Understanding and Exploiting the Connections between NMF and SVM

Vamsi K. Potluru
Dept. of Computer Science
University of New Mexico, Albuquerque, USA 87131
Email: ismav@cs.unm.edu

***Abstract*—Support Vector Machines (SVM) and Nonnegative Matrix Factorization (NMF) are standard tools for data analysis. We explore the connections between these two problems, thereby enabling us to import algorithms from SVM world to solve NMF and vice-versa. In particular, one such algorithm developed to solve SVM is adapted to solve NMF. Empirical results show that this new algorithm is competitive with the state-of-the-art NMF solvers.**

I. Introduction

Life is full of positive things. For instance, consider a person's height or salary. They are always positive or at the very least zero. When a quantity is zero or greater we say it is nonnegative. Quite a few real-world observations/data consist of positive or to be precise, nonnegative values. Examples include color intensities in images, chemical concentrations in experiments or radiation dosages in cancer treatments.

Why is this important? If algorithms do not take nonnegativity in to account, we could end up with negative "solutions" which do not have a physical interpretation. Of course, we could use a simple strategy of setting the negative components to zero but that could lead to a sub-optimal solution. Therefore, it is important to account for nonnegativity in the problem formulation.

An important problem formulation in classification is the Support Vector Machine (SVM). Similarly, Nonnegative Matrix Factorization (NMF) is a standard formulation for low rank approximations. Both formulations require nonnegativity on the recovered solutions. We study the connections between these two problems thereby enabling us to export algorithms from one problem domain to solve the other. In particular, we show how an SVM algorithm can be adapted to solve NMF.

II. Preliminaries and Previous Work

We give an introduction to NMF and the SVM problem formulations. Also, some of the algorithms for solving these are discussed.

A. Nonnegative Matrix Factorization

We present a brief introduction to NMF mechanics using notation that is standard in NMF literature. NMF is a tool to split a given nonnegative data matrix into a product of two nonnegative matrix factors [1]. The constraint of nonnegativity (all elements are ≥ 0) usually results in a parts-based representation and is different from other factorization techniques which result in more holistic representations (e.g. principal components analysis (PCA) and vector quantization (VQ)).

Given a nonnegative $m \times n$ matrix $\boldsymbol{X}$, we want to represent it with a product of two nonnegative matrices $\boldsymbol{W}, \boldsymbol{H}$ of sizes $m \times r$ and $r \times n$ respectively:

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}.$$

The NMF problem corresponding to Frobenius norm for the reconstruction error is given by:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 \quad \text{s.t.} \boldsymbol{W} \geq \boldsymbol{0}, \boldsymbol{H} \geq \boldsymbol{0} \tag{1}$$

In [1], simple multiplicative updates for $\boldsymbol{W}$ and $\boldsymbol{H}$ are presented which work well in practice and are as follows:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}\boldsymbol{H}^T}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T}, \tag{2}$$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{W}^T\boldsymbol{X}}{\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}}, \tag{3}$$

where the operator $\odot$ represents element-wise (Hadamard) multiplication. Division is also element-wise. It should be noted that the cost function to be minimized is convex in either $\boldsymbol{W}$ or $\boldsymbol{H}$ but not in both [1].

The slightly mysterious form for the above updates can be understood from the following description and is adapted from [1]. A simple additive update for $\boldsymbol{H}$ is given by:

$$\boldsymbol{H} = \boldsymbol{H} + \boldsymbol{\eta} \odot (\boldsymbol{W}^T\boldsymbol{X} - \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H})$$

If the learning rate given by the matrix elements of $\boldsymbol{\eta}$ are all set to some small positive number then this is the conventional gradient descent. However, setting the learning rate matrix as follows:

$$\boldsymbol{\eta} = \frac{\boldsymbol{H}}{\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}}$$

gives us the NMF updates. We note the multiplicative factors for the updates correspond to the negative component of the derivative divided element-wise by the positive component of the derivative respectively.

Other algorithms have been proposed to solve the NMF problem. Some of these are the projected gradient method [2] which we shall refer to as ProjGrad, a block pivoting method [3] called BlockPivot, a sequential constrained method [4] called FastHals and finally a greedy coordinate descent method [5] called GCD.

B. Support Vector Machines

Let the set of labeled examples be $\{(\boldsymbol{s}_i, y_i)\}_{i=1}^N$, with binary class labels $y_i = \pm 1$ corresponding to two classes, denoted by A and B respectively. Let the mapping $\Phi(\boldsymbol{s}_i)$ be the representation of the input datapoint $\boldsymbol{s}_i$ in space Φ, where we denote the space by the name of the mapping function performing the transformation. We now consider the problem of computing the maximum margin hyperplane for SVM in the case where the classes are linearly separable and the hyperplane passes through origin (we will relax this constraint presently.).

The dual quadratic optimization problem for SVM [6] is given by minimizing the following loss function:

$$S(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\boldsymbol{s}_i, \boldsymbol{s}_j) - \sum_{i=1}^{n} \alpha_i \qquad (4)$$
$$\text{subject to } \alpha_i \geq 0, i \in \{1..n\},$$

where $k(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is a kernel that computes the inner product $\Phi(\boldsymbol{s}_i)^T\Phi(\boldsymbol{s}_j)$ in the space Φ by performing all operations only in the original data space on $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, thus defining a Hilbert space Φ.

Recently, the cost of training of kernel SVM's has shifted the focus of the SVM community back to linear SVM for large scale applications. This has lead to the formulation of very efficient linear SVM solvers which converge to a ϵ precision solution in linear (in the number of training points) time [7], [8].

III. Connections between NMF and SVM

In this section, we will formalize some insights in to the similarities between the NMF and SVM problems. In particular, we will first show how to view SVM as a matrix factorization. Secondly, we will show how the subproblem in the alternate minimization scheme for NMF can be reduced to a single class SVM.

A. SVM as Matrix Factorization

Consider the Equation (4). The first sum can be split into three terms: two terms contain kernels of elements that belong to the same respective class (one term per class), and the third contains only the kernel between elements of the two classes. This rearrangement of terms allows us to drop class labels y_i, y_j from the objective function. Denoting $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with k_{ij} and defining $\rho_{ij} = \alpha_i \alpha_j k_{ij}$ for conciseness, we have:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\left(\sum_{ij\in A}\rho_{ij} - 2\sum_{\substack{i\in B\\ j\in A}}\rho_{ij} + \sum_{ij\in B}\rho_{ij}\right) - \sum_{i=1}^{n}\alpha_i \qquad (5)$$
$$\text{subject to } \alpha_i \geq 0, i \in \{1..n\}.$$

Noticing the square and the fact that $k_{ij} = \Phi(\boldsymbol{x}_i)^T\Phi(\boldsymbol{x}_j)$ we rewrite the problem as:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\Phi(\boldsymbol{X}_A)\boldsymbol{\alpha}_A - \Phi(\boldsymbol{X}_B)\boldsymbol{\alpha}_B\|_2^2 - \sum_{i\in\{A,B\}}\alpha_i \qquad (6)$$
$$\text{subject to } \alpha_i \geq 0,$$

where the matrices $\boldsymbol{X}_A, \boldsymbol{X}_B$ contain the datapoints corresponding to groups A and B respectively with the stacking being column-wise. The map Φ applied to a matrix corresponds to mapping each individual column vector of the matrix using Φ and stacking them to generate the new matrix. The vectors $\boldsymbol{\alpha}_A$ and $\boldsymbol{\alpha}_B$ contain the dual variables of the two groups A and B respectively. We will use the vector $\boldsymbol{\alpha}$ to denote the concatenation of vectors $\boldsymbol{\alpha}_A, \boldsymbol{\alpha}_B$. Expression (6) is a form of matrix factorization problem and resembles NMF with an additional term in the objective [1]. This connection was exploited to give multiplicative updates for solving SVM [9]. The above formulation enables other metrics $D(\Phi(\boldsymbol{X_A})\boldsymbol{\alpha}_A||\Phi(\boldsymbol{X}_B)\boldsymbol{\alpha}_B)$ than least squares for SVM such as more general Bregman divergence [10]. However, to be computationally efficient the metric used has to admit the use of the kernel trick.

B. NMF Reduced to Sequence of SVMs

We solve NMF by using the general framework of alternate minimization of the matrix factors. This leads to a sequence of convex sub-problems. We show that each of these can be reduced to solving a hard-margin single class SVM problem. First, we define the Nonnegative Least Squares (NNLS) problem which will aid us in this reduction. Let $\boldsymbol{W} \in R^{m\times n}$ be a matrix and $\boldsymbol{x} \in R^m$ a column vector. The nonnegative least squares problem (NNLS) is to find a column vector $\boldsymbol{h} \in R^n$ which solves the following problem:

$$\min_{\boldsymbol{h}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{W}\boldsymbol{h}\|_2^2$$
$$\text{s.t. } \boldsymbol{h} \geq \boldsymbol{0} \qquad (7)$$

If in addition, we have that the inputs $\boldsymbol{W}, \boldsymbol{x}$ are nonnegative, we get the Totally Nonnegative Least Squares (TNNLS) formulation.

Now, we are in a position to sketch the reduction. First, we show that TNNLS can be reduced to a single class SVM. Second, that each sub-problem of NMF can be reduced to a TNNLS.

Let $\boldsymbol{D}$ denote the diagonal matrix whose diagonal is given by the vector $\frac{1}{\boldsymbol{W}^T\boldsymbol{x}}$. Also, let $\boldsymbol{h} = \boldsymbol{D}\boldsymbol{z}$. Then,

$$\begin{aligned} G(\boldsymbol{z}) &= \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{W}\boldsymbol{D}\boldsymbol{z}\|_2^2 \\ &= \frac{1}{2}\boldsymbol{z}^T(\boldsymbol{W}\boldsymbol{D})^T(\boldsymbol{W}\boldsymbol{D})\boldsymbol{z} - \boldsymbol{x}^T\boldsymbol{W}\boldsymbol{D}\boldsymbol{z} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x} \\ &= \frac{1}{2}\boldsymbol{z}^T(\boldsymbol{W}\boldsymbol{D})^T(\boldsymbol{W}\boldsymbol{D})\boldsymbol{z} - \boldsymbol{1}^T\boldsymbol{z} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x} \end{aligned}$$

Ignoring the $\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}$, which does not change the location of the minimum, we see that it is an instance of the SVM objective in equation 4, with $\boldsymbol{\alpha}$ corresponding to $\boldsymbol{z}$ and $\boldsymbol{s}_i$ corresponding to $\boldsymbol{W}_i\boldsymbol{D}_{ii}$ with the kernel function being linear. We have a single class maximum margin classifier passing through origin where the datapoints given by $\{\boldsymbol{W}_i D_{ii}\}_i^n$ lie in the positive orthant. The reduction was used to efficiently solve the TNNLS problem leveraging fast SVM solvers [11].

The NMF problem can be written as a sequence of subproblems by using the framework of alternate optimization as

follows:

$$\min_{\boldsymbol{H}} \frac{1}{2}\|\text{vec}(\boldsymbol{X}) - (\boldsymbol{I} \otimes \boldsymbol{W})\text{vec}(\boldsymbol{H})\|_F^2$$
$$\text{s.t.} \boldsymbol{H} \geq \boldsymbol{0}$$
$$\min_{\boldsymbol{W}} \frac{1}{2}\|\text{vec}(\boldsymbol{X}^T) - (\boldsymbol{I} \otimes \boldsymbol{H})\text{vec}(\boldsymbol{W}^T)\|_F^2$$
$$\text{s.t.} \boldsymbol{W} \geq \boldsymbol{0} \quad (8)$$

Note that the TNNLS problems in (8) can be cast as single-class SVM problems and therefore NMF can be reduced to a sequence of SVM problems.

IV. Algorithm from SVM to NMF

A fixed-point algorithm was presented to solve SVM [12]. We adapt the algorithm to solve NMF by exploiting the connection between SVM and NMF as shown in Section III-B. We use the previously mentioned framework of alternate optimization for the matrix factors $\boldsymbol{W}$ and $\boldsymbol{H}$. As noted previously, it is sufficient to present the optimization scheme for one the factors while the other is fixed. Fix the matrix $\boldsymbol{W}$ in NMF (1) and solve the resulting convex sub-problem by the following updates for matrix $\boldsymbol{H}$:

$$\boldsymbol{H} \leftarrow \quad \boldsymbol{Q}^{-1}(\boldsymbol{W}^\top \boldsymbol{X} + (\boldsymbol{Q}\boldsymbol{H} - \boldsymbol{W}^\top \boldsymbol{X} - \mu \boldsymbol{H})_+) \quad (9)$$

where $\boldsymbol{Q} = \boldsymbol{W}^\top \boldsymbol{W}$ and $\mu = 1.9(\min(\text{eig}(\boldsymbol{Q})))$. Analogously, we have updates for optimizing the matrix $\boldsymbol{W}$ while the matrix $\boldsymbol{H}$ is fixed.

In practice, we found the first iteration of these updates to be slow. To ameliorate this, we initialize the matrix factors by using random matrices with entries in $[0, 1]$ and running multiplicative updates as given in (2) and (3) for the first few iterations and then switch to the above updates. Combining multiplicative updates initialization and the above updates [12], we obtain a novel approach for solving NMF which we denote by the name of its component algorithms as M&Ms.

The fixed-point operator corresponding to the update of matrix $\boldsymbol{H}$ has to be iterated to converge to the solution of the sub-problem. We use a stopping parameter ϵ to control the quality of solution of the sub-problem. We use the termination condition of requiring the Frobenius norm of the difference of the updated matrix $\boldsymbol{H}$ and its value at the previous iteration to be less than ϵ.

V. Experiments for NMF

In this section, we compare our M&Ms algorithm with the other state-of-the-art solvers for NMF. Running times are presented for all the algorithms when applied to two real-world datasets.

Experiments report scaled reconstruction error ($\|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F$) instead of objective value for convenience of display. All experiments were run on a 2.8 GHz machine with 8GB RAM running Linux. The number of cores was set to 1. Our M&Ms algorithm was implemented in MATLAB similar to the other algorithms.

A. Datasets

We consider the following two real-world datasets:

1) CBCL face dataset consists of 2429 images of size 19×19 and can be obtained at http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html.
2) ORL face dataset consists of 400 images of size 112×92 and can be obtained at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

B. Methodology

We applied our algorithm on the CBCL face dataset with rank set to 49. The running times of our algorithm and the others is shown in Figure 1. Similarly, we applied our algorithm to the ORL face dataset with rank set to 25. The running times are shown in Figure 1.

The matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ are initialized by multiplicative updates given by (2) and (3), and the stopping tolerance ϵ is initialized to 0.1. It is halved for every 10 subsequent iterations.

C. Discussion

Initial experiments show that our algorithm is competitive with the state-of-the-art algorithms. More extensive comparison needs to be done. However, the simplicity of our algorithm makes a good argument for taking a closer look at SVM algorithms and thereby develop efficient NMF algorithms.

VI. Proposed Future Work

Nonnegative Quadratic programming (NQP) involves optimizing a quadratic objective function subject to nonnegative constraints. It is defined as follows:

$$\min_x \frac{1}{2}\boldsymbol{x}^\top \mathbf{A}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

NQP encompasses a wide umbrella of important problems such as Least Absolute Shrinkage and Selection Operator (LASSO), Support Vector Machines (SVM), Nonnegative Least Squares(NNLS) and sub-problems of Nonnegative Matrix Factorization (NMF). Formulation for LASSO is:

$$\min_{\boldsymbol{h}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{W}\boldsymbol{h}\|_2^2 + \lambda\|\boldsymbol{h}|_1$$
$$\text{s.t. } \boldsymbol{h} \geq \boldsymbol{0}$$

An isomorphism was established between sparse separation and ϵ-SVM regression [13] and was used it kernelize sparse separation. Similarly, a connection between LASSO and SVM's was established and exploited for the kernel version of LASSO [14]. Furthermore, the kernel adatron (KA) algorithm for solving SVM [15] resurfaced as an NNLS algorithm [16].

Since the problems we have formulated so far are all NQP's it is conceivable that algorithms developed for one can be adapted to solve others. We would like to understand the connections betweens these various special cases of NQP and use the insight to develop faster algorithms among them.

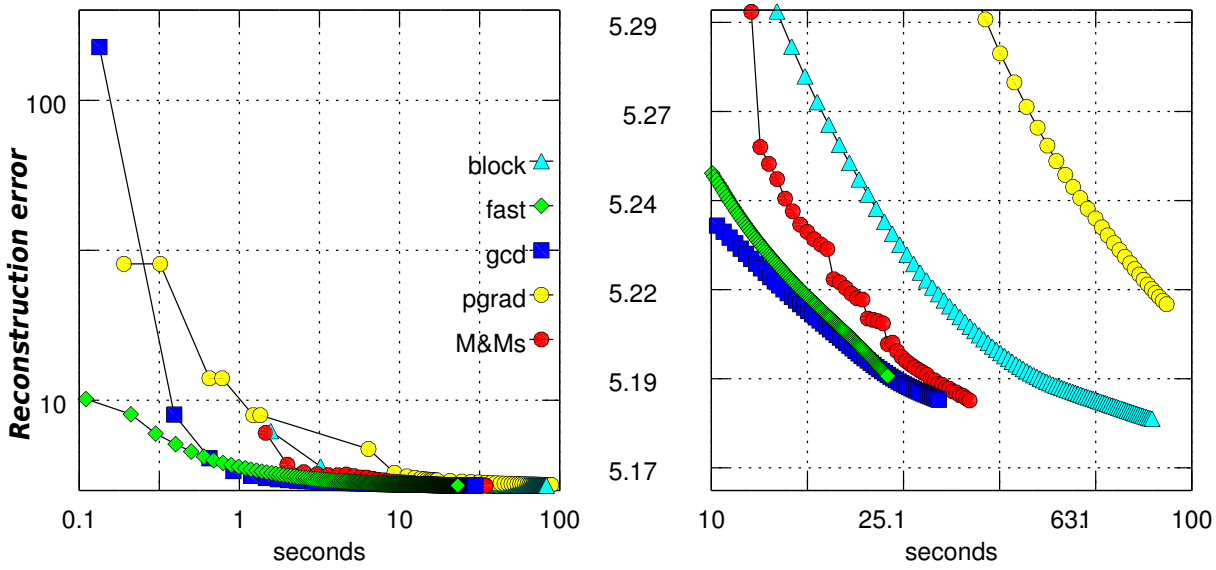

Fig. 1. Comparison between our algorithm M&Ms and several state of the art algorithms BlockPivot, FastHals, GCD, ProjGrad. Running time vs scaled reconstruction error for the CBCL face dataset (Left). Similarly for the ORL face dataset (Right).

ACKNOWLEDGEMENT

The author would like thank the following people for their valuable comments: Drs Vince Calhoun, Sergey Plis, Thomas Hayes, Morten Mørup, Terran Lane and Barak Pearlmutter. This work was supported with NIBIB grants 1 R01 EB 000840 and 1 R01 EB 005846.

REFERENCES

[1] D. D. Lee and S. H. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001, pp. 556–562. [Online]. Available: http://citeseer.ist.psu.edu/lee01algorithms.html

[2] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comp.*, vol. 19, no. 10, pp. 2756–2779, October 2007. [Online]. Available: http://neco.mitpress.org/cgi/content/abstract/19/10/2756

[3] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," *Data Mining, IEEE International Conference on*, vol. 0, pp. 353–362, 2008.

[4] A. Cichocki and A. Phan, "Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. 92, pp. 708–721, 2009.

[5] C. J. Hsieh and I. Dhillon, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," *ACM SIGKDD Internation Conference on Knowledge Discovery and Data Mining*, p. xx, 2011.

[6] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, 2001. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&path=ASIN/0262194759

[7] V. Franc and S. Sonnenburg, "Optimized cutting plane algorithm for support vector machines," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 320–327.

[8] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear svm," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 408–415. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390208

[9] V. Potluru, S. Plis, M. Mørup, V. Calhoun, and T. Lane, "Efficient multiplicative updates for support vector machines," in *Proceedings of the 2009 SIAM Conference on Data Mining (SDM)*, 2009.

[10] I. Dhillon and S. Sra, "Generalized nonnegative matrix approximations with Bregman divergences," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schlkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 283–290.

[11] V. K. Potluru, S. M. Plis, S. Luan, V. D. Calhoun, and T. P. Hayes, "Sparseness and a reduction from totally nonnegative least squares to svm," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 31 2011-aug. 5 2011, pp. 1922 –1929.

[12] O. L. Mangasarian and D. R. Musicant, "Lagrangian support vector machine classification," *Journal of Machine Learning Research*, no. 03, pp. 161–177, March 2001.

[13] S. Hochreiter and M. Mozer, "Monaural separation and classification of mixed signals: A support-vector regression perspective," in *3rd International Conference on Independent Component Analysis and Blind Signal Separation, San Diego, CA*. Citeseer, 2001.

[14] F. Li, Y. Yang, and E. Xing, "From lasso regression to feature vector machine," *Advances in Neural Information Processing Systems*, vol. 18, p. 779, 2006.

[15] T.-T. Frieß, N. Cristianini, and C. Campbell, "The Kernel-Adatron algorithm: a fast and simple learning procedure for support vector machines," in *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1998, pp. 188–196.

[16] V. Franc, V. Hlavac, and M. Navara, "Sequential coordinate-wise algorithm for the non-negative least squares problem," in *Computer Analysis of Images and Patterns*, 2005, p. 407. [Online]. Available: http://dx.doi.org/10.1007/11556121_50