Supervised Nonnegative Matrix Factorization to Predict ICU Mortality Risk

Guoqing Chao
Feinberg School of Medicine
Northwestern University
Chicago, U.S
guoqingchao10@gmail.com

Chengsheng Mao
Feinberg School of Medicine
Northwestern University
Chicago, U.S
chengsheng.mao@northwestern.edu

Fei Wang
Weill Cornell Medicine
Cornell University
New York, U.S
few2001@med.cornell.edu

Yuan Zhao
Feinberg School of Medicine
Northwestern University
Chicago, U.S
yuan.zhao1@northwestern.edu

Yuan Luo
Feinberg School of Medicine
Northwestern University
Chicago, U.S
yuan.luo@northwestern.edu

***Abstract*—ICU mortality risk prediction is a tough yet important task. On one hand, due to the complex temporal data collected, it is difficult to identify the effective features and interpret them easily; on the other hand, good prediction can help clinicians take timely actions to prevent the mortality. These correspond to the interpretability and accuracy problems. Most existing methods lack of the interpretability, but recently Subgraph Augmented Nonnegative Matrix Factorization (SANMF) has been successfully applied to time series data to provide a path to interpret the features well. Therefore, we adopted this approach as the backbone to analyze the patient data. One limitation of the original SANMF method is its poor prediction ability due to its unsupervised nature. To deal with this problem, we proposed a supervised SANMF algorithm by integrating the logistic regression loss function into the NMF framework and solved it with an alternating optimization procedure. We used the simulation data to verify the effectiveness of this method, and then we applied it to ICU mortality risk prediction and demonstrated its superiority over other conventional supervised NMF methods.**

***Index Terms*—Nonnegative matrix factorization, Logistic regression, Supervised learning, Representation, ICU mortality risk**

I. Introduction

With the fast development of machine learning and data mining, automated predictive modeling becomes possible when combining with the increasingly available medical data in hospitals and clinical institutions. These models can discover latent patterns hidden in the data and help clinicians make timely and accurate decisions. The data generated in the Intensive Care Unit (ICU) are abundant and complicated due to the diverse variable types (especially the continuous time series data). One key event in ICU is patient death. However, the complex nature of the ICU data usually makes the clinicians difficult to make timely and correct decisions. This makes predictive modeling of the mortality risk an important problem.

In this paper, we describe a supervised nonnegative matrix factorization (NMF) algorithm that performs predictive modeling by the exploration of atomic and the higher-order features jointly. We automate the mining of higher-order features by converting data into graph representation. Moreover, supervised latent group identification reduces dimensionality of different feature types for patients, and simultaneously group temporal trends to form effective features for patient outcome prediction. Applications on patient physiological time series [1] show significant performance improvements from multiple baselines.

II. Related work

Nonnegative Matrix Factorization (NMF) refers to the set of problems on approximating a nonnegative matrix as the product of several nonnegative matrices. The problem has become popular since the work [2], where they form a nonnegative matrix by concatenating the set of pixel intensity vectors stretched from human facial images. After factorizing such matrix into the product of two matrices, they found that one matrix can be interpreted as the set of image basis with part based representation of human faces, and the other matrix has the coefficients if we were to reconstruct the face image from those bases. Because of the nonnegativity constraints, NMF is not a convex problem and they developed a multiplicative updates algorithm to obtain a stationary solution, and they prove the convergence of the algorithm in their paper [3].

Since then people have been working on NMF from various aspects. [4] showed that there is some equivalence between NMF and K-means/spectral clustering and thus claimed NMF can be used for data clustering purpose [5]. [6] further developed a t-NMF approach that can perform co-clustering on both matrix columns and rows. They also discussed the various variants of NMF in [7]. [8] extended NMF to the case when the matrix approximation loss is measured by Bregman divergence, which is a much more general loss with both

Frobenius norm and KL divergence that was introduced in [3] as its special cases. On the solution procedure aspect, multiplicative updates has been recognized for its slow convergence and poor quality. [9] reviewed the general algorithms before 2007, three classes of algorithms are categorized. The first class is multiplicative updates, the second class is gradient based methods such as [10], [11], [12], the third class is the alternating least squares (ALS) algorithm [13], [14], [15]. Also in 2007, [16] proposed an efficient projected gradient approach for NMF, which adopted Taylor expansion strategy to reduce the cost. [17] also proposed an active set type of method called principal block pivoting to solve the NMF problem. [18] adopted the alternating direction method of multipliers (ADMM) to solve the NMF with beta-divergence. Hsieh and Dhillon [19] designed a fast coordinate descent method to accelerate the FastHals [20].

Apart from basic NMF method introduced above, there are also many variants. They can be grouped into three groups. The first group enforced some constraints into basic NMF to obtain certain desirable charateristics, such as sparse NMF [10], [21], [22], orthogonal NMF [23], [24], [25], discrininant NMF [26], [27] and manifold NMF [28], [29]. The second group named structured NMF modified the standard formulation of NMF, including weighted NMF [30], convolutive NMF [31] and nonnegative matrix trifactorization [25]. The third group is the generalized NMF, including semi-NMF [32], nonnegative tensor factorization [33], matrix-set factorization [34] and kernel NMF [35]. For details, refer to the NMF survey paper [36].

Although these general NMFs decrease the dimension of the original data successfully, it cannot guarantee that the prediction ability is retained because they're unsupervised methods that have not used the label information. To tackle this problem, several supervised NMF methods were proposed. They can be classified into two categories. The first category included [26], [27], [37] is the method incorporating the idea of linear discriminative analysis to improve the prediction ability of the reduced representation, while the second category included [38], [39] is the method introducing a loss function to take the label information into consideration.

Our proposed supervised NMF belongs to the second category. The previous two methods [38], [39] enforced a frebenius loss to constrain the label information, however, it seems more like the regression not classification. Therefore, the frebenius loss constraint is not enough to exploit the label information for classification. Instead of the frebenius loss constraint, we enforce the classification loss constraint explicitly to guarantee the prediction ability of the learned representation. In fact, we proposed a supervised NMF framework that can incorporate any classification loss function. In this paper, we implement one with the logistic regression, and other classification loss functions can be considered to be integrated in furture work.

III. Methods

A. NMF and Logistic Regression

NMF [2] is a popular representation learning technique. Assume that we have data matrix $\boldsymbol{X} \in \mathcal{R}^{n\times m}$ and its corresponding label $y \in \mathcal{R}^n$. NMF aims to learn two nonnegative matrices $\boldsymbol{U} \in \mathcal{R}^{n\times r}$ and $\boldsymbol{V} \in \mathcal{R}^{r\times m}$ to approximate a nonnegative matrix $\boldsymbol{X} \in \mathcal{R}^{n\times m}$, each row of which contains m features of one of the n samples. Then NMF can be formulated in matrix form $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}$, and at the same time $\boldsymbol{U}$ and $\boldsymbol{V}$ are constrained to be nonnegative. Therefore, NMF is formulated as the following optimization problem.

$$\begin{aligned} &\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}\|_F^2 \\ &s.t. \boldsymbol{U} \succeq 0, \boldsymbol{V} \succeq 0, \end{aligned} \tag{1}$$

where $\|\cdot\|_F^2$ indicates squared frobenius norm and $\boldsymbol{U} \succeq 0, \boldsymbol{V} \succeq 0$ mean that $\boldsymbol{U}$ and $\boldsymbol{V}$ are both entry-wise nonnegative. r rows of $\boldsymbol{V}$ are considered as new bases while r columns of $\boldsymbol{U}$ are viewed as the coefficients with which the original samples can be represented with a linear combination of bases. $\boldsymbol{U}$ can be considered as a low-dimensional representation of $\boldsymbol{X}$ since generally $r < m$.

It can be seen that NMF cannot utilize the label information to learn the representation and thus it cannot guarantee the classification performance. To take the advantage of the supervised information, we will introduce the supervised learning methods, precisely, the loss function for supervised learning methods. Supervised learning is the learning paradigm that uses labels to learn, like classification. Herein, we will introduce a popular supervised learning method Logistic regression (LR).

Different from NMF, LR can make good use of the label information to classify. The optimization problem of LR can be formulated as follows:

$$\min_{\boldsymbol{w},b} \sum_{i=1}^{n} \ln\Big(1 + \exp(-y_i\big(\sum_{j=1}^{r} w_j \boldsymbol{X}_{ij} + b\big))\Big) + \frac{1}{2}\beta\big(\sum_{j=1}^{r} w_j^2 + b^2\big) \tag{2}$$

where $\boldsymbol{X}_{i,j}$ indicates the jth feature of ith data point of the $\boldsymbol{X}$, $\boldsymbol{w}$ and b indicate the weight and bias of LR. In Eq. (2) the first item is the loss function while the second one is the regularization item to prevent over-fitting. Parameter β is used to balance the loss and regularization items.

B. Supervised NMF Problem Formulation

Given data matrix $X^{n\times m}$ and label vector $y^{n\times 1}$, where n is the number of patients, m is the number of subgraphs, and the number of the labels is 1. Note that entries of y is 1 or 0. Coefficient matrix $U^{n\times r}$, basis matrix $V^{r\times m}$ and weight matrix $w^{r\times 1}$, where r is the number of subgraph groups. The supervised nonnegative matrix factorization (SNMF) can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{w},b} L_f + L_{lr} + L_r \\ &s.t. \boldsymbol{U} \succeq 0, \boldsymbol{V} \succeq 0, \end{aligned} \tag{3}$$

where the first item $L_f = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}\|_F^2$ is the loss function for NMF, the second item $L_f = \alpha \sum_{i=1}^{n} \ln\Big(1 + \exp(-y_i\big(\sum_{j=1}^{r} w_j u_{ij} + b)\big)\Big)$ is the loss function for LR and the third item $L_r = \frac{1}{2}\beta(\sum_{j=1}^{r} w_j^2 + b^2) + \frac{1}{2}\gamma\|\boldsymbol{U}\|_F^2$ is the regularization for NMF and LR.

Obviously, NMF and LR are integrated into this united framework. Since $\boldsymbol{U}$ is the new representation we aim to learn, LR works on it instead of original data matrix $\boldsymbol{X}$. The last item is used to regularize $\boldsymbol{U}$. α, β and γ are used to balance the role of those corresponding items.

C. Optimization

For the supervised NMF optimization problem (3), we can find that NMF and LR objectives are integrated together. In introduction part we have mentioned that projected gradient descent method [38] can solve this problem and gradient descent method is also the general algorithm to solve LR. we can also see that some of the variables $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{w}$ and b are interwined with each other, alternative minimization is an suitable tool to solve this kind of problems.

Our optimization will be split into four subproblem minimizations: minimize $\boldsymbol{U}$ with fixed $\boldsymbol{V}$, $\boldsymbol{w}$, and b; minimize $\boldsymbol{V}$ with fixed $\boldsymbol{U}$, $\boldsymbol{w}$, and b; minimize $\boldsymbol{w}$ with fixed $\boldsymbol{U}$, $\boldsymbol{V}$, and b; minimize b with fixed $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{w}$. These four subproblem minimizations will be alternatively executed until the predefined termination condition is satisfied. For each of the four steps, projected gradient descent or gradient descent will be adopted to solve the subproblems.

The gradients of the objective function in Eq. (3) with respect to the four variables are given below, and the detailed derivations are provided in Appendix.

$$\nabla_{\boldsymbol{U}} L = -\boldsymbol{X}\boldsymbol{V}^T + \boldsymbol{U}\boldsymbol{V}\boldsymbol{V}^T + \gamma\boldsymbol{U} - \alpha(\boldsymbol{y}\boldsymbol{w}^T) \oslash \boldsymbol{D}. \tag{4}$$

$$\nabla_{\boldsymbol{V}} L = -\boldsymbol{U}^T\boldsymbol{X} + \boldsymbol{U}^T\boldsymbol{U}\boldsymbol{V}. \tag{5}$$

$$\nabla_{\boldsymbol{w}} L = -\alpha(\boldsymbol{U} \odot \boldsymbol{Y} \oslash \boldsymbol{D})^T * \boldsymbol{e_n} + \beta\boldsymbol{w}. \tag{6}$$

$$\nabla_{\boldsymbol{b}} L = -\alpha\boldsymbol{e_n}^T * \Big(\boldsymbol{y} \oslash \big(1 + \exp(\boldsymbol{U}\boldsymbol{w} + b) \odot \boldsymbol{y}\big)\Big) + \beta b. \tag{7}$$

where $\oslash$ indicates element-wise division and $\odot$ indicates element-wise multiplication. $\boldsymbol{D}$ is obtained by repeating the column vector $(1 + \exp(\boldsymbol{U}\boldsymbol{w} + b\boldsymbol{e_n})) \odot \boldsymbol{y}$ r times to form the matrix of size $n \times r$. $\boldsymbol{Y}$ is obtained with the same operator to vector $\boldsymbol{y}$.

The update formulation of these four variables are given below:

$$\boldsymbol{U}^{t+1} = \mathcal{P}_+[\boldsymbol{U}^t - \eta_{\boldsymbol{U}}^t \nabla_{\boldsymbol{U}} L(\boldsymbol{U}^t, \boldsymbol{V}^t, \boldsymbol{w}^t, b^t)] \tag{8}$$

$$\boldsymbol{V}^{t+1} = \mathcal{P}_+[\boldsymbol{V}^t - \eta_{\boldsymbol{V}}^t \nabla_{\boldsymbol{V}} L(\boldsymbol{U}^{t+1}, \boldsymbol{V}^t, \boldsymbol{w}^t, b^t)] \tag{9}$$

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta_{\boldsymbol{w}}^t \nabla_{\boldsymbol{w}} L(\boldsymbol{U}^{t+1}, \boldsymbol{V}^{t+1}, \boldsymbol{w}^t, b^t) \tag{10}$$

$$b^{t+1} = b^t - \eta_b^t \nabla_b L(\boldsymbol{U}^{t+1}, \boldsymbol{V}^{t+1}, \boldsymbol{w}^{t+1}, b^t) \tag{11}$$

The scalar form of the problem Eq.(3) can be written in matrix form as follows.

$$\begin{aligned} &\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{w},b} L_f + L_{lr} + L_r \\ &s.t. \boldsymbol{U} \succeq 0, \boldsymbol{V} \succeq 0, \end{aligned} \tag{12}$$

where the first item $L_f = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}\|_F^2$ indicates the frobenius loss function for NMF, the second item $L_{lr} = \alpha\boldsymbol{e_n}^T * \ln\Big(1 + \exp(-(\boldsymbol{U}\boldsymbol{w} + b\boldsymbol{e_n}) \odot \boldsymbol{y}))\Big)$ indicates the loss function for LR and the third item $L_r = \frac{1}{2}\beta(\boldsymbol{w}^T\boldsymbol{w} + b^2) + \frac{1}{2}\gamma\|\boldsymbol{U}\|_F^2$ indicates the regulairzation for NMF and LR.

To understand the algorithm clear, we summarized the algorithm in Algorithm 1.

Algorithm 1: Supervised Non-Negative Matrix Factorization

```
Input: $X, Y, \alpha, \beta, \gamma, \lambda$
Output: $U, V$
Initialization: Initialize $U$ and $V$ as $U_0$ and $V_0$ with the
algorithm NNDSVD and initialize $w$ and $b$ as $w_0$ and $b_0$;
while not reach the maximal step do
    calculate $n$: the sum of the frobenius norms of $U$,
      $V$, $w$ and $b$. if n is not less than the tolerance then
        1. Update $U^t$ according to Eq. (8);
        2. Update $V^t$ according to Eq. (9);
        3. Update $w^t$ according to Eq. (10);
        4. Update $b^t$ according to Eq. (11);
    else
        break;
    end
end
```

IV. Experiments

Before applying the proposed method SNMF to the ICU mortality risk prediction problem, we verify the effectiveness of SNMF on the simulation data.

A. Simulation study

To demonstrate the effectiveness of the proposed supervised NMF method, we generate a simulation data to verify it. The data generating process includes three steps: Firstly, the coefficient matrix $\boldsymbol{U}$ is generated with two Gaussian distributions to indicate the true latent representation, the mean vectors and covariance matrices of the two Gaussian distributions are $\boldsymbol{\mu}_1 = (1, 1, 1, 1, 1)$, $\boldsymbol{\Sigma 1} = \textbf{diag}(1, 1, 1, 1, 1)$ and $\boldsymbol{\mu}_2 = (3, 3, 1, 1, 1)$, $\boldsymbol{\Sigma 2} = \textbf{diag}(1, 1, 1, 1, 1)$, respectively. 250 points are generated for each group, so that $\boldsymbol{U}_1 \in \mathcal{R}^{250\times 5}$ and $\boldsymbol{U}_2 \in \mathcal{R}^{250\times 5}$ indicate the coefficient matrices corresponding to two groups of data. In order to guarantee $\boldsymbol{U}$ nonnegative, all the negative entries are set to zero. Here, we use vector $\boldsymbol{y}$ to indicate which Gaussian distribution the point is from and form the labels of the generated data. Secondly, the basis matrix $\boldsymbol{V} \in \mathcal{R}^{5\times10}$ is generated with the Uniform distribution in $(0, 1)$ and set all the negative entries to zero to make sure $\boldsymbol{V}$ is nonnegative. Thirdly, the noise $\boldsymbol{e}$ is generated using a Gaussian distribution with the same mean and covariance matrix with $\boldsymbol{U} * \boldsymbol{V}$ and then added into the matrix $\boldsymbol{U} * \boldsymbol{V}$, see Eq. (13).

$$\boldsymbol{X} = \boldsymbol{U} * \boldsymbol{V} + \eta * e \tag{13}$$

where e is the noise from the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ indicate the mean and covariance matrix of data matrix $\boldsymbol{U} * \boldsymbol{V}$, η indicates the noise level. Final data $\boldsymbol{X}$ is $\mathcal{R}^{500\times10}$. It should be noted that all the negative entries of $\boldsymbol{X}$ are set to zero to obtain the nonnegative matrix $\boldsymbol{X}$. $\boldsymbol{y}$ takes values 0 or 1 to indicate the label. If the data point is generated from the first Gaussian distribution, its label will be 0 otherwise 1.

We split the obtained data matrix $\boldsymbol{X}$ and corresponding labels $\boldsymbol{y}$ into training and test sets with the ratio half to half, and then we compared the representation obtained from unsupervised NMF with our proposed SNMF. In addition, another supervised NMF method FNMF [27] is also added to compare. For the generated coefficient matrix $\boldsymbol{U}$, we know there are 2 distinguished dimensions, to make sure these methods can capture this distinguished information, for all the four methods, we set the dimension of the new representation as 2. For our proposed SNMF, the range of parameters α, β, γ are from $\{0, 0.001, 0.01, 0.1\}$. It should be noted that when all of α, β, γ take value 0, SNMF degenerates to NMF, and in our experimental results we adopt this as unsupervised NMF. To tune parameters efficiently, the parameters α, β, γ are set to α/p, β/r and $\gamma/(n \times r)$ where p is the number of the features of data matrix $\boldsymbol{X}$, r is the number of the features of the latent representation matrix $\boldsymbol{U}$ and n is the number of the samples of data matrix $\boldsymbol{X}$. To further explore how these methods perform on the data with different noise level, we conduct the experimental comparison on the simulation data with three noise level 0, 0.2, and 0.5. The metric we adopt is the area under ROC (receiver operating characteristic) curve (AUC). All the experimental results are demonstrated in Table I, the best result is shown in bold font for each case.

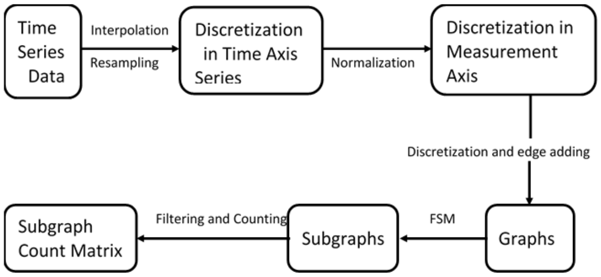
TABLE I
THE AUC ON THE SIMULATION DATA (%), THE VALUE IN BOLD INDICATES THE DIFFERENCE WITH THE REST IS SIGNIFICANT (p VALUE <0.01).

Noise Level	Data Split	NMF	FNMF	SNMF
$\eta = 0.5$	Test	90.35	89.19	**91.78**
$\eta = 0.2$	Test	93.91	94.40	**95.68**
$\eta = 0$	Test	95.23	95.96	**97.15**

From Table I, we can find that on the data without any noise, FNMF performs better than NMF but worse than our SNMF, because FNMF learned the new representation with the help of the label information while NMF doesn't. Compared with SNMF, FNMF just guarantees the data points within the same class close and those in different classes far away in the new representation but it still doesn't use the label information directly to approach the final goal classification, so SNMF is more effective. With increasing the noise level, we can see all the methods' performance degrade, but SNMF outperforms all the other methods all the time. During all the comparisons, NMF performs worse than SNMF is because when it learned the 2 dimensional new representation, it maybe not exploit the distinguished dimensional information sufficiently because of its unsupervised property, but for SNMF, it makes full use of that distinguished dimensional information due to the classification loss item in the Eq. (3).

B. ICU Mortality Risk Prediction

1) Data Processing: The data we adopted to predict ICU mortality risk is processed as in Paper [1]. To make the data processing clearly, a schematic of the data processing workflow is illustrated in Figure 1. The time series data from the second half of the first day after patients' admissions to ICU are the original data we need to process. To address the issues of missing data and irregular sampling interval, we performed linear interpolation and resampling at regularly spaced time intervals on the original time series to discretize in the time axis. Although a more complex imputation algorithm may be more plausible [40], [41], we follow the SANMF approach to enable fair comparison. Note that the time interval 2 hours are determined by 5-fold cross validation over choices of 1, 2, 4, or 6 hour intervals. For detailed processing information, the authors can refer to Paper [1].



FSM: Frequent Subgraph Mining

Fig. 1. Schematic of the time series data processing.

2) Experiment Setting and Result: The processed subgraph count data are split equally, stratified by mortality, into the training set and test set. There are 3932 cases in the training set while 3931 cases in the test set. The statistics of the data are demonstrated in Table II. For our proposed SNMF, we need to learn the new representation $\boldsymbol{U}$ from the training data and then train the logistic regression on top of $\boldsymbol{U}$ to check its performance on the $\boldsymbol{U}$ obtained on test data. Besides parameter r (the number of groups), our SNMF still have parameters

α, β, γ, we will use 5-fold cross validation to identify the parameters α, β, γ. The range of parameters α, β, γ are from -2 to 2 with exponential base 10. To tune these parameters efficiently, the parameters α, β, γ are set to α/p, β/r and $\gamma/(n \times r)$, which is the sampe with that in simulation data. For the number of groups, we will follow the way in [1], and extend its range if the performance will still rise up to 120. From the subsquent Figure 2, we can find that the range for number of groups are from 50 to 150 (We did not show the performance in range 10 to 40 because their performance is worse than the current range). The metric we adopt is the area under ROC (receiver operating characteristic) curve (AUC). All the experimental results are demonstrated in Figure 2.

From Figure 2, we can see that almost with each group number, our proposed SNMF outperforms NMF, this should because SNMF take the label information into consideration when learning the low-dimensional representation while NMF doesn't. In some case NMF can perform a bit better than SNMF on the hold-out test data, because there are additional 3 parameters to tune for SNMF and the distributions between training and test data may be inconsistent and this may cause the parameters identified during cross validation not perform well on the hold-out test data. Figure 2 shows that the best test AUC 0.8562 for SNMF occurs at 120 groups while that best test AUC 0.8508 for NMF happens at 60 groups, and the superiority of SNMF over NMF on ICU data is statistically significantly (p-value of permutation test is 0.0307).

TABLE II
STATISTICS OF EXPERIMENTAL DATA

Mortality	Total Cases	Training Cases	Test Cases
$\leq$ 30 days	788	383(9.7%)	405(10.3 %)
$>$ 30 days or alive	7075	3549(90.3%)	3526(89.7 %)

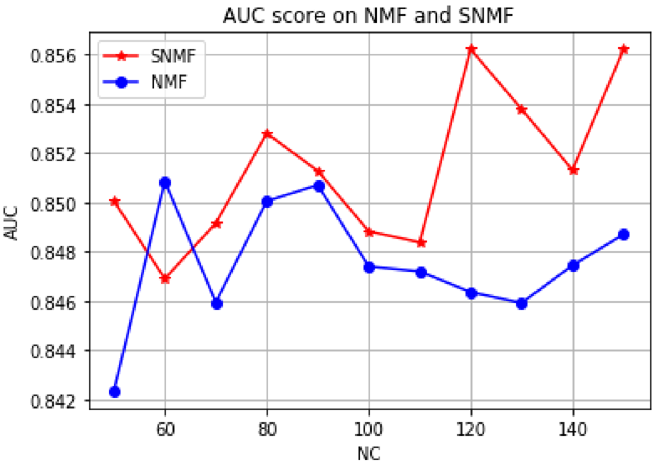


Fig. 2. Average AUC comparison under different number of groups on the hold-out test data.

V. Discussion

From the classification results on the ICU data, our SNMF indeed performs better than the unsupervised NMF counterpart from the statistic significancy perspective. However, the benifit is not big, this maybe because the information NMF retained has already possessed discriminative classification prediction ability.

As for the ICU data, its original data form is time series, the processing workflow proposed in paper [1], [42] indeed opened a feasible way to deal with the prediction problem easily, meanwhile, some discriminative information may be lost during the process, such as the interpolation and sampling phases may be not precise. A direct way to deal with time series data and conduct classification is interesting and promising. In addition, feature selection [43], [44] is closely related to the present work, it can be further compared with the proposed method while working on time series data directly in future work.

There are several different data types of ICU data, like vital sign and lab test information, they can be considered as multi-view information, and then multi-view learning methods [45], [46], [47], [48] can be considered to be adopted to solve this problem.

VI. Conclusions

In this paper, we have proposed a supervised NMF method to learn a discriminative representation, based on which the classification performance is improved compared with its unsupervised NMF counterpart and other supervised NMF method. We adopted the projected gradient descent algorithm to solve this problem. The results on synthetic data and ICU data verified its superiority. The learned representation with our method has better prediction ability, which can guide the clinician well in reality.

Since we just explored to integrate NMF with the simple classifier logistic regresssion, other advanced classifier (classification loss function) can be used to replace logistic regression. For instrance, as with the fast development of deep learning, it is promising to combine deep neural network with NMF to learn a more discriminative representation to boost the prediction performation of the ICU mortality risk further.

References

[1] Y. Luo, Y. Xin, R. Joshi, L. Celi, and P. Szolovits, "Predicting icu mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements," in *Proceedings of Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 42–50.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 19999.

[3] ——, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems (nips)*, 2001, pp. 535–541.

[4] C. Ding, X. He, and H. D, Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005.

[5] G. Chao, "Discriminative k-means laplacian clustering," *Neural Processing Letters*, pp. 1–13, 2018.

[6] T. L. W. P. Ding, Chris and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[7] T. L. Ding, Chris HQ and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 45–55, 2010.

[8] S. S. Inderjit Dhillon, "Generalized nonnegative matrix approximations with bregman divergences," in *Advances in neural information processing systems (nips)*, DEC 2005, pp. 283–290.

[9] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational statistics & data analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[10] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[11] M. Chu, F. Diele, R. Plemmons, and S. Ragni, "Optimality, computation, and interpretation of nonnegative matrix factorizations," in *SIAM Journal on Matrix Analysis*. Citeseer, 2004.

[12] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear algebra and its applications*, vol. 416, no. 1, pp. 29–47, 2006.

[13] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[14] P. Paatero, "The multilinear enginea table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model," *Journal of Computational and Graphical Statistics*, vol. 8, no. 4, pp. 854–888, 1999.

[15] A. N. Langville, C. D. Meyer, R. Albright, J. Cox, and D. Duling, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," *arXiv preprint arXiv:1407.7299*, 2014.

[16] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[17] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2011, p. 32613281.

[18] D. L. Sun and C. Fevotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 6201–6205.

[19] I. S. D. Cho-Jui Hsieh, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, AUG 2011, pp. 1064–1072.

[20] A. Cichocki and A.-H. 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.

[21] P. O. Hoyer, "Non-negative sparse coding," in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*. IEEE, 2002, pp. 557–565.

[22] M. Morup, K. H. Madsen, and L. K. Hansen, "Approximate l 0 constrained non-negative matrix and tensor factorization," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 1328–1331.

[23] Z. Li, X. Wu, and H. Peng, "Nonnegative matrix factorization on orthogonal subspace," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 905–911, 2010.

[24] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 126–135.

[25] J. Yoo and S. Choi, "Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds," *Information processing & management*, vol. 46, no. 5, pp. 559–570, 2010.

[26] C. H. Y. Wang, Y. Jia and M. Turk., "Fisher non-negative matrix factorization for learning local features." in *Asian Conference on Computer Vision*, 2004, pp. 27–30.

[27] B. I. P. I. Zafeiriou S., Tefas A., "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683–695, 2006.

[28] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 63–72.

[29] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[30] Y.-D. Kim and S. Choi, "Weighted nonnegative matrix factorization," 2009.

[31] P. D. O'grady and B. A. Pearlmutter, "Convolutive non-negative matrix factorisation with a sparseness constraint," in *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*. IEEE, 2006, pp. 427–432.

[32] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 45–55, 2010.

[33] A. Cichocki, R. Zdunek, and S.-i. Amari, "Nonnegative matrix and tensor factorization [lecture notes]," *IEEE signal processing magazine*, vol. 25, no. 1, pp. 142–145, 2008.

[34] L. Li and Y.-J. Zhang, "Non-negative matrix-set factorization," in *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*. IEEE, 2007, pp. 564–569.

[35] D. Zhang, Z.-H. Zhou, and S. Chen, "Non-negative matrix factorization on kernels," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2006, pp. 404–412.

[36] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[37] W.-S. C. W. Z. Yun Xue, Chong Sze Tong, "A modified non-negative matrix factorization algorithm for face recognition," in *Proceedings of the 18th International Conference on Pattern Recognition*, 2006, pp. 495–498.

[38] S. C. Hyekyoung Lee, Jiho Yoo, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 4–7, 2010.

[39] C. Z. Liping Jing and M. K. Ng, "Snmfca: Supervised nmf-based image classification and annotation," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4508–4521, 2012.

[40] Y. Luo, P. Szolovits, A. S. Dighe, and J. M. Baron, "Using machine learning to predict laboratory test results," *American journal of clinical pathology*, vol. 145, no. 6, pp. 778–788, 2016.

[41] ——, "3d-mice: integration of cross-sectional and longitudinal imputation for multi-analyte longitudinal clinical data," *Journal of the American Medical Informatics Association*, vol. 25, no. 6, pp. 645–653, 2017.

[42] Y. Xue, D. Klabjan, and Y. Luo, "Predicting icu readmission using grouped physiological and medication trends," *Artificial intelligence in medicine*, 2018.

[43] Y. Dai, B. Hu, Y. Su, C. Mao, J. Chen, X. Zhang, P. Moore, L. Xu, and H. Cai, "Feature selection of high-dimensional biomedical data using improved sfla for disease diagnosis," in *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 458–463.

[44] B. Hu, Y. Dai, Y. Su, P. Moore, X. Zhang, C. Mao, J. Chen, and L. Xu, "Feature selection for optimized high-dimensional biomedical data using the improved shuffled frog leaping algorithm," *IEEE/ACM transactions on computational biology and bioinformatics*, 2016.

[45] G. Chao and S. Sun, "Consensus and complementarity based maximum entropy discrimination for multi-view classification," *Information Sciences*, vol. 367, pp. 296–310, 2016.

[46] ——, "Multi-kernel maximum entropy discrimination for multi-view learning," *Intelligent Data Analysis*, vol. 20, no. 3, pp. 481–493, 2016.

[47] Y. Luo, F. S. Ahmad, and S. J. Shah, "Tensor factorization for precision medicine in heart failure with preserved ejection fraction," *Journal of cardiovascular translational research*, vol. 10, no. 3, pp. 305–312, 2017.

[48] Y. Luo, F. Wang, and P. Szolovits, "Tensor factorization toward precision medicine," *Briefings in bioinformatics*, vol. 18, no. 3, pp. 511–514, 2016.

Appendix

For convenience to derive the gradients with respect to each variable, let $L_X = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}\|_F^2 + \frac{1}{2}\gamma\|\boldsymbol{U}\|_F^2$ and $L_y = \alpha\sum_{i=1}^{n}\ln\Big(1+\exp(-y_i\big(\sum_{j=1}^{r} w_j u_{ij} + b\big))\Big) + \frac{1}{2}\beta(\sum_{j=1}^{r} w_j^2 + b^2)$, $L = L_X + L_y$ denotes the objective function in Eq (12) .

$$\begin{aligned}
\nabla_{\boldsymbol{U}} L_X &= \frac{1}{2}\nabla_U\big(\mathrm{Tr}\big((\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V})(\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V})^T\big) + \gamma\mathrm{Tr}(\boldsymbol{U}\boldsymbol{U}^T)\big)\\
&= \frac{1}{2}\nabla_{\boldsymbol{U}}\big(\mathrm{Tr}(-\boldsymbol{X}\boldsymbol{V}^T - \boldsymbol{U}\boldsymbol{V}\boldsymbol{X} + \boldsymbol{U}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{U}^T) + \gamma\mathrm{Tr}(\boldsymbol{U}\boldsymbol{U}^T)\big)\\
&= \frac{1}{2}(-2\boldsymbol{X}\boldsymbol{V}^T + 2\boldsymbol{U}\boldsymbol{V}\boldsymbol{V}^T + 2\gamma\boldsymbol{U})\\
&= -\boldsymbol{X}\boldsymbol{V}^T + \boldsymbol{U}\boldsymbol{V}\boldsymbol{V}^T + \gamma\boldsymbol{U}
\end{aligned} \tag{14}$$

$$\begin{aligned}
\nabla_{\boldsymbol{V}} L_X &= \frac{1}{2}\nabla_V\Big(\mathrm{Tr}\big((\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V})(\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V})^T\big))\Big)\\
&= \frac{1}{2}\nabla_{\boldsymbol{V}}\Big(\mathrm{Tr}(-\boldsymbol{X}\boldsymbol{V}^T - \boldsymbol{U}\boldsymbol{V}\boldsymbol{X}^t + \boldsymbol{U}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{U}^T)\Big)\\
&= \frac{1}{2}(-2\boldsymbol{u}^T\boldsymbol{X} + 2\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{V})\\
&= -\boldsymbol{U}^T\boldsymbol{X} + \boldsymbol{U}^T\boldsymbol{U}\boldsymbol{V}
\end{aligned} \tag{15}$$

$$\begin{aligned}
\nabla_{w_j} L_y &= \alpha\sum_{i=1}^{n}\frac{\exp\big(-y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big) * (-1) * y_i u_{ij}}{1 + \exp\big(-y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big)}\\
&\quad + \beta w_j\\
&= -\alpha\sum_{i=1}^{n}\frac{y_i u_{ij}}{1 + \exp\big(y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big)} + \beta w_j
\end{aligned} \tag{16}$$

$$\nabla_{\boldsymbol{w}} L_y = -\alpha(\boldsymbol{U} \odot \boldsymbol{Y} \oslash \boldsymbol{D})^T * \boldsymbol{e_n} + \beta\boldsymbol{w} \tag{17}$$

The denominator matrix $\boldsymbol{D}$ is obtained by repeating the column vector $(1 + \exp(\boldsymbol{U}\boldsymbol{w} + b\boldsymbol{e_n})) \odot \boldsymbol{y}$ r times to form the matrix of size $n \times r$. $\boldsymbol{Y}$ is obtained with the same operator to vector $\boldsymbol{y}$.

$$\nabla_{u_{ij}} L_y = \alpha\frac{\exp\big(-y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big) * (-1) * y_i w_j}{1 + \exp\big(-y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big)} \tag{18}$$

It can be written in matrix form as follows.

$$\nabla_{\boldsymbol{U}} L_y = -\alpha(\boldsymbol{y}\boldsymbol{w}^T) \oslash \boldsymbol{D} \tag{19}$$

The denominator matrix $\boldsymbol{D}$ is the same with that in Eq. (17).

$$\begin{aligned}
\nabla_b L_y &= \alpha\sum_{i=1}^{n}\frac{\exp\big(-y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big) * (-1) * y_i}{1 + \exp\big(-y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big)} + \beta b\\
&= -\alpha\sum_{i=1}^{n}\frac{y_i}{1 + \exp\big(y_i(\sum_{j=1}^{r} w_j u_{ij} + b)\big)} + \beta b.
\end{aligned} \tag{20}$$

It can be written in matrix form as follows.

$$\nabla_b L_y = -\alpha\boldsymbol{e_n}^T * \Big(y \oslash \big(1 + \exp(\boldsymbol{U}\boldsymbol{w} + b) \odot \boldsymbol{y}\big)\Big) + \beta b. \tag{21}$$