NUMERIC METHODS FOR NONNEGATIVE MATRIX FACTORIZATION [*]

LARSON HOGSTROM[†]

Abstract. This report presents a framework and analysis to compare three common methods for computing nonnegative matrix factorization (NMF). Parts-based representations obtained through NMF have become a popular approach for dimensionality reduction in a variety of scientific and technical domain. This study focuses on the algorithms and numerical aspects of NMF computation. We review the derivation and continual descent properties of the multiplicative update rules which helped to first popularized the NMF approach. Alternating least squares and projection gradient methods are also presented as faster, more recent approaches. The accuracy and convergence of these algorithms is compared using simulated data and two of the methods were applied to a database of gene expression profiles measuring drug responses in the MCF7 cancer cell line. Lastly, the objective function of alternating least squares was directly modified to impose sparsity during matrix factorization. The fraction of near-zero entries was compared before and after modifying the objective function, as sparsity results are of known interest to many researchers using NMF.

Key words. dimensionality reduction, matrix factorization, least squares, nonnegativity, data representation

1. Introduction.

Non-negative data is ubiquitous in many domains. Chemical concentrations, word counts from written text, and pixels in digital images all take strictly non-negative values. Researchers working with these and related datatypes often seek to manipulate and interpret large matrices of input data. Low-rank approximations can aid in identifying the most important components of such data by ignoring nonessential components related to noise or known artifacts. Common goals in performing such low-rank approximations include clustering similar items into groups, retrieving related samples based on an user's input query, or isolating representative components of data that can be directly interpreted [1].

Nonnegative matrix factorization (NMF) is a dimensionality reduction technique that has been employed to address a variety of scientific and engineering challenges including text data mining [1, 2, 6] chemometrics [3, 4], and microarray analysis [8]. Like principal component analysis (PCA), NMF seeks to describe a large matrix of data using a small set of information dense components. Unlike PCA, which imposes orthogonality constraints on components, NMF requires nonnegativity after factorization. This procedure allows for only additive combinations of positive components to estimate the input matrix. The endpoint is a parts-based representation of the starting data where each component is more easily interpreted than PCA.

A number of algorithms have been proposed in order to construct NMF factorization with non-negative constraints. This approach was largely popularized by a 1999 article by Lee and Seung [6]. Their study, written for a general audience, examined the benefits of parts-based representations and presented applications of NMF to domains including text analysis and the decomposition of facial images. The report was followed by an additional study outlining more mathematical details of NMF as well as a simple multiplicative update algorithm [7]. The publications of Lee and Seung, however, were predated by lesser known work by Pentti Paatero. The method he developed and named 'positive matrix factorization' was based on a constrained alternating least squares algorithm and represents the first presentation of what is

[*] A final project for 18.335, Spring 2015

[†] (hogstrom@mit.edu).

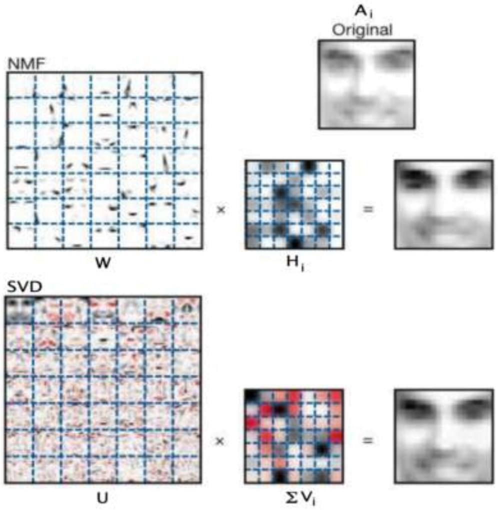

Fig. 1. *A comparison of NMF and SVD in creating low-rank approximations of facial data. The parts-based representation of NMF yields comments that can be directly interpreted as facial features such as noses, mouths and eyes. The interpretation of SVD components is more challenging. Reproduced from Lee and Seung, 1999.*

today known as NMF [3, 4].

For the purposes of this study, three algorithms were implemented and evaluated in the python language using the 'numpy' and 'scipy' packages for scientific computation. In simulations, the algorithms were compared by their accuracy after a fixed number of iterations. To illustrate a real-world application of NMF, nonnegative, low-rank approximations were used to describe gene expression profiles of known pharmacological groups. This proceeding report is organized in the following structure. Section 2 defines the NMF approach and reviews commonly used objective functions. Section 3 outlines the multiplicative update rule which is one of the oldest and most widely used procedures to perform matrix decomposition with NMF. Section 4 covers the alternating least squares approach which has variations to better characterized convergence properties. This is followed by a discussion of NMF sparsity and applications to gene expression data.

2. Problem Definition and Objective Functions.

Definition 2.1. *For a nonnegative matrix $A \in \Re^{mxn}$, select a low-rank approximation of size k such that there are two nonnegative matrices $W \in \Re^{mxk}$ and $H \in \Re^{kxn}$ which minimizes a function such as*

$$f(W, H) = \frac{1}{2}||A - WH||_F^2 \tag{2.1}$$

Other commonly used objective functions include Euclidian distance and Kullback-Leibler (KL) divergence. KL can be extended to a more general information-based framework using Renyi's divergence [9]. Here, a single parameter α is used to represent a continuum of distance measures and KL airises as a special case where $\alpha \to 1$.

$$KL(A||WH) = \sum_{ij} [A_{ij} \log \frac{A_{ij}}{(WH)_{ij}} - A_{ij} + (WH)_{ij}] \tag{2.2}$$

For its simplicity and widespread use, this report will focus on the objective function defined in (2.1).

Regardless of the objective used to obtain them, the resulting W and H matrices are often interpreted directly. The matrix W $\in \Re^{mxk}$ typically represents k different components of signal extracted from the original matrix. The H $\in \Re^{kxn}$ matrix can be interpreted as weighting factors defining the linear combination of k components used to reconstruct a sample in the original data matrix. In the application of NMF to gene expression data for example, each of the k columns of W represent a 'metagene' response. A small number of these metagene columns can be combined to estimate an original experimental column found in the original data matrix. The H matrix defines the weighting of each metagene.

3. Multiplicative Update Rule.

In their seminal paper, Lee and Seung began their analysis with a simple additive update rule for H that reduces the squared distance:

$$H_{aj} \leftarrow H_{aj} + \eta_{aj}[(W^T A)_{aj} - (W^T WH)_{aj}] \tag{3.1}$$

The authors noted that this update represents a typical gradient descent rule if each of the η_{au} terms are all equally set to a small positive number. Alternatively, the variables can be diagonally rescaled by setting

$$\eta_{aj} = \frac{H_{aj}}{(W^T WH)_{aj}} \tag{3.2}$$

and when placed into equation (3.1), this motivates the multiplicative update rules that seek to minimize the euclidian distance $||A - WH||$:

$$H_{aj} \leftarrow H_{aj} \frac{(W^T A)_{aj}}{(W^T WH)_{aj}} \qquad W_{ia} \leftarrow W_{ia} \frac{(W^T A)_{ia}}{(W^T WH)_{ia}} \tag{3.3}$$

Where the same logic for applies for both the update of H and W. Making use of auxiliary functions, Lee and Seung went on to demonstrate the non-increasing properties of NMF under the multiplicative update rules. It has been noted that they incorrectly claimed that the algorithm converges to a local minimum [10, 11, 22]. It is possible, therefore, that continual descent property of multiplicative updates could descend to a saddle point [13]. Furthermore, if $W_{ia} = 0$ after the kth iteration, then this entry remains $W_{ia} = 0$ for all subsequent iterations. This property is undesirable because the algorithm could theoretically remain fixed in a non-optimal descent path.

```
Data: Input data matrix: $\mathbf{A} \in \Re^{\mathbf{mxn}}$
Result: nonnegative factorization of A using k components, creating
        matrices $\mathbf{W} \in \Re^{mxk}$ and $\mathbf{H} \in \Re^{kxn}$
initialization;
$\mathbf{W} \leftarrow$ random dense (m x k) matrix
$\mathbf{H} \leftarrow$ random dense (k x n) matrix
for $i$ = 1 to maxiter do
    $\mathbf{H} = \mathbf{H}.*(\mathbf{W^T A})./(\mathbf{W^T WH})$
    $\mathbf{W} = \mathbf{W}.*(\mathbf{AH^T})./(\mathbf{WHH^T})$
end
```

Algorithm 1: Multiplicative update

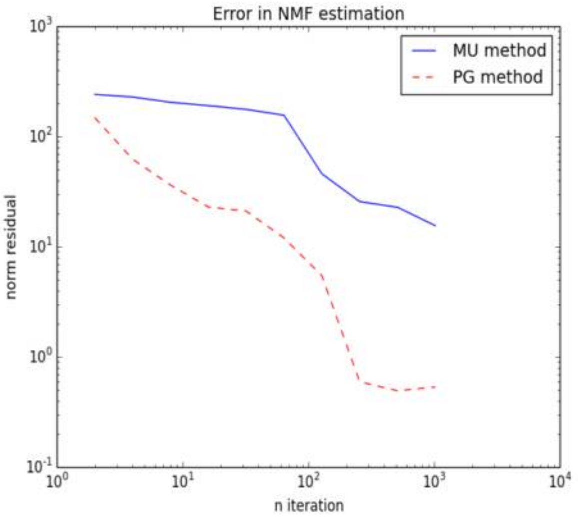

FIG. 2. *Error comparison of multiplicative update and projection gradient methods for performing NMF calculation. Both algorithms were applied to a matrix of fixed size and the number of iterations were systematically increased.*

The multiplicative update algorithm (Alg. 1) was compared to the NMF projection gradient method from [22] which is built into the python scientific computing package. The projection gradient method is a more complex and refined algorithm that approaches NMF as a minimization problem with bound constraints (see KKT conditions listed in equation 4.3). Figure 2 shows the normed error of NMF estimation after a fixed number of iterations with Alg.1 and the projection gradient method. The relative error obtained with multiplicative update increases with the size of the input matrix (see Fig. 3), though this is consistent with results from the projection gradient algorithm. It is important to note that the multiplicative approach will breakdown if a zero appears in the denominator. In practice, it is common to add a very small floating point value to the denominator during each iteration. The required work of O(mnk) per iteration is quite high compared to other exiting algorithms and multiplicative update tends to converge slowly for real world problems. Nevertheless, with its early establishment and its straightforward implementation this algorithm has maintained its popularity for NMF computation.

4. Alternating Least Squares. Alternating least squares can be viewed as a special case of block coordinate descent method. For this class of algorithms, the objective function is minimized with respect to a coordinate vectors x_i^k. Multidimensional vectors are minimized in cyclical order, with one block coordinate vector minimized during each iteration [13]. A single iteration takes the form:

$$x_i^{k+1} \in \text{argmin}_{\varepsilon \in X_i} = f(x_1^{k+1}, ..., x_{i-1}^{k+1}, \varepsilon, x_{i+1}^k, ..., x_n^k) \tag{4.1}$$

For the case of NMF, this scenario is especially simple as there are only two block variables, W and H. Sequentially one matrix is fixed and the other is improved.

$$\begin{aligned} W^{k+1} &= \text{argmin}_{W \geq 0} f(W, H^k) \\ H^{k+1} &= \text{argmin}_{H \geq 0} f(W^{k+1}, H) \end{aligned} \tag{4.2}$$

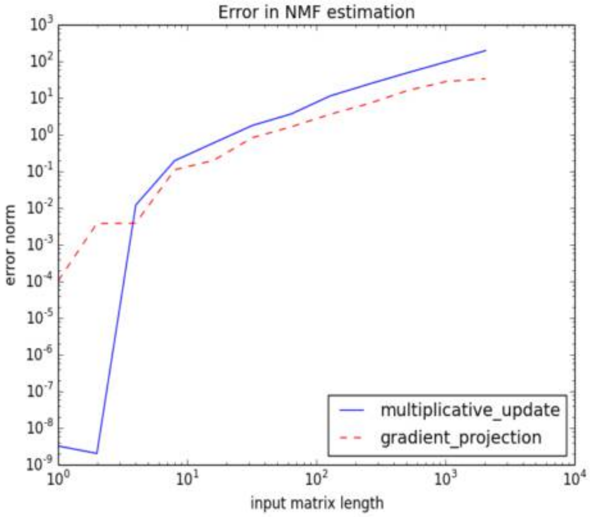

FIG. 3. *Estimation error results for multiplicative update and projection gradient methods increase at approximately the same rate with increasing matrix size.*

This takes advantage of the fact that while objective (2.1) is not convex in both W and H, it is convex in either W or H individually. One basic ALS algorithm that excites this cycle is outlined by Berry et al. [13]. This strategy imposes nonnegativity by setting all nonzero entries of W and H to zero during each iteration. Unlike multiplicative update rules, however these zero entries are not forced to remain at 0 for all proceeding iterations.

```
Data: Input data matrix: $\mathbf{A} \in \Re^{\mathbf{mxn}}$
Result: nonnegative factorization of $\mathbf{A}$ using k components, creating
        matrices $\mathbf{W} \in \Re^{mxk}$ and $\mathbf{H} \in \Re^{kxn}$
initialization;
$\mathbf{W} \leftarrow$ random dense (m x k) matrix
$\mathbf{H} \leftarrow$ random dense (k x n) matrix
for $i$ = 1 to maxiter do
    Solve for $\mathbf{H}$ in $\mathbf{W^TWH} = \mathbf{W^TA}$
    set negative elements in $\mathbf{H}$ to 0
    Solve for $\mathbf{W}$ in $\mathbf{HH^TW^T} = \mathbf{HA^T}$
    set negative elements in $\mathbf{W}$ to 0
end
```

Algorithm 2: Alternating least squares

In basic form of ALS (Alg. 2), each subproblem is solved in an unconstrained manner and the resulting W and H solutions are modified during each iteration to satisfy the desired non-negativty constraints. The simplicity of setting negative values in W and H to zero provides for an especially fast approach as demonstrated in the comparisons of this study. Theoretical evaluation of this algorithm's convergence properties, however, are difficult because each subproblem is formulated as an unconstrained least squares problem, but the solutions are directly modified and therefore do not map onto the original formulations. Other problem definitions allow for a framework to better understand the convergence properties of the two block coordinate descent method. Alternating non-negativity constrained least squares (ANLS), for

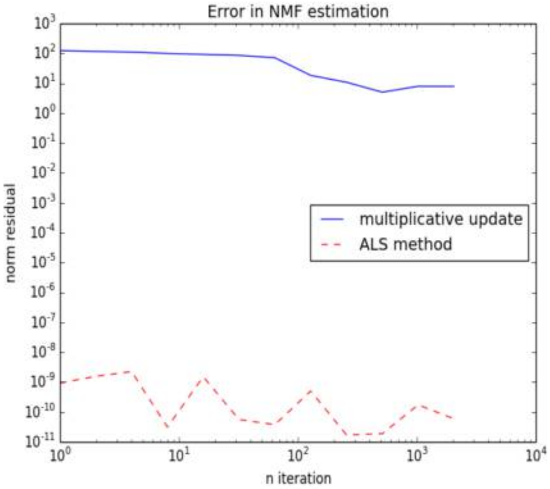

FIG. 4. *Estimation error resulting from the alternating least squares algorithm compared to multiplicative update.*

instance, provides a formulation that directly incorporates non-negative constraints in each block coordinate subproblem [19]. Adding the $W \geq 0$ and $H \geq 0$ constraints onto equation (4.2), the Karush-Kuh-Tucker (KK) optimality conditions can be used to define stationary points of the objective function (2.1) if and only if

$$\begin{aligned} W &\geq 0, \\ \bigtriangledown_W f(W,H) = WHH^T - AH^T &\geq 0, \\ W. * \bigtriangledown_W f(W,H) &= 0 \\ \\ H &\geq 0, \\ \bigtriangledown_H f(W,H) = W^T WH - W^T A &\geq 0, \\ H. * \bigtriangledown_H f(W,H) &= 0 \end{aligned} \tag{4.3}$$

For the block coordinate descent algorithms, it has been shown that the limit point of a sequence of sub-blocks is a stationary point if each subproblem has a unique solution [14]. Unfortunately, ANLS subproblems are not unique as there exists matrices in the form $\mathbf{X} \in \Re^{kxk}$ which represent scaling and permutation that satisfy $||\mathbf{A}\text{ - }\mathbf{WH}||_F = ||\mathbf{A}\text{ - }\mathbf{WXX}^{-1}\mathbf{H}||_F$ [19]. In the case of two bock problems, however, stationary points can be found as the limit point of the sequence of two sub-block solutions [15]. One classical approach to solving the two block subploblems is by way of an *active set algorithm* [16]. Here, the difference between a input matrix $Y \in \Re^{lxp}$ and its resultant factorization cab be split into vectors:

$$\min_{G \geq 0} ||BG - Y||_F^2 \rightarrow \min_{G \geq 0} ||B\mathbf{g}_1 - \mathbf{y}_1||_2^2, ..., \min_{G \geq 0} ||B\mathbf{g}_l - \mathbf{y}_m||_2^2 \tag{4.4}$$

where $B \in \Re^{lxm}$ and $y_i \in \Re^{lx1}$. Since there are m inequity constraints after the split, the ith constraint will be active if the ith regression coefficient is set to zero. Thus if the active set is known, active constraints can be treated as equality constraints instead of inequality constraints. The overall least square problem can be solved by examining only the variables in the passive set and calculating their unconstrained least squares solution [17]. The ANLS algorithm begins with an initial feasible set of regression coefficients. The zero vector of coefficients is often used as an initial feasible set as no constraints are violated. ANLS proceeds by iteratively removing variables from the active set until the true active set is found. The variables removed from the active set are used in an unconstrained linear regression to calculate the block coordinate descent solution. While the convergence properties of ANLS are better understood, this report omits this strategy in comparative analysis due to the complexity of the algorithm and its slow computation. Recent work has succeeded in improving ANLS speed by precomputing cross-product terms during the unconstrained linear regression calculations [17].

5. Sparsity in Alternating Least Squares. Sparse basis vectors often naturally arise during nonnegative matrix factorization. The are two potential benefits of sparsity in NMF representations. First, there is potential for saving storage and future computation relative to alternative low-rank approximations that occur with dense factorization methods such as SVD. Second and more importantly, sparsity may aid in interpreting the NMF results. It is often desirable to understand how input sample data can be constructed using a small number of parts-based representations, rather than being create from a linear combination of many diffuse components. The property of spareness, therefore, can aid in giving physical meaning to NMF results. A number of approaches have been created in order to manipulate sparsity in nonnegative matrix factorization [19,20,23]. To better control the fraction of zero or near zero entries, for example, Kim and Park propose directly modifying The objective function of the ALS method [20].

$$f(,H) = \frac{1}{2}[||A - WH||_F^2 + \lambda_W ||W||_F^2 + \lambda_H ||H||_F^2] \tag{5.1}$$

Here additional terms are used to penalize large positive values in the W and H factorization. These new penalties to objective function, therefore, encourage sparse solutions. This new objective function can be easily incorporated into the two block coordinate framework providing a new update rules for the W and H matrices:

$$\min_{W \geq 0} || \begin{pmatrix} H^T \\ \sqrt{\lambda_W} I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{kxm} \end{pmatrix} ||_F^2 \tag{5.2}$$

where I_k is the k x k identity matrix and 0_k is a zero matrix with the size k x m. Similarly the update rule for the H matrix can be modified to:

$$\min_{H \geq 0} || \begin{pmatrix} W \\ \sqrt{\lambda_H} I_k \end{pmatrix} H - \begin{pmatrix} A \\ 0_{kxm} \end{pmatrix} ||_F^2 \tag{5.3}$$

These new update rules called alternating constrained least squares (ACLS) were compared against the original alternating least squares approach outlined in section 4. The analysis presented in Fig. 1 shows the fraction of non-zero entries in the resulting W matrix for ALS and ACLS factorization. Similar results can be obtained by manipulating the λ_H parameter for the H factorization matrix. These results show a powerful approach for researchers to control NMF sparsity according to their research question

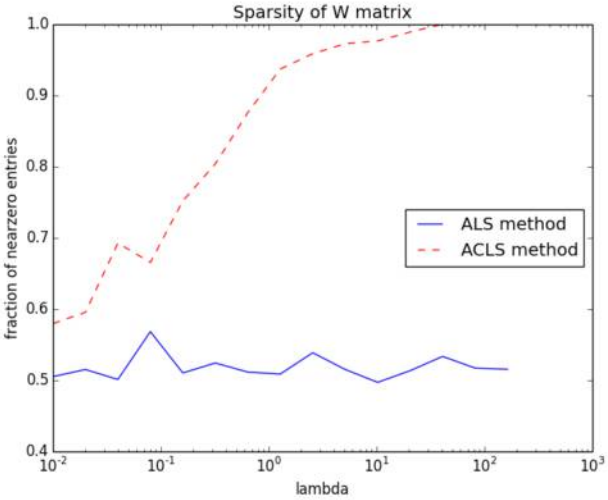

Fig. 5. *NMF sparsity manipulation using the ACLS method. Increasing the value of λ allows users to increase the fraction of near-zero entries in W and H.*

6. Applications to gene expression analysis. A classical example of NMF is the factorization of a large database of human face images. Factorization using PCA will yield a series of 'eigenfaces' which can be added or subtracted, but individual components are not easy to interpret on their own. The NMF parts-based representation however, yields components like noses, eyes, ears, and mouths which can be added linearly to construct a face. Parts-based representations have been helpful in identifying and interpreting patterns in a number of biological contexts. For example, NMF has shown to be less sensitive to *a priori* gene selection or initial conditions when identifying context-dependent patterns of gene expression [8].

To examine the ability of NMF to identify unique parts-based representations in gene expression profiles, experimental data was obtained from the Connectivity Map Projects (lincscloud.org). This database represents the world?s largest systematic collection of gene expression profiles performed in cancer cell lines. A small fraction of drug profiles were obtained from database. specifically, those drugs which were tested MCF7 cells and fit into one of four pharmacological classes; Topoisomerase II inhibitors, RAR agonists, PKC activators, and JNK inhibitors. These represent important chemical tools for research and clinical practice. The advantage examining this selection of drugs is that chemicals grouped in the same pharmacological classes are expected to have similar gene expression responses. The NMF was tested for its ability to recover these known relationships.

Both ALS and multiplicative update approaches were applied to this data. The number of components was selected *a priori* at k=20. Fig. () shows H weighting factors for the 20 NMF components obtained with multiplicative update rule for each of the four pharmacological classes. Similar patterns were also found for ALS (though with different component ordering.) The similarity of both these methods suggests researchers might opt to spend their time examining other numerical factors beside convergence that may influence biological interpretation of NMF. The effects of sparsity were not examined with this particular dataset, but could have a significantly influence on the interpretation of the NMF results. Other factors, including smoothness, and the uniqueness of factorization could also impact the biological findings.

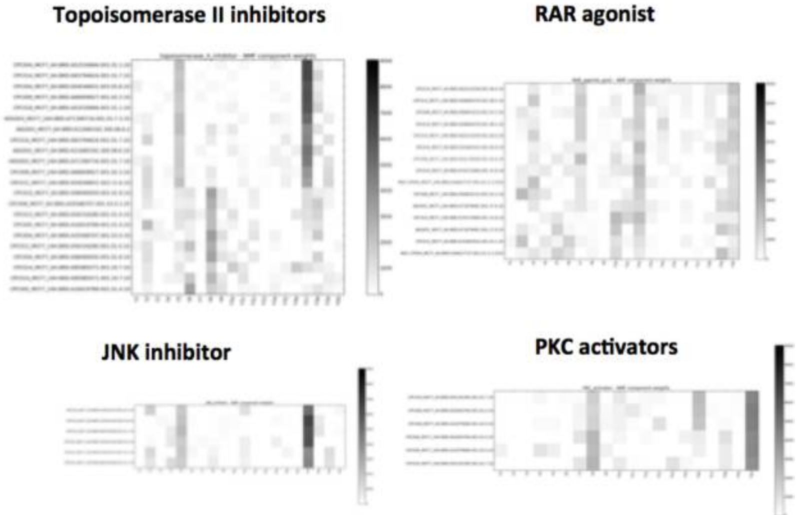

Fig. 6. *Nonnegative matrix factorization used to identify parts-based representations in gene expression profiles. The weighting factors from the H matrix are displayed for drugs from four pharmacological class tested in MCF7 cells. Drug profiles from the same pharmacological class can be constructed using similar NMF components.*

New numerical approaches that give better control on components constraints could improve the utility of NMF in biomedical research.

8. Conclusion. Despite the simplicity of defining nonnegative matrix factorization, the landscape of numeric methods used to address the problem remains complex. Recent efforts have sought to create new NMF algorithms with well defined convergence properties. Methods such as alternating non-negativity constrained least squares (ANLS) have succeeded in improving our analytical understanding of NMF convergence, but remain slow and cumbersome when compared to simpler methods that appear to converge in most real-world scenarios. When evaluating NMF tools, researchers should be guided by the parameters of their data and the research question of interest. For especially large datasets the speed and simplicity ALS might serve as a reasonable starting point for NMF calculation. Researchers with modest datasets, might prefer ANLS or the multiplicative update approach. Finally, the results presented in this report also show that researchers can manipulate the sparsity of factorization by adopting specialized objective functions.

REFERENCES

[1] A. Langville, C. Meyer, R. Albright, *Initializations for the nonnegative matrix factorization*, Preprint (2006)

[2] V Pauca et al, *Text mining using non-negative matrix factorizations*, Proceedings SIAM International Conference on Data Mining (2004) SDM'04.

[3] P. Paatero, *Positive matrix factorization; a non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics (1994) 5, 111-126.

[4] P. Paatero, *Least squares formulation of robust non-negative factor analysis*, Chemometrics and Intell. Laboratory Syst. (1997) 37, 23-35.

[5] P. Paatero, *The multilinear engine - a table-driven least squares program for solving multilinear problems, including the n-way parallel factor analysis model*, J. Comput. Graphical Statist. (1999) 8, 1-35.

[6] D. Lee, H. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature (1999) 401, 788-791.

[7] D. Lee, H. Seung, *Algorithms for Nonnegative Matrix Factorization*, Advances in Neural Information Processing Systems (2001) 13, 556-562.

[8] J. Brunet et al. , *Metagenes and molecular pattern discovery using matrix factorization*, Proceedings of the Nat. Acad. Sci (2004) 101, 4164-4169.

[9] K. Devarajan, K. Ebrahimi , *Molecular pattern discovery using non-negative matrix factorization based on Renyi's information measure*, Preprint (2005)

[10] M. Chu, D. Plemmons, S. Ragni, *Optimality, computation, and interpretation of nonnegative matrix factorizations*, SIAM Journal on Matrix Analysis (2004)

[11] E. Gonzalez, Y. Zhang, *Accelerating the lee-swung algorithm for nonnegative matrix factorization*, Technical Report TR-05-02, Rice University (2005)

[12] C. Lin, *Projected gradient methods for non-negative matrix factorization*, Information and Support Services Technical Report (2005)

[13] M. Berry, M. Browne, A. Langville, P. Pauca, R. Plemenos, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics and Data Analysis 52 (2007) pp. 155 - 173.

[14] D. Bertsekas, *Nonlinear Programing*, Athena Scientific, Belmont, MA (1999)

[15] L. Grippo et al., *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett (2000) 26(3), p. 127-136.

[16] C. Lawson, R. Hanson, *Solving Least Squares Problems*, SIAM, Philadelphia, PA (1995)

[17] R. Bro, S. Jong, *A fast non-negativity constrained linear least squares algorithm*, J. Chemometrics (1997) 11), p. 393-401.

[18] A. Cichocki, R. Zdunek, A. Huy Phan, S. Amari, *Nonnegative Matrix and Tensor Factorizations*, Wiley, Natick, MA, 2009.

[19] H. Kim, H. Park, *Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method*, SIAM Journal on Matrix Analysis and Applications (2008) 30(2), 713?730.

[20] H. Kim, H. Park, *Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis*, Bioinformatics (2007) 23, 1495-1502.

[21] R. Albright, J. Cox, et al., *Algorithms, initializations, and convergence for the nonnegative matrix factorization*, Preprint (2014)

[22] C. Lin, *Projected Gradient Methods for Nonnegative Matrix Factorization*, Neural Computation (2007) 19, 2757-2779.

[23] S. Li, *Learning spatially localized parts-based representations*, Pro. IEEE Conference on Computer Vis. and Patt. Rec. (2001) pp. 207-212.