 808–815.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, July 2009.

[12] Bradley Whitaker and David Anderson, "Learning anomalous features via sparse coding using matrix norms," in *2015 IEEE Signal Processing Workshop (SPW2015)*, Salt Lake City, USA, Aug 2015, pp. 196–201.

[13] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, Jun 2001.

[14] Han Liu, Mark Palatucci, and Jian Zhang, "Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery," in *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, 2009, ICML '09, pp. 649–656, ACM.

[15] Bruno A Olshausen and David J Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[16] I. Tosic and P. Frossard, "Dictionary learning," *Signal Processing Magazine, IEEE*, vol. 28, no. 2, pp. 27–38, March 2011.

[17] Julien Mairal, Francis Bach, and Jean Ponce, "Sparse modeling for image and vision processing," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[18] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J.C. Platt, and T. Hoffman, Eds., pp. 801–808. MIT Press, 2007.

[19] A. Benedek and R. Panzone, "The space l^p , with mixed norm," *Duke Math. J.*, vol. 28, no. 3, pp. 301–324, 09 1961.

[20] Matthieu Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303 – 324, 2009.

[21] Adam S Charles, Bruno A Olshausen, and Christopher J Rozell, "Learning sparse codes for hyperspectral imagery," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 5, pp. 963–978, 2011.

[22] Michael Grant and Stephen Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.