A Neurodynamics-Based Nonnegative Matrix Factorization Approach Based on Discrete-Time Projection Neural Network

Nian Zhang and Keenan Leatham

University of the District of Columbia
Department of Electrical and Computer Engineering
Washington, D.C. 20008, USA
{keenan.leatham, nzhang}@udc.edu

Abstract. This paper contributes to study the influence of various NMF algorithms on the classification accuracy of each classifier as well as to compare the classifiers among themselves. We focus on a fast nonnegative matrix factorization (NMF) algorithm based on discrete-time projection neural network (DTPNN). The NMF algorithm is combined with three classifiers in order to find out the influence of dimensionality reduction performed by the NMF algorithm on the accuracy rate of the classifiers. The convergent objective function values in terms of two popular objective functions, Frobenius norm and Kullback-Leibler (K-L) divergence for different NMF based algorithms on a wide range of data sets are demonstrated. The CPU running time in terms of these objective functions on different combination of NMF algorithms and data sets are also shown. Moreover, the convergent behaviors of different NMF methods are illustrated. In order to test its effectiveness on classification accuracy, a performance study of three well-known classifiers is carried out and the influence of the NMF algorithm on the accuracy is evaluated. Furthermore, the confusion matrix module has been incorporated into the algorithms to provide additional classification accuracy comparison.

Keywords: Nonnegative Matrix Factorization, Discrete-time Projection Neural Network, Dimensional Reduction, Feature Selection, Classification.

1 Introduction

Modern technologies have produced an explosion of massive data. In 2020 an estimated 40 trillion gigabytes of data will be generated, imitated, and consumed (Gantz et al. 2012). The rapid growth of complex and heterogeneous data has posed great challenges to data processing and management. Established data processing technologies are becoming inadequate given the growth of data. Advanced machine learning technologies are urgently needed to overcome big data challenges. They can help to ascertain valued insights for enhanced decision-making process in critical sectors such as healthcare, economy, smart energy systems, and natural catastrophe prediction, etc.

One of the biggest challenges that traditional classification methods face is that when the dimensionality of data is high but with few data, a large number of class prototypes existing in a dynamically growing dataset will lead to inaccurate classification results. Therefore, selection of effective dimensionality reduction techniques is of great importance. Feature selection is one of the powerful dimensionality reduction techniques that selects an optimal subset based on various statistical tests for correlation with the outcome variable without losing the best predictive accuracy. Although numerous combinations of feature selection algorithms and classification algorithm have been demonstrated, we explore an emerging and increasingly popular technique in analyzing multivariate data - non-negative matrix factorization (NMF) technique, and combine it with three state-of-the-art classifier, namely Gaussian process regression, Support Vector Machine, and Enhanced K-Nearest Neighbor (ENN), in order to investigate the influence of NMF on the classification accuracy.

NMF is one of the low-rank approximate techniques and is popular for dimensionality reduction. However, dimensionality reduction techniques incorporate non-negative constraints and, thus, obtains part-based representation (Xiao et al. 2014). Nevertheless, since it was first introduced, NMF and its varied forms were primarily studied in image retrieval and classification (Che and Wang 2018; Wang et al. 2017; Li et al. 2017). The effectiveness of NMF for classifying numerical features other than images is still under investigation. In this paper, we explore this aspect to find out if NMF can significantly improve the classification accuracy. Moreover, there is lack of study on the performance of a combined NMF with classifiers to our best knowledge. Thus, we extend research concerning integrate NMF with different classifiers with the goal to determine appropriate ones. A discrete-time projection neural network will be used develop the NMF algorithm due to the power of global convergence and fast convergence rate (Xu et al. 2018).

As a global optimization approach, neurodynamic optimization approach was proposed for robust pole assignment via both state and output feedback control systems by minimizing the spectral condition number (Le et al. 2014). A novel neurodynamic optimization approach for the synthesis of linear state feedback control systems via robust pole assignment based on four alternative robust measures was proposed (Le and Wang 2014). A two-time-scale neurodynamic approach to constrained minimax optimization using two coupled neural networks was presented (Le and Wang 2017). Neurodynamic systems for constrained biconvex optimization consists of two recurrent neural networks (RNNs) operating collaboratively at two timescales. By operating on two timescales RNNs can avoid instability and optimize initial states (Gorski et al. 2007).

Because of the superior computing capability of the neurodynamic optimization approach, this paper will present a neurodynamics-based NMF algorithm based on a discrete time projection neural network. The rest of the paper is organized as follows. In Section 2, non-negative matrix factorization (NMF) and different classifiers are discussed. In Section 3, continuous-time projection neural network and discrete-time projection neural network are introduced. In Section 4, the NMF algorithm based on the discrete-time projection neural network (DTPNN) are described. In Section 5, the comparison of convergent objective function values and CPU running time on different

NMF based algorithms in terms of the two objective functions are presented. The comparison of different classifiers is also demonstrated. Finally, the paper is concluded in Section 5.

2 Related Works

2.1 Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF), is an emerging algorithm where a matrix V is factorized into two matrices, W and H, with all three matrices containing no negative elements in them, as shown in Fig. 1. Part of the reason is because the non-negativity will make the new matrices easier to investigate (Gong et al. 2018). Let matrix V be the product of the matrices W and H,

$$V = W \times H$$

By computing the column vectors of V as linear combinations of the column vectors in W using coefficients supplied by columns of H, each column of V can be computed as follows:

$$v_i = W \times h_i$$

where v_i is the i-th column vector of the product matrix V and h_i is the i-th column vector of the matrix H.

The most attractive advantage by adopting NMF is dimensional reduction. When factorizing matrices, the dimensions of the factor matrices will be significantly lower than the original matrix. For example, if V is an $m \times n$ matrix, W is an $m \times p$ matrix, and H is a $p \times n$ matrix, then p can be significantly smaller than both m and n.

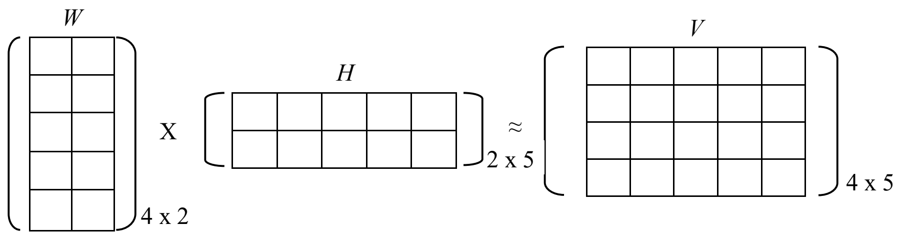

Fig. 1. Representation of non-negative matrix factorization. The matrix V is factorized into two reduced matrices, W and H. When multiplied, they approximately reconstruct V.

2.2 Gaussian Process Regression (GPR)

One of the most well-known nonparametric kernel-based probabilistic models with infinite-dimensional generalization of multivariate normal distributions is Gaussian process regression (GPR) models. Gaussian processes have wide applications in statistical modeling, regression to multiple target values, and analyzing mapping in higher dimensions. There are four varied models with different kernels. The rational quadratic GPR kernel allows us to model data varying at multiple scales. Square exponential GPR is a function space expression of a radial basis function regression model with infinitely many basis functions. A fascinating feature is that inner products are replaced by the basis functions with kernels. The advantage to this feature is handling large data sets in higher dimensions will unlikely produce huge errors. Also, it handles discontinuities well. The matern 5/2 kernel takes spectral densities of the stationary kernel and create Fourier transforms of RBF kernel. Exponential GPR is identical to the Squared Exponential GPR except that the Euclidean distance is not squared. Exponential GPR replaces inner products of basis functions with kernels slower than the Squared Exponential GPR. It handles smooth functions well with minimal errors, but functions with discontinuities does not handle well. A comprehensive comparison of classification performance among them is shown in terms of various model statistics. The classification error rates of these four models are also compared to the extended nearest neighbor (ENN), classic k-nearest Neighbor (KNN), naive Bayes, linear discriminant analysis (LDA), and the classic multilayer perceptron (MLP) neural network (Zhang et al. 2018).

2.3 Support Vector Machine (SVM)

Support vector machine (SVM) analysis is identified as one of the most popular supervised learning models for classification and regression. SVM regression is well-known for its nonparametric capability and has various kernel models. Linear SVM is a linearly scalable routine meaning that it creates an SVM model in a CPU time. If data are not linearly separable, Quadratic SVM is adopted to decide an interval between two classes. It is implemented by mapping the original feature space to a higher dimensional feature space where the training data is separable. The Gaussian kernel depends on the Euclidean distance between two points and is based on the assumption that similar points are close to each other in terms of Euclidean distance. The comparison of their performance on the photo-thermal infrared imaging spectroscopy classification is demonstrated in (Zhang and Leatham 2017).

2.4 Enhanced K-Nearest Neighbor (ENN)

Unlike the conventional k-nearest neighbor (KNN) method, the enhanced KNN method is devised to find out the k nearest neighbors of each sample in the training dataset, as well as the unknown test object (Tang and He 2017). A concept of validity rating is used to measure how similar a pre-determined group of samples resemble their k nearest neighbors (Zhang et al. 2017). Finally, a classifier will assign the unknown test object to a class membership based on the validity ratings.

2.5 Frobenius Norm

Frobenius norm, sometimes called the Hilbert-Schmidt norm is one of the oldest and simplest matrix norms (Chellabonia et al. 2003). Frobenius norm of a matrix is established when only if the matrix A is a rank-one matrix or a zero matrix. The Frobenius norm, sometimes also called the Euclidean norm (a term used for the vector L^2-norm), is matrix norm of an mxn matrix A defined as the square root of the sum of the absolute squares of its elements,

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}\right|^2}$$

2.6 Biconvex Optimization

Biconvex Optimization is where the objective function and constraint set can be biconvex. Biconvex optimization frequently occurs in numerous scientific and engineering applications such as spectrum sensing in cognitive radio networks, sparse 3-D reconstruction of dynamic objects, wireless energy transfers of communication systems, classification, visual recognition, robust stability analysis of control systems, and among other applications. Several algorithms are available for biconvex optimization. For example, alternate convex search (ACS) is presented to optimize x and y in alternately until attaining a partial optimum. The block coordinate descent (BCD) method is proposed for multiconvex optimization. Also, biconvex optimization is a parallel solution for neurodynamic optimization in the development of field-programmable gate arrays (FPGAs).

3 Background

3.1 Continuous-Time Projection Neural network

We formulate an optimization problem as follows:

$$\text{Min } f(x) \quad \text{s.t. } l \le x \le h \tag{1}$$

This problem can be solved by the following one-layer continuous-time projection neural network solution [17].

$$\epsilon \frac{dx}{dt} = -x + g(x - \nabla f(x)) \tag{2}$$

Where $\epsilon > 0$ is a time constant, $\nabla f(x)$ denotes the gradient of f, and $g(\cdot)$ is a piecewise linear activation function.

$$g(\xi_i) = \begin{cases} l_i, & \xi_i < l_i \\ \xi_i, & l_i \le \xi_i \le h_i \\ h_i, & \xi_i > h_i \end{cases}$$

To customize to the NMF algorithms, l_i will be 0 and h_i will be ∞. Accordingly, $g(\cdot)$ has become a rectified linear unit (ReLU) activation function.

$$g(\xi_i) = \begin{cases} 0, & \xi_i < 0 \\ \xi_i, & \xi_i \geq 0 \end{cases} \tag{3}$$

3.2 Discrete-Time Projection Neural Network

Considering the needs for global convergence and fast convergence rate, a discrete-time projection neural network has been used to develop the NMF algorithm. By applying Euler discretization to the continuous-time projection neural network in (2), it will be transformed into a discrete-time projection neural network (DTPNN).

$$x_{k+1} = x_k + \lambda_k[-x_k + g(x_k - \nabla f(x_k))] \tag{4}$$

where λ_k is a step size.

4 Non-Negative Matrix Factorization Method Based on DTPNN

4.1 Dynamic Equation of Discrete-Time Projection Neural Network (DTPNN)

The dynamic equations of DTPNN for two factorization matrices are formulated based on (4):

$$w_{k+1} = w_k + \lambda_k[-w_k + g(w_k - \nabla f(w_k))]$$
$$h_{k+1} = h_k + \lambda_k[-h_k + g(h_k - \nabla f(h_k))] \tag{5}$$

where λ_k is a step size.

The selection of step size λ_k is extremely important. The stability of the DTPNN will be unstable if λ_k equals or exceeds a certain bound (Xia and Wang 2000). The procedure of the selection of step size λ_k can be found in Section 4.2.

4.2 Backtracking Line Search

In order to minimize $f(x_k + \lambda_k p_k)$ in (5), we use the following procedure to find the step size λ_k.

Algorithm 1. Backtracking Line Search Algorithm

Given $\lambda_{init} > 0,\ i.e.\ \lambda_{init} = 1, \alpha \in \left(0, \frac{1}{2}\right), \beta \in (0,1),\ i.e.\ \beta = 1/2$

Set $\lambda_0 = \lambda_{init}$

Repeat $\lambda_{k+1} = \beta\lambda_k$ **Until**

$$f(x_k + \lambda_k p_k) \leq f(x_k) + \alpha\lambda_k \nabla f(x_k)^T p_k \tag{6}$$

4.3 Neurodynamics-Based Non-Negative Matrix Factorization Algorithm

A non-negative matrix factorization algorithm named PN^3MF based on biconvex optimization formulation is developed in (Che and Wang 2018).

Algorithm 2. The PN^3MF algorithm

Initialization

Set k = 0, $\alpha, \beta, w_0, h_0, \lambda_k^w, \lambda_k^h$, error tolerance ϵ and maximum iteration K.

while $k < K$ **and** $|f(w_{k+1}, h_{k+1}) - f(w_k, h_k)| > \epsilon$ **do**

  while (6) is not satisfied **do**

    $\lambda_k^w = \lambda_k^w \cdot \beta$

    $\lambda_{k+1}^w = \lambda_k^w$

    $$w_{k+1} = w_k + \lambda_{k+1}^w[-w_k + g_w(w_k - \nabla_w f(w_k, h_k))] \quad (7)$$

  end while

  while (6) is not satisfied **do**

    $\lambda_k^h = \lambda_k^h \cdot \beta$

    $\lambda_{k+1}^h = \lambda_k^h$

    $$h_{k+1} = h_k + \lambda_{k+1}^h[-h_k + g_w(h_k - \nabla_h f(w_{k+1}, h_k))] \quad (8)$$

  end while

  k = k + 1

end while

return w_k, h_k

4.4 Combined NMF and Classification Algorithm

In this paper, we combine the NMF algorithm with different classifier to explore the efficiency of the PN^3MF algorithm.

Algorithm 3. Combined PN^3MF-Classification Algorithm

Input: V: training set

  r: cluster numbers

  S: p unknown samples without labels

Output: c: predicted class labels of the p unknown samples

Training Procedure:

1. Normalize the training set
2. Solve the NMF optimization problem:

$$[W, H] = PN^3MF(V, r)$$

Test Procedure:

1. Normalize the test set
2. Solve the NMF optimization problem:

$$\min f(W,H) = \frac{1}{2}\|V - WH\|_F^2$$

3. Predict the class label, c_i
4. Return *c*

4.5 On the Complexity of PN³MF

As an additional estimation to our work, in this sub-section, we use analysis of algorithms to determine the time complexity of PN³MF. In particular, we use Big O notation as an indicator of the efficiency and scalability of our approach to big data.

First, we analyze time complexity for matrix multiplication using Big O. For NMF, the running time depends on the size of the matrices. That is *m*x*p* and *p*x*n*; hence, we can say the complexity is $\boldsymbol{O}(mnp)$. If we assume that **V** is quadratic, meaning that *m* is equal to *n*, and we consider the worst case value of *p*, i.e., when *p is also equal to n*, the complexity can be simplified to $\boldsymbol{O}(n^3)$.

Second, we analyze complexity for a one-layer neural network. Since, in a dense or fully-connected layer, each neuron is connected to the previous layer and the activation function is computed for each neuron, the forward propagation running time of PN^2 depends on the on the size of the matrices (*m*x*p* and *p*x*n*). And since the learning procedure need multiple calculations of Gradient descent, the backpropagation running time of PN^2 depends not only on the size of the matrices (*m*x*p* and *p*x*n*) but also on the average number of Gradient's checks *t* needed to converge. Hence, the complexity of the test or prediction procedure is $\boldsymbol{O}(mnp)$ and the complexity of the training or learning procedure is $\boldsymbol{O}(mnpt)$. Assuming that **V** is quadratic, *p* is equal to *n* and that *t* is equal to *n,* we obtain a forward propagation complexity of $\boldsymbol{O}(n^3)$ and a backpropagation complexity of $\boldsymbol{O}(n^4)$. We can further simplify the total polynomial time complexity of PN³MF to $\boldsymbol{O}(n^3 + n^4)$.

Finally, three classifiers are used in our experiments. The time complexity of GPR, SVM, and ENN depends on the cardinality of the training set and the dimensionality of each sample; well-known implementations of these classifiers result in a cubic complexity $\boldsymbol{O}(n^3)$. The table below presents the comparison on the complexity of the algorithms evaluated.

Table 1. Time Complexity Comparison

Algorithm	Time Complexity using Big O Notation
MUR	$\boldsymbol{O}(nmp + mp^2 + np^2)$
ALS	$\boldsymbol{O}(mp^2 + mnp) + \boldsymbol{O}(np^2 + mnp)$
PG	$\boldsymbol{O}(nmp) + k \text{ x } \boldsymbol{O}(tmp^2 + tnp^2)$
AS	$\boldsymbol{O}(nmp + mp^2 + np^2) + k \text{ x } \boldsymbol{O}(mp^2 + np^2)$
BBP	$\boldsymbol{O}(nmp + mp^2 + np^2) + k \text{ x } \boldsymbol{O}(mp^2 + np^2 + p^3 + n\log_2 m + m\log_2 n)$
NeNMF	$\boldsymbol{O}(nmp + mp^2 + np^2) + k \text{ x } \boldsymbol{O}(mp^2 + np^2)$
PN³MF	$\boldsymbol{O}(nmp) + \boldsymbol{O}(tmnp)$

5 Experimental Results

In this section, we intend to study various NMF algorithms on the classification accuracy of each classifier as well as to compare the classifiers among themselves. NMF algorithms are used to decompose original data set V according to the cluster number r. MUR (Lee and Seung 2001), ALS (Berry et al. 2007), PG (Lin et al. 2007), AS (Kim et al. 2007), BBP (Kim and Park 2007), NeNMF (Guan et al. 2007), and the proposed PN^{3}MF algorithms are compared. Three classifiers are applied to both the original and reduced dimensionality. Nine commonly used real-world datasets from UCI Machine Learning Repository are chosen to conduct the experiments (Lichman 2013).

5.1 Initialization

In the experiments, the error tolerance is set to be 10^{-7} and the maximum iterations is initialized to 5,000. Let $\alpha \in \left(0, \frac{1}{2}\right)$ and $\beta \in (0,1)$. The initial value of λ_{init} for f_1 (Frobenius-norm) and f_2 (Kullback-Leibler divergence) is set to 2.0 and 1.0.

5.2 Convergent Objective Function Values

Two objective functions, Frobenius-norm and Kullback-Leibler (K-L) divergence are adopted to evaluate the optimization performance of factorization. Table 2 shows convergent values of Frobenius-norm function. Compared with six NMF algorithms, most of the time PN^{3}MF reaches the lowest objective function value.

Similarly, Table 4 records convergent values of the Kullback-Leibler (K-L) divergence function. PN^{3}MF gets the best results on most data sets.

5.3 CPU Running Time

Table 3 presents CPU running time of these algorithms when Frobenius-norm function is used. Although MUR and ALS algorithms consume less time on the breast tissue data set, they fails to achieve the minimum objective function value.

Table 5 provided the CPU running time of those algorithms when Kullback-Leibler (K-L) divergence function is used, and show that PN^{3}MF always consumes less CPU running time than other NMF algorithms.

Table 2. Convergent objective function values of Frobenius-norm function

	Vertebral	Breast Cancer	Haberman	Breast Tissue	Movement Libras	ILPD	Ionosphere	Vowel	Segmentation
MUR	1.20	**15.84**	1.59	1.58	**0.98**	**15.91**	2.00	2.00	**1.40**
ALS	1.20	**15.84**	**1.33**	1.58	**0.98**	**15.91**	1.58	1.11	**1.40**
PG	1.20	**15.84**	1.59	1.00	0.99	**15.91**	1.11	1.11	**1.40**
AS	1.20	**15.84**	1.44	**0.99**	1.00	**15.91**	1.11	1.11	**1.40**
BBP	**1.00**	**15.84**	**1.33**	**0.99**	1.28	**15.91**	**1.00**	1.11	**1.40**
NeNMF	**1.00**	**15.84**	**1.33**	**0.99**	1.11	**15.91**	**1.00**	**1.00**	**1.40**

PN^{3}MF	**1.00**	**15.84**	**1.33**	**0.99**	1.11	16.00	**1.00**	**1.00**	**1.40**

Table 3. CPU running time in seconds when Frobenius-norm function is used

	Verte-bral	Breast Cancer	Haber-man	Breast Tissue	Move ment Libras	ILPD	Iono-sphere	Vowel	Seg-men-tation
MUR	0.200	0.0434	0.0250	**0.0100**	**0.099**	0.12	**0.112**	**0.111**	0.113
ALS	0.200	0.0400	**0.0200**	**0.0100**	**0.099**	0.12	**0.112**	**0.111**	0.113
PG	0.100	0.0233	0.0240	0.0150	**0.099**	**0.11**	**0.112**	**0.111**	0.113
AS	0.110	0.0233	**0.0200**	**0.0100**	**0.099**	**0.11**	0.113	**0.111**	**0.112**
BBP	**0.100**	**0.0200**	0.0300	0.0340	**0.099**	**0.11**	**0.112**	**0.111**	**0.112**
NeNMF	**0.100**	**0.0200**	**0.0200**	0.0240	**0.099**	**0.11**	**0.112**	**0.111**	**0.112**
PN^{3}MF	**0.100**	**0.0200**	**0.0200**	**0.0100**	**0.099**	**0.11**	**0.112**	**0.111**	**0.112**

Table 4. Convergent objective function values in terms of Kullback-Leibler (K-L) divergence

	Verte-bral	Breast Cancer	Haber-man	Breast Tissue	Move ment Libras	ILPD	Iono-sphere	Vowel	Seg-menta-tion
MUR	1.00	1.11	1.40	1.20	**0.50**	**0.32**	1.00	1.00	1.12
ALS	1.00	1.11	1.11	1.20	**0.50**	**0.32**	1.00	**0.50**	**0.60**
PG	**0.50**	**0.99**	1.11	1.00	1.00	0.50	**0.40**	**0.50**	**0.60**
AS	**0.50**	**0.99**	1.33	1.00	1.00	0.50	**0.40**	**0.50**	**0.60**
BBP	**0.50**	**0.99**	**0.20**	1.00	1.00	1.00	1.11	1.00	**0.60**
NeNMF	**0.50**	**0.99**	**0.20**	**0.20**	**0.50**	**0.32**	1.11	1.00	1.00
PN^{3}MF	**0.50**	**0.99**	**0.20**	**0.20**	**0.50**	**0.32**	1.11	1.00	1.00

Table 5. CPU running time in seconds when Kullback-Leibler (K-L) divergence is used

	Verte-bral	Breast Cancer	Haber-man	Breast Tissue	Move-ment Libras	ILPD	Iono-sphere	Vowel	Seg-men-tation
MUR	0.030	0.0144	**0.04**	**0.030**	**0.0009**	0.02	**0.012**	**0.01**	0.03
ALS	0.030	0.0144	**0.04**	**0.030**	**0.0009**	0.02	**0.012**	**0.01**	0.03
PG	0.030	0.0155	**0.04**	**0.030**	**0.0009**	**0.01**	**0.012**	**0.01**	0.03
AS	0.030	0.0155	**0.04**	0.033	**0.0009**	**0.01**	0.013	**0.01**	**0.02**
BBP	**0.020**	0.0155	**0.04**	0.033	**0.0009**	**0.01**	**0.012**	**0.01**	**0.02**
NeNMF	**0.020**	**0.0100**	**0.04**	**0.030**	**0.0009**	**0.01**	**0.012**	**0.01**	**0.02**
PN^{3}MF	**0.020**	**0.0100**	**0.04**	**0.030**	**0.0009**	**0.01**	**0.012**	**0.01**	**0.02**

5.4 Convergent Objective Function Values vs. Iterations

We compare convergent behaviors values on the wine data set among five NMF algorithms in terms of Frobenius-norm function. Fig. 2 shows that PN^{3}MF algorithm takes the minimum number of iterations to converge on wine data set. Fig. 3 demonstrates the convergent behaviors of these algorithms on the Haberman data set in terms of Frobenius-norm function and shows that PN^{3}MF takes the minimum number of iteration to reach the convergence.

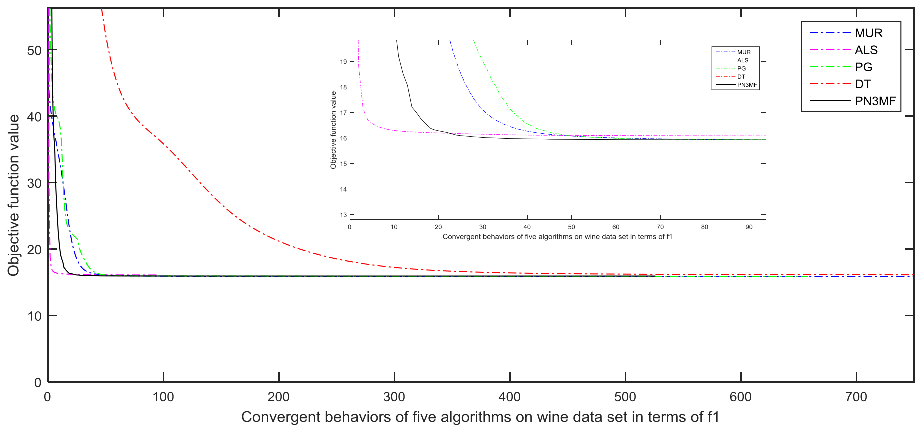

Fig. 2. Convergent behaviors of five algorithms on wine data set using Frobenius-norm function.

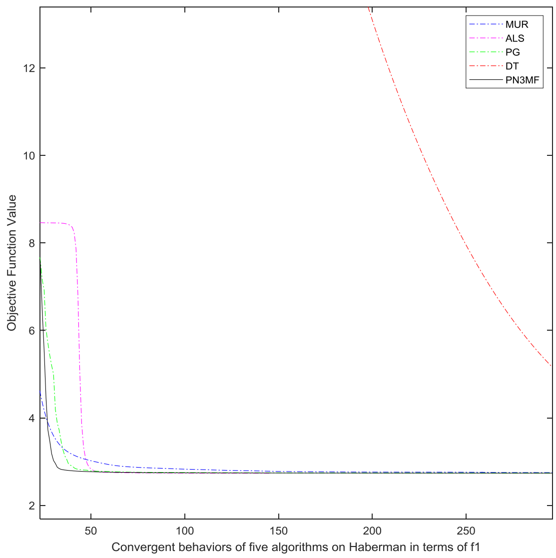

Fig. 3. Convergent behaviors of five algorithms on Haberman data set using Frobenius-norm function.

5.5 Classification Results

We further investigate the influence of various NMF algorithms on the classification accuracy as well as the performance among GPR, SVM, and ENN classifiers. In Table 6, the experimental results demonstrate that PN^{3}MF can improve the classification accuracy on most data sets. In addition, the combination of PN^{3}MF+SVM performs better than other combinations.

Table 6. Classification accuracy comparison (percentage)

	Breast Cancer	Haberman Survival	Breast Tissue	Movement Libras	Vowel	Pen Digits
GPR	96.35	97.45	95.63	98.64	97.45	100
SVM	96.45	97.45	95.45	98.45	97.35	100
ENN	96.35	97.45	95.35	98.45	97.45	97.84
PN^{3}MF+GPR	98.75	**100**	**100**	98.56	98.65	**100**
PN^{3}MF+SVM	**100**	98.75	98.75	**100**	**100**	**100**
PN^{3}MF+ENN	98.75	98.75	98.75	98.75	98.65	**100**

5.6 Confusion Matrix

We then conducted the performance evaluation by calculating the evaluation metric, including the accuracy. The evaluation metric is defined as follows (Zhang et al. 2018):

Accuracy: Accuracy = (TP+TN)/(TP+FP+FN+TN)

Where TP represents true positive (correctly identified), FP represents false positive (incorrectly identified), TN represents true negative (correctly rejected), and FN represents false negative (incorrectly rejected).

The confusion matrix module has been incorporated into the algorithms in Table 5. The classification accuracy of the confusion matrix utilizing different combinations is shown in Table 7. The way this module works is that matrix is broken down into column vectors in order to check for the prediction, number of false positives, and true positive rates. Once the simulations are completed after multiple checks, the confusion matrix true accuracy is revealed. The advantage of this module is that we get a true sense of how much the PN^{3}MF is improving performance. We also found that the combination of PN^{3}MF+SVM performs better than other combinations.

Table 7. Confusion matrix classification accuracy comparison (percentage)

	Breast Cancer	Haberman Survival	Breast Tissue	Movement Libras	Vowel	Pen Digits
GPR	96.45	97.45	98.63	99.64	98.45	100
SVM	96.45	97.45	98.45	99.45	98.35	100
ENN	96.35	97.45	98.35	99.45	98.45	98.84

PN^3MF+GPR	98.75	**100**	**100**	**100**	98.65	**100**
PN^3MF+SVM	**100**	**100**	98.75	**100**	**100**	**100**
PN^3MF+ENN	**100**	**100**	98.75	98.75	**100**	**100**

6 Conclusions

In this paper, the NMF algorithm is combined with three classifiers in order to find out the influence of dimensionality reduction performed by the NMF algorithm on the accuracy rate of the classifiers, as well as to compare the classifiers among themselves. The results show that the classification accuracy has been improved after applying the NMF algorithm. In addition, the combination of NMF algorithm with the SVM classifier performs better than other combinations. Furthermore, the confusion matrix has verified the superior classification accuracy of our NMF algorithm. In future works we plan to apply the neurodynamic approach to global and combinatorial optimization. This will open another opportunity to apply the models to more feature selection and picture restoration.

Acknowledgements. This work was supported in part by the DoD grant #W911NF1810475, National Science Foundation (NSF) grants, HRD #1505509, HRD #1533479, and DUE #1654474.

References

Berry MW, Browne M, Langville AN, Pauca VP, Plemmons RJ (2007) Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics & Data Analysis 52(1):155-173

Che H, Wang J (2018) A nonnegative matrix factorization algorithm based on a discrete- time projection neural network. Neural Networks 103(1):63-71

Che H, Wang J (2018) A collaborative neurodynamic approach to symmetric nonnegative matrix factorization. In: The 25th international conference on neural information processing (ICONIP), pp 453-462

Chellaboina V, Haddad WM (2003) Authors' reply - comments on is the frobenius matrix norm induced? IEEE Transactions on Automatic Control 48(3):519-520

Gantz J, Reinsel D (2012) The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. IDC-EMC Corporation

Gong M, Jiang X, Li H, Tan KC (2018) Multiobjective sparse non-negative matrix factorization. IEEE Transactions on Cybernetics 49(8): 2941-2954

Gorski J, Pfeuffer F, Klamroth K (2007) Biconvex sets and optimization with biconvex functions: a survey and extensions. Mathematical Methods of Operations Research 66(3):373-407

Guan NY, Tao DC, Luo ZG, Yuan B (2012) NeNMF: an optimal gradient method for nonnegative matrix factorization. IEEE Transactions on Signal Processing 60(6):2882-2898

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

Kim H, Park H (2008) Toward faster nonnegative matrix factorization: a new algorithm and comparisons. In: Proceedings of the eighth IEEE international conference on data mining, pp 353-362

Le X, Wang J (2014) Robust pole assignment for synthesizing feedback control systems using recurrent neural networks. IEEE Trans Neural Network and Learning Systems 25(2):383-393

Le X, Wang J (2014) Neurodynamic optimization approaches to robust pole assignment based on alternative robustness measures. In: Proceedings of the international joint conference on neural networks (IJCNN), pp 1-8

Le X, Wang J (2017) A two-time-scale neurodynamic approach to constrained minimax optimization. IEEE Trans Neural Network and Learning Systems 28(3):620-629

Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems 13(1):556-562

Li X, Cui G, Dong Y (2017) Graph regularized non-negative low-rank matrix factorization for image clustering. IEEE Transactions on Cybernetics 47(11):3840-3853

Lichman M (2013) UCI Machine Learning Repository. School of Information and Computer Science. Irvine, CA: Univ. California. Available: http://archive.ics.uci.edu/ml/

Lin CJ (2007) Projected gradient methods for nonnegative matrix factorization. Neural Computation 19(10):2756-2779

Tang B, He H (2015) ENN extended nearest neighbor method for pattern recognition. IEEE Computational Intelligence Magazine 10(3):52-60

Wang S, Deng C, Lin W, Huang GB, Zhao B (2017) NMF-based image quality assessment using extreme learning machine. IEEE Transactions on Cybernetics 47(1):232-243

Xia Y, Wang J (2000) On the stability of globally projected dynamical systems. Journal of Optimization Theory and Applications 106(1):129-150

Xiao Y, Zhu Z, Zhao Y, Wei Y, Wei S, Li X (2014) Topographic NMF for data representation. IEEE Transactions on Cybernetics 44(10):1762-1771

Xu B, Liu Q, Huang T (2019) A discrete-time projection neural network for sparse signal reconstruction with application to face recognition. IEEE Transactions on Neural Networks and Learning Systems 30(1):151-162

Zhang N, Karimoune W, Thompson L, Dang H (2017) A between-class overlapping coherence-based algorithm in KNN classification. In: The 2017 IEEE international conference on systems, man, and cybernetics (SMC), pp 572-577

Zhang N, Leatham K (2017) Feature selection based on SVM in photo-thermal infrared (IR) imaging spectroscopy classification with limited training samples. WSEAS Transactions on Signal Processing 13(33):285-292

Zhang N, Xiong J, Zhong J, Leatham K (2018) Gaussian process regression method for classification for high-dimensional data with limited samples. In: The 8th international conference on information science and technology (ICIST), pp 358-363