Application of Data Classification Method Based on Non-Negative Matrix Factorization

LingJiao Li, YingLin Liu, HongYuHuang, XiaoPing Fan*
Hunan University of Finance and Economics, Changsha, 410205, China

*1343200106@qq.com

Abstract: At present, there are many methods for data dimensionality reduction, but most of the results of decomposition methods allow negative values. Obviously, these negative values have no physical meaning in practical problems. The non-negative matrix factorization method is a dimensionality reduction under the condition of ensuring non-negative values. This paper mainly uses a non-negative matrix factorization algorithm to perform dimensionality reduction representation and local feature extraction of data, and then classify. Experiments show that the algorithm in this paper is reasonable and effective.

Keywords: Non-negative matrix factorization, data dimensionality reduction, data classification

1. Introduction
In artificial intelligence, classification algorithms undoubtedly have a pivotal position. With the increase of data, the traditional methods and corresponding methods in the past cannot meet the current large-scale data mining. Important content, and how to better distinguish high-dimensional data plays key parts in the relevant analysis. The current non-negative matrix factorization has evolved in many ways into a new form of analysis within the current machine learning range because this kind of matrix factorization has its own characteristics and can transform data operations and correlation levels on a scale, so they will re-neutralize the development needs in the future.

2. Related Works
In practice, high-dimensional data will appear in many cases, and machine algorithms that judge the degree of correlation under the premise of distance cannot carry out relevant mining for high-dimensional data. In this case, with its own characteristics, non-negative matrix factorization can bring new forms in the search and discovery of high-dimensional data.

Scholars such as Liu and Zheng invented the sparse NMF based on the Lagrangian multiplier method, and then the scholar Hoyer P. O immediately invented the NMF algorithm under the sparse constraint[1]. In addition to sparse NMF, scholars such as Li first gave the idea of strengthening orthogonalization constraints[2], so that the basis vectors of the basis matrix itself can be orthogonalized to the greatest extent. The so-called equidistant NMF can also be included in the manifold range to a large extent. This type is advocated by Vasiloglou[3].

3. Classification Algorithm Based On Non-Negative Matrix Factorization
3.1. Naive NMF Classification Algorithm

Generally speaking, the classification algorithm is to learn from the labeled data, and then build the corresponding classifier according to the results, so as to achieve the purpose of automatically classifying the unlabeled data. With the increase in the dimensions, the algorithm itself will take time. It will be significantly extended, and the algorithm based on the non-negative matrix classification can be better used in high-dimensional data.

3.1.1. Definition And Explanation

The two classification modes of the naive NMF classification algorithm are:

\[ v \approx Wh \]  

(3.1)

Here, the vector \( v \in \mathbb{R}^{M \times 1} \) is the class label matrix, M is the number of data points, the matrix \( v \in \mathbb{R}^{M \times N} \) is the original data matrix, N is the number of original data features, and the vector \( h \in \mathbb{R}^{M \times 1} \) is the probability matrix.

In addition, observing the probabilistic interpretation of the naive NMF classification algorithm, it can be found that this algorithm does not use Euclidean distance. Instead, it determines whether all features and class labels are based on each characteristic content. The degree of correlation between.

Among them, this article will use the bag-of-words mode of the text itself as an analysis method. The initial data matrix is the word vector matrix in the text. Then the probability of the text data classification model can be described by the following formula:

\[
P(label_{document}) = \sum_{i} P(word_{i}, document) \times (label_{word_{i}})
\]  

(3.2)

3.1.2. Algorithm Framework

With the help of the implementation of the standard NMF iteration rules, we can find similar explanations about H, and H is the classification tool, and with the help of W and H, we can find the label matrix about the test set itself.

In other words, just iterate to clarify H, that is:

\[
H_{ij} = H_{ij} \times \frac{(W^TW)_{ij}}{(W^TWH)_{ij}}
\]  

(3.3)

Since it occurs in the iterative resolution stage, the original matrix H must exist. Under normal circumstances, the factors that constitute H exist above 0 and below 1, so the initialization of the data in this range without target setting is sufficient to meet the demand, but because of the possibility of randomness, it is possible to repeat the original. In order to construct a variety of classification models, the degree of randomness is weakened as much as possible.

Specifically, the so-called naive NMF algorithm is composed of: determining the approximate solution of H according to the existing matrices \( W_{train} \) and \( W_{trans} \) with the help of related rules. After this, according to the correlation matrix \( W_{test} \) and H, \( V_{test} \) can be determined. Then follow the two classification algorithm to clarify the most suitable distinction boundary, and finally clarify the matrix \( V_{test} \).
Algorithm 1 Naive NMF Classification Algorithm

Input: $V_{\text{train}}$: $V_{\text{train}} \in \mathbb{R}_{\geq 0}^{M_{\text{train}} \times L}$, training set label matrix
$W_{\text{train}}$: $W_{\text{train}} \in \mathbb{R}_{\geq 0}^{M_{\text{train}} \times N}$, training set feature matrix
$W_{\text{test}}$: $W_{\text{test}} \in \mathbb{R}_{\geq 0}^{N_{\text{test}} \times N}$, test set feature matrix

Output: test set label matrix $V_{\text{test}}$

Iterate:
Solve based on $V_{\text{train}}$, $W_{\text{train}}$, and formula (3.3) to get $H$
Solve based on $W_{\text{train}}$, $H$, and formula (3.1) to get $V_{\text{train}}$
for $i = 1$: $L$ do
Based on the $i$-th column vector of $V_{\text{train}}$ and $V_{\text{train}}$, get the corresponding class threshold $\theta_i$
Based on $W_{\text{test}}$, $H$, and formula (3.1), $V_{\text{test}}$ is obtained.
Based on $V_{\text{test}}$ and $\theta_i$, get $V_{\text{test}}$

3.1.3. Algorithm Analysis And Expansion
What we can find is that the label probability of the sample itself is largely affected by the linear weighting of all features and because this probability value can only exist in 0-1, but NMF naturally requires that all matrices should be Covered in non-negative, so in this state is beneficial to carry out classification experiments.

The data in higher dimensions, because the data itself is relatively sparse, and there is more serious tilt, in this state, no matter what kind of sample, its own characteristics are largely lower than All samples have an average dimension, so the naive NMF algorithm works best when applied to sparse and higher-dimensional data.

3.2. Application And Extension
The standard NMF algorithm cannot be attributed to the supervised dimensionality reduction algorithm, but feature selection can be attributed to this method under certain circumstances. The lack of supervision of the NMF dimensionality reduction algorithm is feature-mapped, that is, the space in the higher dimensionality is converted to the lower-dimensional space by means of mapping, and then the feature vector in this kind of space is used to compare Data in high dimensions are represented.

Algorithms based on feature selection rely on feature calculations, and with the help of features with a higher degree of correlation, correspondingly discarding features with a lower degree of correlation.

4. Experiment

4.1. Data Set
This part uses 4 benchmark data sets, namely: Connectionist bench, Ionosphere, Genbase, LangLog, the first two are binary classification data sets, and the last two are multi-label classification data sets. Some basic statistics about the data set are shown in Tab.2.

| Dataset      | # of Instance | # of Attributes | # of Classes | Sparsity |
|--------------|---------------|-----------------|--------------|----------|
| Connectionist| 208           | 60              | 2            | No       |
| Ionosphere   | 351           | 34              | 2            | No       |
| Genbase      | 661           | 294             | 6            | Yes      |
| LangLog      | 1460          | 1001            | 75           | Yes      |

Tab.2 Statistics of the benchmark data set
4.2. Evaluation Index

4.2.1. Two classification algorithm
In this kind of algorithm, ROC curve, accuracy rate, recall rate, F1 value, precision rate, and AUC are all used more measurement standards. In this article, these indicators will be used to evaluate the algorithm.

The abscissa of the ROC curve is FPR, which is the false positive rate, and the ordinate is TPR, which is the true rate. AUC refers to the area under the ROC curve, which is usually between 0.5 and 1.0. The closer this value is to 1, the better it can show that this algorithm has a better performance in classification, and the accuracy is also maintained.

| Actual | Predict | 1     | 0     |
|--------|---------|-------|-------|
| 1      | True Positive(TP) | False Positive(FP) |
| 0      | False Negative(FN) | True Negative(TN) |

Fig.1 Confusion Matrix

From Fig.1, we can see that for this part of the two classification algorithms, according to the actual label and predicted performance, there will be many different situations. In the experiment, the real label belongs to the positive class, and the predicted label still belongs to the positive class, so it is the real class in the general sense like TP. In the experiment, the true label belongs to the negative class, and the predicted label also belongs to the negative class, so it is the true negative class in the general sense like FN.

According to the above definition, the accuracy can be defined as:

\[
\text{precision} = \frac{TP}{TP + FN}
\]

(4.1)

The recall rate is defined as:

\[
\text{recall} = \frac{TP}{TP + FN}
\]

(4.2)

In general, there is a negative correlation between accuracy and recall. The higher the accuracy, the lower the recall. Conversely, the lower the accuracy, the higher the recall. The F1 value represents the harmonic average of accuracy and recall. It effectively balances the relationship between accuracy and recall. Generally, the larger the F1 value, the better the classifier.

The F1 value is defined as:

\[
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

(4.3)

The accuracy rate and recall rate are mainly for the classification effect of positive samples. For the classification effect of all samples, the accuracy rate is a commonly used evaluation index.
The definition of accuracy is:

\[
\text{accuracy} = \frac{TP + TN}{TP + FN + FP + TN}
\]  

(4.4)

But in the normal state, the addition is not as good as the former in the index sense. This is because in the field of imbalanced data classification when one of the positive sample and the negative sample replace a larger proportion, the accuracy is still very high, but the accuracy is meaningless and optional.

4.2.2. Multi-label classification algorithm
In this kind of algorithm, there is a big difference between the frequently used evaluation criteria and the classification algorithm. The so-called multi-label classification generally means that any sample in a certain sample data has one or more labels. In practice, it is very common for a data set to have many labels, and a text data set is a typical example of this label. Next, this article will focus on analyzing the measurement indicators of multi-label classification that are used more in practice.

4.3. Experimental Results And Comparison

4.3.1. Low-dimensional data set
In this part of the experiment, the 52 fold is used for repeated verification, that is, the same experiment is carried out 5 times, and mutual verification is carried out with each 2 fold. All the data in the experiment will use these average values.

First of all, this article selects the KNN algorithm and the naive NMF classification algorithm for comparative experiments on the low-dimensional data sets Connectionist Bench and Ionosphere. In this comparative experiment, this paper chose the KNN algorithm with K=10 and K=20 to compare with the naive NMF classification algorithm. As shown in Tab.3 and Tab.4.

| Algorithms | Precision | Recall | FI     | AUC     |
|------------|-----------|--------|--------|---------|
| KNN (K=10) | 1         | 0.9591 | 0.9792 | 0.9787  |
| KNN (K=20) | 1         | 0.9216 | 0.9592 | 0.9574  |
| NMF        | 1         | 1      | 1      | 1       |

Tab.3 Experimental results of the data Connectionist Bench set

| Algorithms | Precision | Recall | FI     | AUC     |
|------------|-----------|--------|--------|---------|
| KNN (K=10) | 0.8879    | 1      | 0.9406 | 0.8833  |
| KNN (K=20) | 0.8362    | 1      | 0.9108 | 0.8166  |
| NMF        | 1         | 1      | 1      | 1       |

Tab.4 Experimental results of the Ionosphere data set

What we can find from this is that the naive NMF algorithm is better than the KNN algorithm in terms of effect. In addition, the algorithm is also well used in data classification in lower dimensions.

4.3.2. Multi-label data set
For the purpose of better analysis of the classification algorithm based on NMF, this paper selects three large-scale high-dimensional multi-label data sets as objects and conducts various comparison experiments. The three objects are LangLog, TMC2007-500, Delicious. The last one has very low sparsity.
This article chooses the multi-label data set Genbase and compares the common classification algorithm with the classification algorithm based on NMF. This comparison test ride selected Accuracy and Hamming Loss. The common classification algorithm selects the K nearest neighbor algorithm.

| Algorithms | KNN | Naive NMF | Improved NMF |
|------------|-----|-----------|--------------|
| Accuracy   | 0.989 | 0.989 | 0.989 |
| Hamming Loss | 0 | 0.001 | 0.001 |

*Tab.5 Experimental results of the Genbase data set*

Genbase is a low-dimensional multi-label data set. By the above, we can see that in the low-dimensional data classification algorithm, the NMF classification algorithm is better in the classic algorithm.

In order to comprehensively evaluate the classification algorithm based on NMF, this paper selects a large-scale high-dimensional multi-label data set. In this group of experiments, this article compares the naive NMF classification algorithm, the improved NMF classification algorithm, and the classic multi-label classification algorithm ML-KNN on a multi-label data set and counts the Accuracy, Hamming Loss, and Average precision indicators of related algorithms, such as As shown in Tab.6.

| Algorithms | Naive NMF | Improved NMF | ML-KNN |
|------------|-----------|--------------|--------|
| LangLog    | 0.022 | 0.023 | 0.016 |
|            | 0.393 | 0.389 | 0.284 |
|            | 0.214 | 0.211 | 0.113 |

*Tab.6 Hamming Loss, Average precision, and Accuracy indicators of the data set*

According to the content presented by the experiment, we can see that the naive NMF algorithm is well presented in Accuracy, but in Hamming Loss, there is a certain gap compared with the traditional algorithm. In Average precision, the naive NMF classification algorithm is effective, followed by the optimized NMF algorithm, the improved NMF classification algorithm has better performance for extremely sparse data sets.

5. Conclusion
This paper presents the construction of a classification algorithm under the premise of non-negative matrix factorization, when classifying high-dimensional data sets, the framework proposed in this paper is superior to past algorithms in terms of efficiency. In addition, under the premise of having a consistent label data set, a semi-supervised classification algorithm based on NMF is given. From the perspective of effect, it is obviously stronger than the naive NMF classification algorithm, and it should meet actual needs. In the classification algorithm based on NMF, a comparative analysis is carried out on different data sets, and the results also tell us that it can give better classification and take less time.

Acknowledgment
This work was supported in part by the National Natural Science Foundation of China, grant number 72073041. Open Foundation for the University Innovation Platform in the Hunan Province, grant number 18K103; 2011 Collaborative Innovation Center for Development and Utilization of Finance and Economics Big Data Property, Universities of Hunan Province. Open project, grant number 20181901CRP03, 20181901CRP04, 20181901CRP05. 2020 Hunan Provincial Higher Education Teaching Reform Research Project under Grant HNJG-2020-1124, HNJG-2020-1130. 2020 General Project of Hunan Social Science Fund under Grant 20B16
Reference

[1] Hoyer P 2004 Non-negative matrix factorization with sparseness constraints *J. Machine Learning Research* pp1457-1469

[2] Li S Z and Hou X W et al 2001 Learning spatially localized, parts-based representation *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* pp 207-212

[3] Vasiloglou N, Gray A G and Anderson D V 2009 Non-negative matrix factorization convexity and isome-try (*SIAM* *Int. Conf. on Data Mining* pp 673-684)