

Non-Negative Matrix Factorization with Exogenous Inputs for Modeling Financial Data

Steven Squires, Luis Montesdeoca, Adam Prügel-Bennett, and Mahesan Niranjana

Electronics and Computer Science, University of Southampton, UK
{ses2g14, ljm1e14, apb, mn}@ecs.soton.ac.uk

Abstract. Non-negative matrix factorization (NMF) is an effective dimensionality reduction technique that extracts useful latent spaces from positive value data matrices. Constraining the factors to be positive values, and via additional regularizations, sparse representations, sometimes interpretable as part-based representations have been derived in a wide range of applications. Here we propose a model suitable for the analysis of multi-variate financial time series data in which the variation in data is explained by latent subspace factors and contributions from a set of observed macro-economic variables. The macro-economic variables being external inputs, the model is termed XNMF (eXogenous inputs NMF). We derive a multiplicative update algorithm to learn the factorization, empirically demonstrate that it converges to useful solutions on real data and prove that it is theoretically guaranteed to monotonically reduce the objective function. On share prices from the FTSE 100 index time series, we show that the proposed model is effective in clustering stocks in similar trading sectors together via the latent representations learned.

Keywords: Non-negative matrix factorisation · Computational finance · Dimensionality reduction

1 Introduction

Many modern problems in machine learning are posed in high dimensions, due to the ease with which we can now acquire and archive data. However, useful information we wish to extract about a problem domain might be expected to be characterised by fewer features. Hence dimensionality reduction is a useful tool in identifying the latent subspaces of interest. By working in a smaller subspace we hope to reduce the noise, compress the data and, potentially, enable classification or regression machines to generalise better. Non-negative matrix factorization (NMF) is an increasingly popular choice of linear dimensionality reduction in large part because it is often capable of producing a sparse and parts based representation of the data.

In standard NMF we consider an input matrix with m dimensions and n samples: $\mathbf{V} \in \mathbb{R}^{m \times n}$. The aim is to find a lower-dimensional representation of the data by factorizing \mathbf{V} into two matrices \mathbf{W} and \mathbf{H} such that $\mathbf{V} \approx \mathbf{WH}$,

where $\mathbf{W} \in \mathbb{R}^{m \times r}$ and $\mathbf{H} \in \mathbb{R}^{r \times n}$. Generally $r \ll m$ and $r \ll n$ so that NMF creates a new representation of the data in a significantly reduced subspace.

Financial systems are inherently complex, driven by the objectives of market players, along with monetary and fiscal policies of governments. Pure time series analysis has been applied extensively to asset returns [1–3], exchange rates [4] and derivatives [5, 6]. NMF has been applied to financial data in several ways, such as identifying underlying trends in stock market data [7]. Also sparse-semi-NMF approaches to portfolio diversification have been used to minimise risk [8, 9]. The appeal of NMF in this context is that returns on assets, expressed as ratios of their market prices, are positive. Factorizing multivariate asset return data into low rank factors can potentially discover low dimensional representations that are determined by sectors of assets that are likely to show similar responses. However, statistical signal analysis methods usually do not take into account exogenous information from macro-economic variables (referred to in this paper as macro-variables) that have significant contributions to market movements.

In this paper, we propose a matrix factorization method that includes known exogenous variables as additional components of subspace modelling. We expect such factorizations to potentially uncover sector-specific drivers from among a wide range of macro-variables available. Specifically, our model represents the variation in any asset as consisting of contributions from sector-specific components and selected macro-variables. Hence the main novel contributions in this paper are the specification of such a factorization model and a learning algorithm for it. We empirically demonstrate the effective performance of our approach on share price data from FTSE 100 companies and theoretically prove that the XNMF algorithm is guaranteed to monotonically reduce the objective function.

This paper is structured as follows: in Section 2 we present our model including the underlying mathematics and the proof of monotonic reduction of the objective function; in Section 3 we discuss the real and synthetic data we used; in Section 4 we display our results; and in Section 5 we conclude and summarize our results.

2 Model and Learning Algorithm

Our aim is to find a combined representation of the share price data using the share price itself with the addition of external macro-variables. We can utilise standard NMF methods to find representations such that $\mathbf{V} \approx \mathbf{W}_1 \mathbf{H}_1$ and, separately, $\mathbf{V} \approx \mathbf{W}_2 \mathbf{H}_2$ where $\mathbf{W}_1 \in \mathbb{R}^{m \times r_1}$, $\mathbf{H}_1 \in \mathbb{R}^{r_1 \times n}$ and $\mathbf{H}_2 \in \mathbb{R}^{r_2 \times n}$ are all matrices to be found. The macro-variables are recorded in $\mathbf{W}_2 \in \mathbb{R}^{m \times r_2}$ and are fixed quantities. Here m represents the number of time points, r_1 is a parameter to select, r_2 is the number of macro-variables and n is the number of stocks.

There are many approaches to perform NMF, a simple method is to utilise the multiplicative update technique of Lee and Seung [10] which gives updates for \mathbf{W} and \mathbf{H} of

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{[\mathbf{V}\mathbf{H}^T]}{[\mathbf{W}\mathbf{H}\mathbf{H}^T]}, \quad \mathbf{H} \leftarrow \mathbf{H} \odot \frac{[\mathbf{W}^T\mathbf{V}]}{[\mathbf{W}^T\mathbf{W}\mathbf{H}]} \quad (1)$$

where \odot is the Hadamard product and $\frac{\square}{\square}$ denotes element-wise division. These updates push the matrices towards a minimum of the objective function $\|\mathbf{V} - \mathbf{W}\mathbf{H}\|_{\text{Fro}}^2$. In our combined representation we want to find matrices \mathbf{W}_1 , \mathbf{H}_1 and \mathbf{H}_2 that satisfy $\mathbf{V} \approx \mathbf{W}_1\mathbf{H}_1 + \mathbf{W}_2\mathbf{H}_2$ which requires us to minimise

$$f = \frac{1}{2} \|\mathbf{V} - \mathbf{W}_1\mathbf{H}_1 - \mathbf{W}_2\mathbf{H}_2\|_{\text{Fro}}^2. \quad (2)$$

As minimising Equation (2) with respect to \mathbf{W}_1 , \mathbf{H}_1 and \mathbf{H}_2 together is non-convex we hold two of the matrices constant whilst updating the third using multiplicative updates. Each individual problem is then convex, although the overall problem remains non-convex and there is no guarantee of reaching an optimal solution. Multiplicative updates are a type of scaled gradient descent therefore we need to find $\nabla_{\mathbf{W}_1} f$, $\nabla_{\mathbf{H}_1} f$ and $\nabla_{\mathbf{H}_2} f$. First we multiply out Equation (2) and get:

$$\begin{aligned} f &= \frac{1}{2} \text{tr} [(\mathbf{V} - \mathbf{W}_1\mathbf{H}_1 - \mathbf{W}_2\mathbf{H}_2)^T(\mathbf{V} - \mathbf{W}_1\mathbf{H}_1 - \mathbf{W}_2\mathbf{H}_2)] \\ &= \frac{1}{2} \text{tr} [\mathbf{V}^T\mathbf{V} - \mathbf{V}^T\mathbf{W}_1\mathbf{H}_1 - \mathbf{V}^T\mathbf{W}_2\mathbf{H}_2 - \\ &\quad \mathbf{H}_1^T\mathbf{W}_1^T\mathbf{V} + \mathbf{H}_1^T\mathbf{W}_1^T\mathbf{W}_1\mathbf{H}_1 + \mathbf{H}_1^T\mathbf{W}_1^T\mathbf{W}_2\mathbf{H}_2 - \\ &\quad \mathbf{H}_2^T\mathbf{W}_2^T\mathbf{V} + \mathbf{H}_2^T\mathbf{W}_2^T\mathbf{W}_1\mathbf{H}_1 + \mathbf{H}_2^T\mathbf{W}_2^T\mathbf{W}_2\mathbf{H}_2]. \end{aligned} \quad (3)$$

We then differentiate Equation 3 with respect to \mathbf{W}_1 , \mathbf{H}_1 and \mathbf{H}_2 respectively to give three equations:

$$\nabla_{\mathbf{W}_1} f = (\mathbf{W}_1\mathbf{H}_1\mathbf{H}_1^T + \mathbf{W}_2\mathbf{H}_2\mathbf{H}_1^T - \mathbf{V}\mathbf{H}_1^T), \quad (4)$$

$$\nabla_{\mathbf{H}_1} f = (\mathbf{W}_1^T\mathbf{W}_1\mathbf{H}_1 + \mathbf{W}_1^T\mathbf{W}_2\mathbf{H}_2 - \mathbf{W}_1^T\mathbf{V}) \quad (5)$$

and

$$\nabla_{\mathbf{H}_2} f = (\mathbf{W}_2^T\mathbf{W}_2\mathbf{H}_2 + \mathbf{W}_2^T\mathbf{W}_1\mathbf{H}_1 - \mathbf{W}_2^T\mathbf{V}). \quad (6)$$

We apply multiplicative updates to \mathbf{W}_1 , \mathbf{H}_1 and \mathbf{H}_2 by:

$$\mathbf{W}_1 \leftarrow \mathbf{W}_1 \odot \frac{[\mathbf{V}\mathbf{H}_1^T]}{[\mathbf{W}_1\mathbf{H}_1\mathbf{H}_1^T + \mathbf{W}_2\mathbf{H}_2\mathbf{H}_1^T]}, \quad (7)$$

$$\mathbf{H}_1 \leftarrow \mathbf{H}_1 \odot \frac{[\mathbf{W}_1^T\mathbf{V}]}{[\mathbf{W}_1^T\mathbf{W}_1\mathbf{H}_1 + \mathbf{W}_1^T\mathbf{W}_2\mathbf{H}_2]} \quad (8)$$

and

$$\mathbf{H}_2 \leftarrow \mathbf{H}_2 \odot \frac{[\mathbf{W}_2^T \mathbf{V}]}{[\mathbf{W}_2^T \mathbf{W}_2 \mathbf{H}_2 + \mathbf{W}_2^T \mathbf{W}_1 \mathbf{H}_1]} \quad (9)$$

where \odot is the Hadamard product and $\frac{\square}{\square}$ indicates element-wise division. We will discuss how changes to \mathbf{W}_1 reduces the objective function noting that the same argument also applies to changes in \mathbf{H}_1 and \mathbf{H}_2 . As we want to follow the gradient down towards a minimum, if $\nabla_{\mathbf{W}_1} f < 0$ then we want to increase \mathbf{W}_1 . This is equivalent to $\mathbf{V} \mathbf{H}_1^T > \mathbf{W}_1 \mathbf{H}_1 \mathbf{H}_1^T + \mathbf{W}_2 \mathbf{H}_2 \mathbf{H}_1^T$, and, as shown in Equation (7), \mathbf{W}_1 is increased. Conversely if $\nabla_{\mathbf{W}_1} f > 0$ then we need \mathbf{W}_1 to decrease, which the multiplicative update does because $\mathbf{W}_1 \mathbf{H}_1 \mathbf{H}_1^T + \mathbf{W}_2 \mathbf{H}_2 \mathbf{H}_1^T > \mathbf{V} \mathbf{H}_1^T$. The final eventuality, that $\nabla_{\mathbf{W}_1} f = 0$, implies we have found a minimum of \mathbf{W}_1 and so want to keep \mathbf{W}_1 the same. Our multiplicative update multiplies \mathbf{W}_1 by one, fulfilling our requirement. We should note that if $\nabla_{\mathbf{W}_1} f = 0$ we are not necessarily at a minimum of the objective function as the other two matrices may still change which might change the situation of \mathbf{W}_1 such that $\nabla_{\mathbf{W}_1} f$ is no longer zero.

While this argument shows that the updates move in the correct direction, that is no guarantee of a monotonic reduction of the objective function as we could overshoot the minimum. However, part of the value of multiplicative updates is that Lee and Seung proved that they do produce a monotonic reduction [11].

We prove that our algorithm monotonically reduces Equation (2) by extending the proof of Lee and Seung [11] to cover the XNMF objective function using the same notation they did. Definition 1 and lemma 1 from their paper remain the same but we change the $K(h^t)$ diagonal matrix of lemma 2 to

$$K_{a,b}(h_{(1)}^t) = \delta_{a,b} (\mathbf{W}_1^T \mathbf{W}_1 \mathbf{h}_1^t + \mathbf{W}_1^T \mathbf{W}_2 \mathbf{h}_2) / h_{(1)a}^t \quad (10)$$

which changes only the $K(h_{(1)}^t)$ term of $G(\mathbf{h}_1, \mathbf{h}_1^t)$. We then prove that $G(\mathbf{h}_1, \mathbf{h}_1^t)$ is an auxiliary function of the altered $F(\mathbf{h}_1)$:

$$F(\mathbf{h}_1) = \frac{1}{2} (\mathbf{v} - \mathbf{W}_1 \mathbf{h}_1 - \mathbf{W}_2 \mathbf{h}_2)^T (\mathbf{v} - \mathbf{W}_1 \mathbf{h}_1 - \mathbf{W}_2 \mathbf{h}_2) \quad (11)$$

which requires the proof that $M_{a,b}(\mathbf{h}_1^t) = h_{(1)a}^t (K(\mathbf{h}_1^t) - \mathbf{W}_1^T \mathbf{W}_1)_{a,b} h_{(1)b}^t$ is positive semidefinite:

$$\begin{aligned} & \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\nu} \\ &= \sum_{a,b} \nu_a M_{a,b} \nu_b \\ &= \sum_{a,b} \left[h_{(1)a}^t \left((\mathbf{W}_1^T \mathbf{W}_1 \mathbf{h}_1^t + \mathbf{W}_1 \mathbf{W}_2 \mathbf{h}_2) / h_{(1)a}^t \right)_{a,b} h_{(1)b}^t \nu_a^2 - \nu_a h_{(1)a}^t (\mathbf{W}_1^T \mathbf{W}_1)_{a,b} h_{(1)b}^t \nu_b \right] \end{aligned}$$

$$\begin{aligned}
 &= \sum_{a,b} \left[(\mathbf{W}_1^T \mathbf{W}_1)_{a,b} h_{(1)a}^t h_{(1)b}^t \nu_a^2 - \nu_a h_{(1)a}^t (\mathbf{W}_1^T \mathbf{W}_1)_{a,b} h_{(1)b}^t \nu_b + (\mathbf{W}_1^T \mathbf{W}_2)_{a,b} h_{(2)b}^t h_{(1)a}^t \nu_a^2 \right] \\
 &= \sum_{a,b} \left[(\mathbf{W}_1^T \mathbf{W}_1)_{a,b} h_{(1)a}^t h_{(1)b}^t \left(\frac{1}{2} \nu_a^2 + \frac{1}{2} \nu_b^2 - \nu_a \nu_b \right) + (\mathbf{W}_1^T \mathbf{W}_2)_{a,b} h_{(2)b}^t h_{(1)a}^t \nu_a^2 \right] \\
 &= \sum_{a,b} \left[(\mathbf{W}_1^T \mathbf{W}_1)_{a,b} h_{(1)a}^t h_{(1)b}^t (\nu_a - \nu_b)^2 + (\mathbf{W}_1^T \mathbf{W}_2)_{a,b} h_{(2)b}^t h_{(1)a}^t \nu_a^2 \right] \\
 &\geq 0. \tag{12}
 \end{aligned}$$

Our proof is then the same as Lee and Seung except, due to the different $K(h^t)$, we end with:

$$h_{(1)a}^{t+1} = h_{(1)a}^t \frac{(\mathbf{W}_1^T \mathbf{v})_a}{(\mathbf{W}_1^T \mathbf{W}_1 \mathbf{h}_1 + \mathbf{W}_1^T \mathbf{W}_2 \mathbf{h}_2)_a}. \tag{13}$$

which proves that our algorithm will monotonically reduce the objective function for \mathbf{H}_1 . Equivalent proofs are trivially shown for \mathbf{W}_1 and \mathbf{H}_2 .

3 Data

We demonstrate the effectiveness of our model and learning algorithm empirically using daily data from FTSE 100 companies taken over a twenty year period. To deal with non-stationarity that may exist over such a long period in time, we also split the data into four equal sections in time and show results on all four separately.

Table 1. Macro-variables used in this study

Macro-variable	Frequency	Macro-variable	Frequency
Gross Domestic Product	Quarterly	Unemployment	Monthly
Interest Rate	Monthly	Inflation Index Rate	Monthly
Imports Goods&Services	Quarterly	Exports	Monthly
Oil Imports	Monthly	Gross National Income	Quarterly
M1 Money Supply	Monthly	Productivity	Quarterly
GBP/USD	Daily	Contribution to CPI	Monthly
Balance of Payments	Monthly	Oil Investment	Daily
Government Gross Reserve	Monthly		

In Table 1 we show the macro-variables used in this study. The choice of which macro-variables to use is somewhat arbitrary, there are many potential macro-variables, and they can be changed. To compensate for the differences in frequency between the share data (recorded on work days) and the macro-variable data we have linearly interpolated between all the macro-variable data so that the dimensionality (the number of time points) are equal.

4 Results

We first confirm empirically that our algorithm achieves the desired goal, the reduction in the error until it reaches a minimum. In Figure 1(a) we show how the error changes with iteration for different values of r for the three different algorithms. We will use the same terminology throughout: NMF results are from the algorithm which minimised $\|\mathbf{V} - \mathbf{W}_1\mathbf{H}_1\|_{\text{Fro}}^2$, XNMF (exogenous inputs NMF) is for the minimisation of $\|\mathbf{V} - \mathbf{W}_1\mathbf{H}_1 - \mathbf{W}_2\mathbf{H}_2\|_{\text{Fro}}^2$ and EX (exogenous inputs alone) is for the minimisation of $\|\mathbf{V} - \mathbf{W}_2\mathbf{H}_2\|_{\text{Fro}}^2$. The blue dashed lines are for different values of r for NMF and the solid black lines for different values of r_1 for XNMF.

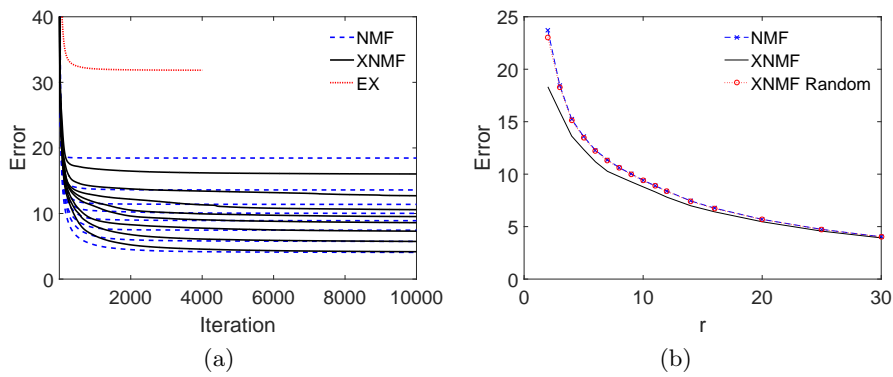


Fig. 1. (a) The extended multiplicative update algorithm reduces the error monotonically with iteration until a plateau is reached. The multiple blue dashed (NMF) and solid black (XNMF) lines are for different sizes of the subspace, r . Generally the XNMF algorithm requires more iterations to approach a minimum than the NMF algorithm, but reaches a lower final error. (b) The final errors for different sizes of the subspace, r , for NMF (blue dashed lines with crosses), XNMF (solid black line) and XNMF using a \mathbf{W}_2 with random values (red dotted with circles). At all values of r that were implemented XNMF produces smaller errors than NMF or the randomised XNMF. As r is increased the difference between the errors produced by the algorithms reduces as the capacity of the NMF model increases and begins to overfit the data.

The EX algorithm (red dotted line) produces a poor approximation as it contains no information from the actual stocks themselves. The results of particular note are those of the XNMF algorithm which works as we expect it to, we see a fall in the objective function with iteration until it approaches a minimum where the error plateaus. The XNMF algorithm takes more iterations than the NMF algorithm to approach a minimum which might be expected as we have three matrices to optimise rather than two. In addition, the third matrix may make the objective function more non-convex than with just two matrices to optimise.

In Figure 1(b) we show the final errors from performing normal NMF (blue dashed line with crosses) and XNMF (solid black line) for different sizes of the subspace, r . At low values of r the model does not have enough subspace dimensions (columns of \mathbf{W}_1) to effectively fit the data and so the errors are high. The additional macro-variables here make a significant difference to the quality of the fit. As r increases the benefit of the additional information decreases as the increased capacity of the $\mathbf{W}_1\mathbf{H}_1$ part of the model means that a good fit to the data is possible without any additional information. As r increases it is likely that the model is overfitting the data, so any use of NMF requires a sensible choice of r to be made [12]. We also include a version of XNMF (red dotted line with circles) called XNMF Random where the \mathbf{W}_2 matrix is composed of random numbers. The NMF and XNMF Random plots are hard to distinguish demonstrating that the XNMF method is extracting real information from the external data, and not just reducing the error by increasing the size of the parameter space.

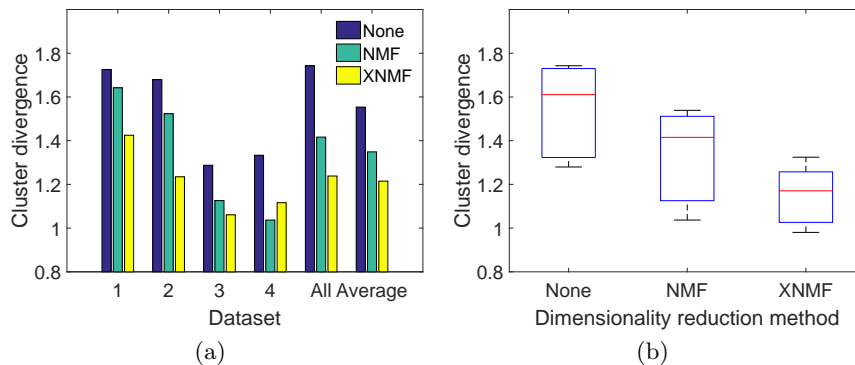


Fig. 2. a) A representation of how much clusters diverge with time. K-means clustering was applied to non-dimensionality reduced data (dark blue bars), dimensional reduction using NMF (light blue bars) and dimensional reduction using XNMF (yellow bars) for four times periods and for a combination of the four periods. The clusters produced from data with no dimensional reduction diverge the most, with application of NMF the divergence is reduced and with XNMF we see the smallest divergence, the clusters tend to hold together better through time. b) Boxplots of the same results demonstrating the improvement of XNMF over NMF.

A particular appeal of NMF is noise suppression, by reducing the noise we might expect to be able to extract more real features from the data. A key result demonstrated with gene expression data is that the reduction in noise achieved by matrix factorization leads to stable clustering and biologically relevant inference about genes [13, 14]. In financial data we are often interested in how stocks and shares move together through time, a balanced portfolio would not contain lots

of shares which are likely to fall in the same period. If we can effectively cluster the shares we can then build a more resilient portfolio.

We can cluster stock data into groups using a range of techniques including the popular K-means clustering. We are then interested in the quality of the clustering in the future, clusters that hold together better would be desirable. While NMF is not a clustering technique we can use the dimensionality reduction to create a new sub-space in which we apply clustering.

We performed K-means clustering on three versions of the data: a) no dimensionality reduction; b) dimensionality reduced using NMF; c) dimensionality reduced using XNMF. A measure of the similarity of a cluster is the average distance to the cluster centre using the non-dimensionality reduced data. We are interested in the change in the average distance to the cluster centre as this gives us a measure of how similar the cluster is at different time points. In general, we would expect an increase in distance as clusters will tend to diverge with time. If we see a smaller increase using the dimensionality reduced versions, it shows that the NMF techniques are allowing us to produce clusters which generalise better.

In Figure 2 we see the results of this forward prediction of clustering. First the data was split in half into a “training” set, the first half of the data in time, and a “testing” set, the second half of the data. The training data was then clustering into seven cluster centres using, respectively: the raw data, \mathbf{V} ; \mathbf{H}_1 from NMF; and \mathbf{H}_1 from XNMF. We chose the size of the subspace, r , using a combination of domain knowledge about numbers of sectors in the data, and automatic techniques to assess subspace size [12]. The y-axis shows the ratio of the average distances from each data-point to its cluster centre between the testing data and the training data. A smaller value means the cluster stayed closer together. We see a clear trend, the raw data performs the worst whilst XNMF gives the best performance, and NMF gives a result in between the other two.

5 Conclusion

In this paper we introduce a matrix factorization model suitable for multi-variate financial time series that includes known exogenous macro-variables. We use real FTSE 100 stock data to show that the multiplicative update factorization algorithm of XNMF produces lower errors than standard NMF and that stock clusters formed with the addition of exogenous data stay tighter bound through time. We also prove theoretically that the algorithm is guaranteed to monotonically reduce the objective function.

References

1. Weigend, A.S., Huberman, B.A., Rumelhart, D.E.: Predicting the future: A connectionist approach. *International journal of neural systems* 1(03), 193–209 (1990)

2. Tamiz, M., Hasham, R., Jones, D., Hesni, B., Fargher, E.: A two staged goal programming model for portfolio selection. In: *Multi-Objective Programming and Goal Programming*, pp. 286–299. Springer (1996)
3. Omran, M.F.: Nonlinear dependence and conditional heteroscedasticity in stock returns: Uk evidence. *Applied Economics Letters* 4(10), 647–650 (1997)
4. Babu, A., Reddy, S.: Exchange rate forecasting using arima, neural network and fuzzy neuron. *Journal of Stock & Forex Trading* 3(4), 1–5 (2015)
5. Niranjana, M.: Sequential tracking in pricing financial options using model based and neural network approaches. In: *Advances in neural information processing systems*. pp. 960–966 (1997)
6. Montesdeoca, L., Niranjana, M.: Extending the feature set of a data-driven artificial neural network model of pricing financial options. In: *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*. pp. 1–6. IEEE (2016)
7. de Fr in, R., Drakakis, K., Rickard, S., Cichocki, A.: Analysis of financial data using non-negative matrix factorization. In: *International Mathematical Forum*. vol. 3, pp. 1853–1870. Journals of Hikari Ltd (2008)
8. de Fr in, R., Drakakis, K., Rickard, S.: Portfolio diversification using subspace factorizations. In: *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*. pp. 1075–1080. IEEE (2008)
9. Wang, J.: Stock trend extraction via matrix factorization. In: *International Conference on Advanced Data Mining and Applications*. pp. 516–526. Springer (2012)
10. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755), 788–791 (1999)
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: *Advances in neural information processing systems*. pp. 556–562 (2001)
12. Squires, S., Pr ugel-Bennett, A., Niranjana, M.: Rank selection in nonnegative matrix factorization using minimum description length. *Neural Computation* (2017)
13. Brunet, J.P., Tamayo, P., Golub, T.R., Mesirov, J.P.: Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences* 101(12), 4164–4169 (2004)
14. Devarajan, K.: Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS Comput Biol* 4(7), e1000029 (2008)