A Unified FOCUSS Framework for Learning Sparse Dictionaries and Non-squared Error.

Brandon Burdge, Kenneth Kreutz-Delgado
Dept. of Electrical and Computer Engineering
University of California San Diego
bburdge@ucsd.edu, kreutz@ece.ucsd.edu

Joseph Murray
jfmurray@jfmurray.org

***Abstract*—FOCUSS is an Iteratively Reweighted Least Squares approximation used to find the inverse solution of an underdetermined linear system when the source vector is assumed to be sparse. It also provides an iterative descent method used to solve for an unknown dictionary. We describe three extensions to the FOCUSS model: First a choice of generalized p-norm reconstruction error which corresponds to differing assumptions on the cost of errors. Second the use of a constraint which encourages sparsity on the dictionary atoms, and third the combination of both sparsity on dictionary atoms and generalized reconstruction error to form one unified framework for solving a wide set of sparsity requirements on sources, on loadings, and on error. Finally, we describe a practical set of algorithms for learning dictionaries and source vectors under each of these model assumptions, and show experimental results using these algorithms.**

I. Sparse Coding and FOCUSS

The area of sparse signal processing and compressive sensing has become increasingly pervasive in a number of application domains, including image compression and capture [10], [7], and audio processing [12]. Describing and extending new algorithms in this field can provide new directions for application, or improved performance in currently active domains. We start with the overcomplete linear model $y = Ax + \nu$, Where y is a densely filled observation vector of the world, A is an overcomplete dictionary, x is a sparse loading onto the columns of A, and ν is random noise. We can formulate the problem of choosing an A and x that represent a dataset as an optimization problem over an appropriate loss function and constraint set [1], [5]. The particular choice of loss function and constraints represents a condensation of beliefs and prior knowledge about the problem domain, and further, directly informs the development of algorithms to solve, exactly or approximately, for unknown quantities.

We state the Basic Sparse Representation Problem (BSRP) for the known overcomplete linear model as,

$$\begin{aligned} &\arg\min_x ||y - Ax|| \\ &\text{such that } ||x||_0 \leq T, \end{aligned} \tag{1}$$

where $||x||_0$ is the *zero-pseudonorm* of x, a somewhat abusive notation for the count of non-zero elements, which is not a proper norm as it does not obey the triangle inequality. This problem is generally intractable, since finding the minimum under the zero-norm constraint requires a combinatorial search over all possible patterns of non-zeros in x having less than T active values at a time. There are known to be multiple approximation methods to approach this problem [1].

In particular, we consider relaxing the zero-norm constraint to a smoother functional form over which to optimize. We choose the p-pseudonorm family used in the Focal Undetermined System Solver (FOCUSS) algorithm which are defined according to $||x||_p^p = \sum_i |x_i|^p$, which for $0 < p \leq 1$ are known to promote sparsity [14]. Further, we replace the constraints with a loss function term with appropriate regularization term, and arrive at the FOCUSS loss function,

$$\mathcal{L}_{FOCUSS}(x) = ||y - Ax||_2^2 + \lambda ||x||_p^p. \tag{2}$$

This has a known approximation algorithm via Iteratively Reweighted Least Squares (IRLS) [4], [8]. We will also use a simple extension to this basic algorithm proposed by Chartrand and Yin [2] and investigated by Wipf and Nagarajan [13] which adds a small term to the IRLS weighting factor that provides significantly improved performance with very little computational expense.

Since we utilize the results IRLS algorithm later, we show the underlying principal used in the approach here. For a p-pseudonorm of a generic vector b,

$$\begin{aligned} ||b||_p^p &= \sum_i |b_i|^p \\ &= \sum_i |b_i|^2 |b_i|^{p-2} \\ &= \sum_i w_i |b_i|^2 \\ &= b^T W(b) b, \end{aligned} \tag{3}$$

for $W(b) = \text{diag}(|b_i|^{p-2})$. Replacing b in $W(b)$ with a current estimate $\hat{b}$ essentially approximates the p-pseudonorm by a weighted 2-norm. For the purpose of optimization this presents a way around the difficulties of directly attacking the p-pseudonorm term, producing an iterative algorithm based on alternately updating the objective value and then the weighting matrix.

[0]With thanks to the National Science Foundation for partial support. NSF Grant CCF-0830612

The BSRP (2) assumes that A is known and minimizes over x, however there are many domains where A is not known. Several methods for learning A from data are known[1]. Dictionary learning algorithms generally work by an alternating minimization algorithm which involves alternating between the sparse solutions for x given by (2) and a solution for the unknown matrix by solving

$$\begin{aligned} &\arg\min_A ||y - Ax||_2 \\ &\text{such that } ||a_j||_2 = 1;\ \forall j, \end{aligned} \tag{4}$$

assuming known sparse vectors x. This can be solved either through an analytic form, or gradient descent using an algorithm developed in [5]. We restate the Column Normalized Dictionary Learning (CNDL) results of [5] here for later application. For a given set of observation and sparse source estimates $[y_k, x_k]_{k=1}^N$ we can define $\Sigma_{xy} = \sum_i x_i y_i^T$ and $\Sigma_x = \sum_i x_i x_i^T$. Then updates for A are given as,

$$\begin{aligned} \delta \mathrm{A} &= \mathrm{A}\Sigma_x - \Sigma_{xy} \\ a_i &\leftarrow a_i - \gamma(I - a_i a_i^T)\delta a_i. \end{aligned} \tag{5}$$

Where γ is a learning rate, and a_i and δa_i are the ith columns of A and δA respectively.

For domains where the 2-norm error shown in (5) is appropriate FOCUSS can provide good performance in finding sparse solutions on a known, or unknown, dense dictionary. However, there are many domains where performance is better measured by alternative measures. For example, in digital communications it is more common to use the number of bit errors, this corresponds to the count of non-zero errors, which is the value of $||y - Ax||_0$. Similarly, in domains where observations are valued on a discrete field, the 1-norm distance is perhaps a more valuable error criterion, since it penalizes small deviations from the discrete values more strongly than the 2-norm distance.

II. RREFOCUSS

We propose altering the standard FOCUSS loss function in the following way,

$$\mathcal{L}_{RREFOCUSS}(x) = ||y - Ax||_r^r + \lambda ||x||_p^p. \tag{6}$$

We call this RREFOCUSS for R-norm Reconstruction Error FOCUSS. Appropriately extending the IRLS methodology described above, we can derive an iterative update algorithm for optimizing this loss function.

$$\begin{aligned} \hat{x}(k+1) &\leftarrow [\Pi^{-1}(k)A^T W(k) A + \lambda I]^{-1} \cdot \\ &\quad \cdot \Pi^{-1}(k) A^T W(k) y \end{aligned} \tag{7}$$

$$\begin{aligned} \Pi^{-1}(k+1) &\leftarrow \mathrm{diag}(|x_i + \delta|^{2-q}) \\ W(k+1) &\leftarrow \mathrm{diag}(|(y - Ax)_i + \epsilon|^{r-2}), \end{aligned}$$

where ϵ is a small parameter used to ensure bounded values, and δ is the IRLS modification suggested by [2]. The variable r provides a parameter to tune for selecting reconstruction error terms, between a Laplacian error assumption at $r = 1$ to a Hamming distance approximation as $r \to 0$.

III. R-norm Dictionary Learning

With the adjusted RREFOCUSS loss function, we could still perform the CNDL algorithm steps to learn a dictionary. However, this is counter-intuitive, as we wish to have agreement in the norm on reconstruction error between the new RREFOCUSS sparse coding stage, and the dictionary update stage. Hence, we will revise the dictionary update steps, starting with the more general loss function on A given by

$$\mathcal{L}(A) = \sum_i ||y_i - Ax_i||_q^q = \sum_i ||y_i - Ax_i||_{\mathrm{W}_i}^2, \tag{8}$$

for a known set of pairs of observations and sparse source vectors $\{y_i, x_i\}_{i=1}^N$, with one weighting matrix for each pair. Taking the partial derivative with respect to A assuming fixed weighting matrices, and letting $\Sigma_{x_i y_i} = x_i y_i^T$ $\Sigma_{x_i} = x_i x_i^T$,

$$\begin{aligned} \frac{\partial \mathcal{L}(A)}{\partial A} &= \sum_i -2\frac{\partial}{\partial A}\mathrm{tr}(\mathrm{W}_i A \Sigma_{xy_i}) \\ &\quad + \frac{\partial}{\partial A}\mathrm{tr}(\mathrm{W}_i A \Sigma_{x_i} A^T)) \\ &= \sum_i -2\Sigma_{xy_i}\mathrm{W}_i + 2\Sigma_{x_i} A^T \mathrm{W}_i \end{aligned} \tag{9}$$

where the transpose of this expression forms the gradient of $\mathcal{L}$ with respect to A. We can further solve for normal equations of fixed points,

$$\begin{aligned} \sum_i \mathrm{W}_i \Sigma_{xy_i} &= \sum_i \mathrm{W}_i A \Sigma_{x_i} \\ \sum_i \mathrm{vec}(\mathrm{W}_i \Sigma_{xy_i}) &= \sum_i \mathrm{vec}(\mathrm{W}_i A \Sigma_{x_i}) \\ \sum_i (\Sigma_{x_i} \otimes \mathrm{W}_i)\mathrm{vec}(A) &= \sum_i \mathrm{vec}(\mathrm{W}_i \Sigma_{xy_i}). \end{aligned} \tag{10}$$

To solve explicity here requires inversion of $\sum_i \Sigma_{x_i} \otimes \mathrm{W}_i$. For sparse x this matrix is most often either rank-deficient or poorly conditioned, leading to a very expensive numerical pseudo-inversion which for problems of most sizes is not computationally reasonable. Thus, it seems we are left with gradient descent, for most practical purposes.

A. Stability Issues for RREDL

However, we run into issues with stability of the gradient descent algorithm. We first note that the use of gradient descent combined with Iteratively Reweighted Least Squares forms a conjugate gradient algorithm that has properties similar to affine-scaling transformation algorithms (AST), as shown by Kreutz-Delgado, et. al. [8]. A difficulty with AST is that the algorithm requires the gradient update steps to remain within the positive orthant in the loss-function space [3]. In the event that a step breaks this restriction the algorithm moves

into an unstable region and will wander off. The updates of the weighting matrix W should remain within this restricted region, and for reasonable gradient values this holds. However, a non-trivial difficulty when learning A with an r-pseudonorm comes from the sharply peaked nature of the pseudonorm loss, and shown in figure (1). For regions away from minima the gradient value is small, but as you near a minima the gradient values increase sharply. Coupled with the large number of degrees of freedom available to the matrix A, this leads to erratic gradient behavior that can result in instability in the IRLS algorithm. Attempts to control this behavior to date have included renormalizing update steps to have either unit Frobenius norm, or columns with unit 1-norm. While these steps do prevent the algorithm from reaching unstable regions, they seem to retard learning and the dictionary does not improve by any reasonably significant amount. Hence, for our experiments we use the CNDL algorithm despite the mismatch in loss functions.

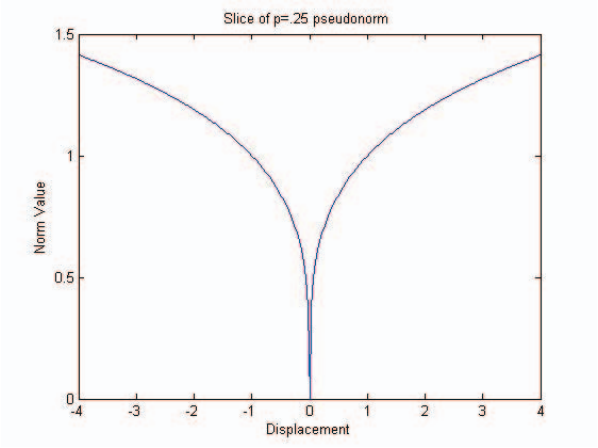

Fig. 1. Comparison of FOCUSS and RREFOCUSS reconstructions

IV. Sp-DL: Sparse Dictionary Atoms

For some purposes the interpretability of the dictionary atoms is important, and requiring sparsity within the atoms is one means of improving interpretability. Towards this end we introduce a term,

$$\mathcal{SP}(A) = \sum_j ||a_j||_q^q \tag{11}$$

to the loss function which penalizes the p-pseudonorm on columns of A. To minimize this term we again make the IRLS approximation and replace the pseudonorm with a weighted 2-norm such that $\Theta_j = \text{diag}(|a_{ij} + \epsilon|^{2-q})$, this gives a term of the form $\mathcal{SP}(A) = \gamma \sum_j ||a_j||_{\Theta_j}^2$. This term can be arranged into a quadratic form as $\mathcal{SP}(A) = \gamma \sum_j e_j^T A^T \Theta_j A e_j$, where e_j represents the j-th canonical basis vector. Taking the partial derivative with respect to A gives,

$$\frac{\partial \mathcal{SP}(A)}{\partial A} = \gamma A^T \sum_j \Theta_j, \tag{12}$$

This term may be directly added to the CNDL derivative function from [5], and an appropriate δA formed, providing new update steps which have the added sparsity constraint. Alternatively, we find that we can perform gradient descent using this adjusted gradient term without a column normalization constraint, as the added sparsity constraint provides sufficient regularization, we call this Sp-DL.

V. Experiment

To verify the utility of our extensions to the FOCUSS algorithm we perform several tests. We choose to test on the domain of Go board configurations. The game of Go is simple in it's presentation, but remarkly complex in strategy, and represents one of the largest challenges to computer game playing [6]. The configuration of a Go game board consists of a 19×19 grid, with each intersection point being a playable location, for which there could be either an empty space, or a white, or black piece or stone. Two players take turns placing pieces with the goal of surrounding and capturing opponent stones. Deciding when a game has ended, and assigning a score to choose a winner is a complicated procedure for which there has been previous worked developing neural network game scoring systems [11].

There have been previous examples of intelligent machines designed to learn to play go through various strategies, and a common element of these systems is the use of hand-crafted sets of small groupings of board pieces. Particular board configurations are then represented as being formed from combinations of these groupings and single pieces [9]. We consider that this closely fits the sparse coding paradigm, with hand-crafted overcomplete dictionaries, and a method for coding board configurations on them. Further, the board configurations lie in a discrete-valued space, with various possible encodings of empty, white, or black stone being possible at each grid point. For our examples we numerically represent black stones as -1, white as 1, and empty spaces as 0.

TABLE I
RREFOCUSS Go patch Average Reconstruction Errors

Baseline FOCUSS	r = 1 FOCUSS	r = 1 Learned	r = 0.5 Learned	r = 0.2 Learned
2.6	0.9	0.7	0.72	0.73

This gives us an ideal test situation for our RREFOCUSS and Sp-DL algorithms. First, since we are interested in reconstructions of game boards that minimize the number of pieces placed incorrectly, we wish to have an error objective which penalizes the count of errors, instead of the magnitude. This is exactly the problem that RREFOCUSS targets. Secondly, if we wish to learn a dictionary, from data, of representative groupings of pieces then we are likely to want those groupings to be of a small number of non-zeros which are matched to our discrete-valued domain, though we will not explicitly force this. This suggests applying the Sp-DL algorithm.

To test this we use an internal collection of recorded Go games from which individual board configurations have been taken out for each game step. Since we are interested in finding localized patterns, we divided the boards into 5×5 patches, with a 1 piece overlap on the last column and row. From this collection of patches we extracted 5000 training, and 1000 test examples, taken uniformly at random without replacement. All tests are run with the source sparsity parameter p set to 1, and for source vectors containing 25 non-zero elements.

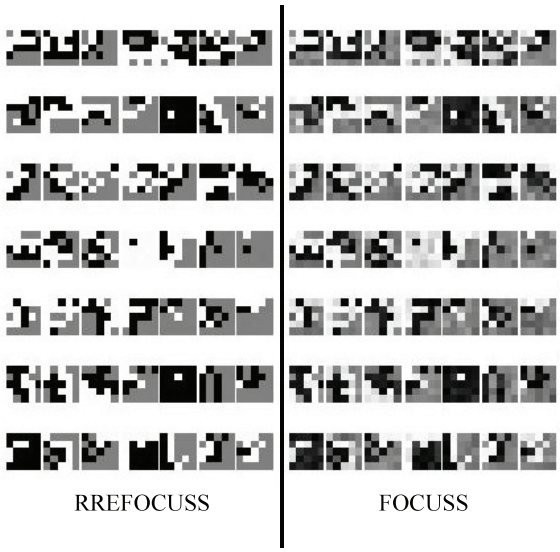

Fig. 2. Comparison of FOCUSS and RREFOCUSS reconstructions

The first experiment investigates the ability for the RREFOCUSS algorithm to reconstruct these patches, the error reported is the average 1-norm of the difference between observed and reconstructed patches over the test set. The choice of 1-norm error here is to give a rough estimate of expected worst case number of piece errors per patch, in the sense that if all the observed discrepancy was concentrated in a few places after a thresholding pass you would expect to find a number of errors similar to the this error. We do not show after threshold piece error rates here since such rates were so low for all methods as to not provide any meaningful comparison of algorithm performance. Further, we note that the second column in TABLE I is the RREFOCUSS algorithm applied on the test set using the dictionary learned with the standard FOCUSS algorithm on the training set. The other dictionaries were learned using RREFOCUSS for the sparse update steps, and CNDL for the dictionary updates.

TABLE II
RREFOCUSS AND SP-DL

FOCUSS q = 1 Sp-DL	r = 1 q = 1 Sp-DL	r = 1 q = 0.8 Sp-DL	r = 0.5 q = 1 Sp-DL
2.83	1.41	1.44	1.70

We note that the performance of the RREFOCUSS system is a definite improvement on the original FOCUSS for this purpose. Figure (2) visually shows the different in reconstruction, note the increase noise in the 2-norm FOCUSS.

We also test the ability to learn a dictionary with sparse elements, to provide more interpretable atoms. TABLE II shows the results of learning dictionaries with sparse elements, as well as the combination of sparsity imposed on all three points, sparse sources, sparse atoms, and sparse reconstruction error. We note that error levels rise when a sparse dictionary is imposed. This is understandable due to the sparsity on atoms reducing the degrees of freedom available for reconstruction. Figure (3) shows a sample of the sparse dictionary elements. While not displaying the level of interpretability wanted, they do show definite sparseness.

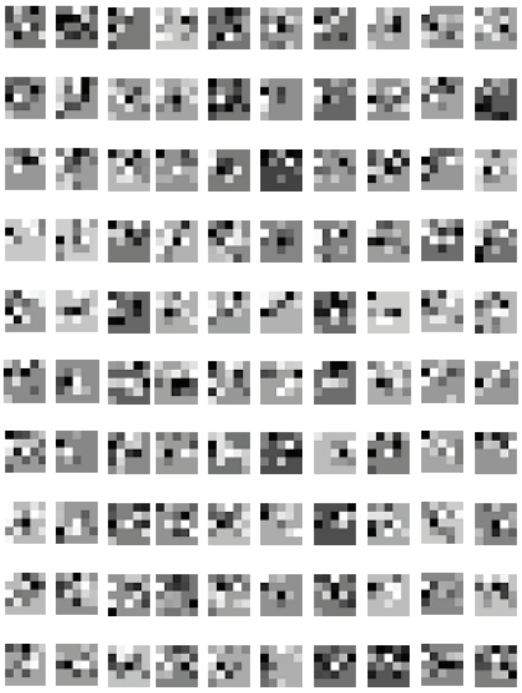

Fig. 3. Sparse Dictionary Atoms

REFERENCES

[1] A. M. BRUCKSTEIN, D. L. DONOHO, AND M. ELAD, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Review, 51 (2009), pp. 34–81.

[2] R. CHARTRAND AND W. YIN, *Iteratively reweighted algorithms for compressive sensing*, in Proceedings of the International Conference on Accoustics Speech and Signal Processing, 2008.

[3] S. FANG AND S. PUTHENPURA, *Linear Optimization and Extentions: Theory and Algorithms*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[4] I. GORODNITSKY AND B. RAO, *Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm*, Signal Processing, IEEE Transactions on, 45 (1997), pp. 600 –616.

[5] K. KREUTZ-DELGADO, J. MURRAY, B. RAO, K. ENGAN, T. LEE, AND T. SEJNOWSKI, *Dictionary Learning Algorithms for Sparse Representation*, Neural Computation, 15 (2003), pp. 349–396.

[6] D. LICHTENSTEIN AND M. SIPSER, *Go is polynomial space hard*, The Journal of the ACM, 27 (1980).
[7] M. LUSTIG, J. M. SANTOS, D. L. DONOHO, AND J. M. PAULY, *k-t sparse: High frame rate dynamic mri exploiting spatio-temporal sparsity*, in Proceedings of the 13th Annual meeting of the ISMRM, 2006.
[8] B. RAO AND K. KREUTZ-DELGADO, *An affine scaling methodology for best basis selection*, IEEE Transactions on Signal Processing, 47 (1999), pp. 187–200.
[9] D. SILVER, R. SUTTON, AND M. MULLER, *Reinforcement learning of local shape in the game of go*, in Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007, pp. 1053–1058.
[10] D. TAKHAR, J. N. LASKA, M. B. WAKIN, M. F. DUARTE, D. BARON, S. SARVOTHAM, K. F. KELLY, AND R. G. BARANIUK, *A new compressive imaging camera architecture using optical-domain compression*, vol. 6065, SPIE, 2006, p. 606509.
[11] E. C. D. VAN DER WERF, H. J. VAN DEN HERIK, AND J. W. H. M. UITERWIJK, *Learning to score final positions in the game of go*, Theoretical Computer Science, 349 (2005), pp. 168–183.
[12] T. VIRTANEN, *Separation of sound sources by convolutive sparse coding*, in in Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, 2004.
[13] D. WIPF AND M. NAGARAJAN, *Iterative reweighted l1 and l2 methods for finding sparse solutions*, Journal of Selected Topics in Signal Processing, 4 (2010). Special Issue on Compressive Sensing.
[14] D. WIPF AND B. RAO, *Sparse Bayesian Learning for Basis Selection*, IEEE Transactions on Signal Processing, 52 (2004), pp. 2153–2164.