Method for Extraction and Fusion Based on KL Measure

Zuocong Chen(✉)

College of Computer Science and Technology,
Hainan Tropical Ocean University, Sanya 572022, China
twsf2005@163.com

Abstract. Feature extraction and fusion is important in big data, but the dimension is too big to learn a good representation. To learn a better feature extraction, a method that combines the KL divergence with feature extraction is proposed. Firstly the initial feature was extracted from the primitive data by matrix decomposition. Then the feature was further optimized by using KL divergence, where KL divergence was introduced to the loss function to make the goal function with the shortest KL distance. The experiment is implemented in four datasets such as COIL-20, COIL-100, CBCI 3000 and USPclassifyAL. The result shows that the proposed method outperforms the other four methods in the accuracy when using least number of features.

Keywords: Feature extraction · KL divergence · Samples · Loss function

1 Introduction

It is very known much useful and valuable information is included in the big data [1]. However, the dimensionality of the data is always big, so it is hard to learn from them. Feature extraction is an important research area in pattern recognition and machine learning for big data. Learning important features from these data can not only reduce the computation complexity but also improve the learning ability of the algorithm. A robust feature representation for big data can make the learning model be better obtained.

The traditional methods for feature extraction and fusion always starts at the view of the statistics [2]. With the development of the information technology and network, the feature extraction is attracting more and more attentions. Feature extraction algorithm can be thought as a composite part for the model learning, and it can be divided into three parts: supervised, semi-supervised and unsupervised. The supervised method refers to that all labeled-data are considered and feed to train the model. If some of the labeled-data are used to train the model, we say the method is semi-supervised. No labeled-data are considered in the unsupervised method. All the three methods are preconditioned that the distance among the data represent the similarity property. The distance in feature extraction is always measured by Euclidean distance. However, a large error will be generated when using the Euclidean distance due to the elements of every data have different units.

X. Cheng et al. (Eds.): ICPCSEE 2019, CCIS 1058, pp. 42–51, 2019.
https://doi.org/10.1007/978-981-15-0118-0_4

In order to get a better feature extraction representation for big data. We propose a better measure method based on KL distance, and designed a loss function for it. From the experiment, it can be shown that our method has better effect on loss function.

2 Related Work

2.1 Traditional Methods

Traditional feature extraction methods are composed of feature sort method and feature search method. The earliest research on feature sort are mainly based the distance measure methods such as Relief algorithm, the improved ReliefF [3] and the mutual information based DMIFS algorithm [4]. The advantage of the feature sort method is the high execution efficiency. However, two problems hinder the development, one is the number of features should be assigned in advance, and the other is that the feature set may be composed of m optimal features. Furthermore, the feature sort method only evaluates the relevance of the feature itself and the label other than considering the relations among features, so that the redundant feature cannot be distinguished.

For the dataset with high dimensionality, the feature extraction cannot get an effective result because of the redundant features. In order to distinguish the redundant features and then search the optimal feature set, many search algorithms for obtaining the feature subset is proposed. The feature subset search algorithm can be divided into two kinds: comprehensive method and two-phase method. The former one combines the relevance and the redundancy to get a comprehensive factor. The classical algorithms are such as CFS algorithm [5], the mRMR algorithm [6] based on the principle of minimal redundancy and the mRR algorithm based on maximal relevance and clustering technology. The latter algorithms consider the relevance and the redundancy respectively. The representative methods include FCBF algorithm [7] based on uncertainty measurement, the IAMB algorithm [8] based on Markov blanket and their improved methods. Generally speaking, the two-phase analysis method for redundancy and relevance have good quality in feature extraction than the comprehensive method.

2.2 Deep Learning Methods

The deep learning method is one of learning method in machine learning. The learning structure can be divided to shallow one and deep one. Support vector machine is one kind of shallow learning structure [9]. Deep structure proposed by Hinton for the first time in 2006. It is gradually well known by the representative ability in classification and recognition. The main methods in deep learning include restricted Boltzmann machine (RBM), deep belief networks (DBN), convolutional neural network (CNN) and auto encoder (AE). AE and DBN are unsupervised methods, while CNN is a deep supervised method. Then a series of deep network who get much process on the match of Imagenet dataset are appeared. The familiar methods are VGGNet [10], GoogleNet [11], Resnet [12] and DenseNet [13].

Though deep learning methods can learn a good representation for data automatically, it needs much time to train the model. At some times, it only extracts the global feature. Local feature may play a more import role in the specific task.

3 Feature Extraction Based on KL Measure

3.1 Feature Extraction

Let X be a matrix composed by n-dimension, and it can be obtained by normalizing the orthogonal vectors u_j:

$$X = \sum_{j=1}^{\infty} a_j u_j \tag{1}$$

X can be evaluated as:

$$\bar{X} = \sum_{j=1}^{N} a_j u_j \tag{2}$$

The mean root square error can be represented as:

$$\zeta = \mathrm{E}[(X - \bar{X})^T (X - \bar{X})] \tag{4}$$

According to the definition of the orthogonal vector, the following formulas all can be satisfied:

$$u_i^T u_i = \delta_{ij} = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \tag{5}$$

$$\zeta = \mathrm{E}[\sum_{j=N+1}^{\infty} a_j^2] \tag{6}$$

$$a_j = u_j^T X \tag{7}$$

After feeding the value of a_j to the Eq. (6), we can get:

$$\begin{aligned} \zeta &= \mathrm{E}[\sum_{j=N+1}^{\infty} u_j^T X \mathrm{X}^T u_j] \\ &= \sum_{j=N+1}^{\infty} u_j^T \mathrm{E}(X\mathrm{X}^T) u_j \end{aligned} \tag{8}$$

Let R be:

$$\zeta = \mathrm{E}[XX^T] \tag{9}$$

Then ζ can be represented by

$$\zeta = \mathrm{E}[\sum_{j=N+1}^{\infty} u_j^2 R u_j] \tag{10}$$

where R is auto-correlation matrix.

The goal is to minimize the goal ζ, so the value of u_j should be determined in advance. Therefore, we can construct the Lagrange formula by introducing Lagrange coefficient:

$$g(u_j) = \sum_{j=N+1}^{\infty} u_j^T R u_j - \sum_{j=N+1}^{\infty} \lambda(u_j^T u_j - 1) \tag{11}$$

Then we can get the result by differentiating Eq. (11) with u_j:

$$(R - \lambda_j \mathrm{I})u_j = 0, \quad j = N+1, \ldots, \infty \tag{12}$$

We can get the mean square error form the former N items:

$$\begin{aligned} \zeta &= \sum_{j=N+1}^{\infty} u_j^T R u_j \\ &= \sum_{j=N+1}^{\infty} tr[u_j R u_j^T] \\ &= \sum_{j=N+1}^{\infty} \lambda_j \end{aligned} \tag{13}$$

From the Eq. (13), we can conclude the feature value λ_j is smaller, the goal value ζ is smaller. They are positive relevance relation between them.

3.2 Feature Generation

The feature extraction process based on KL can be represented as:

(1) The autocorrelation R of X can be denoted as:

$$\begin{aligned} R &= \mathrm{E}[XX^T] \\ &\approx 1/N \sum_{j=1}^{N} X_j X_j^T \end{aligned} \tag{14}$$

(2) The diagnosing of matrix R can be denoted as

$$u_i^T R u_j = u_i^T \lambda_j u_j = \lambda_j \delta_{ij} \tag{15}$$

$$\mathrm{E}[U^T X X^T U] = U^T R U = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{pmatrix}$$

(3) The eigenvalues of the matrix are sorted as:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots \lambda_N \tag{16}$$

(4) The largest K eigenvalues are selected to get the eigenvector. The transform matrix U can be obtained by normalizing them.
(5) The primitive matrix X is transformed by K-L measure, the K-dimension matrix X^* can replace the primitive matrix X

$$X^* = U^T X \tag{17}$$

3.3 Loss Function Based on KL Divergence

KL divergence (Kullback-Leibler divergence) is also called KL distance. It is a tool that can describe the distinction between two distributions. The form of KL divergence has the following form:

$$D_{KL}(p||q) = \sum_{x \in \Omega} p(x) \log \frac{p(x)}{q(x)} \tag{18}$$

where p and q are two probability distribution in the probability space Ω.

$D_{KL}(p||q)$ is the KL divergence of p respective to distribution q. It is easy to see from Eq. (17) that $D_{KL}(p||q) \neq D_{KL}(q||p)$, namely, the KL divergence does not satisfy the symmetry. For the KL divergence as the form $D_{KL}(p||q)$, the distribution p is the true distribution and q is the approximate distribution. The value of $D_{KL}(p||q)$ is larger, the difference between real distribution and the approximate distribution is large.

According to the Jensen inequality formula, $D_{KL}(p||q) \geq 0$ is hold, where the equality equation is hold only when $p = q$.

After the features are extracted by (17), we can use it to apply the specific tasks and then compare it with the other methods by KL divergence. If the KL distance of the result is larger than the threshold, then the feature extraction is not good enough, the

feature extraction has to be optimized further. And the KL distance will be added to the Eq. (11), shown as:

$$g(u_j) = \sum_{j=N+1}^{\infty} u_j^T R u_j + D_{KL}(p||q) - \sum_{j=N+1}^{\infty} \lambda(u_j^T u_j - 1) \tag{19}$$

The final goal will be that KL distance is nearly 0, which is shown as:

$$D_{KL}(p||q) = 0 \tag{20}$$

4 Experiment Analysis

4.1 Dataset

In order to verify the proposed method, we compare with the other methods. The image dataset in Columbia university COIL-20, COIL-100, CBCI 3000 and USPclassifyALL.

(1) COIL-20 dataset includes 1400 images composed by 20 different goods. Every image has the 32 * 32 resolution.
(2) COIL-100 dataset is composed of 7200 images which attribute to different classifications. Every image is with the resolution 32 * 32.
(3) CBCL3000 dataset is organized by 3000 images, where every image with the resolution 19 * 19.
(4) USPSclassifyAll dataset is composed of 11000 images with the resolution 19 * 19.

The threshold of the KL divergence is set to 0.01 in our method.

4.2 Simulation Result

In order to verify the effectiveness of the proposed, we compare it with the classical method such as SVM, DMIFS, mRMR, VGG, DBN and our method. The first experiment is implemented in the first dataset COIL-20. The features are firstly generated by the five methods. Then, the obtained features are then used to recognize the images. The accuracies obtained with the changing of the number of the features for the five methods are shown as Fig. 1.

It is easy to see that our method has the best accuracy in the five methods. When using just 100-dimension feature, our method can get a accuracy about 93.1%, compared with the best accuracy value 79.2%. When the number of selected features reaches 300, the accuracy in our method does not change anymore. This means that our method can get a very high accuracy at the smallest number of features.

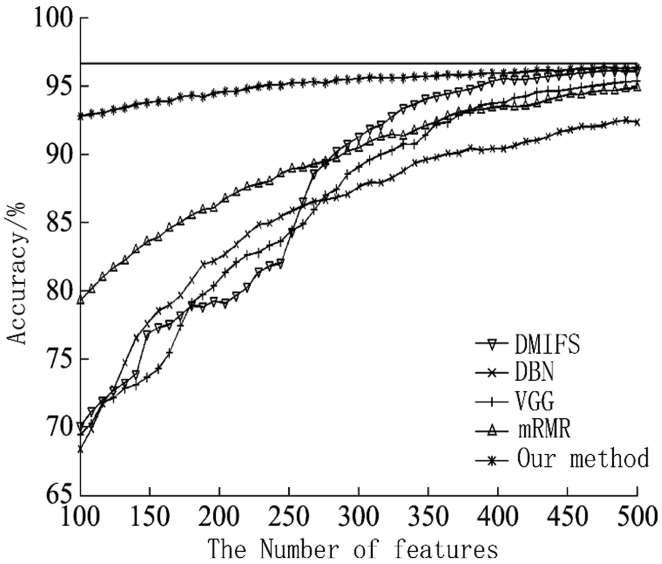

Fig. 1. Accuracy comparison for dataset COIL-20

COIL-100 have 5-times more images than COIL-20. Therefore, the five method are simulated again, and the result is shown in Fig. 2. As the same with the former experiment, our method has the best accuracy in the all five methods. The accuracy seems to be better in this experiment when the number of the features is small. Most of them can obtain an accuracy above 90% when the number of the feature is about 100.

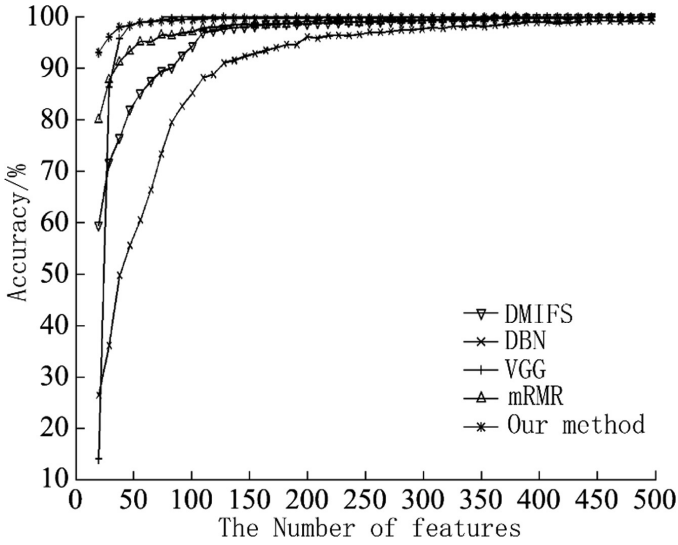

Fig. 2. Accuracy comparison for dataset COIL-100

The method with the poorest performance in accuracy is VGG. This may be caused by that VGG needs a lot of the samples to learn good feature representation. Better than the former experiment, our method obtained the accuracy 96% when the number of the samples is only 50.

The third experiment is implemented in CBCI 3000. The simulation results of five methods are shown in Fig. 3. What we can see is that all five methods have best accuracies in the three experiments. It is clearly to see that our method has the best accuracy when the number of the features is about 9, with the accuracy value of 90%. All the five methods can reach the accuracy 90% except from VGG, when the number of the features are about 10. As the former experiment, VGG perform poorest in the five all.

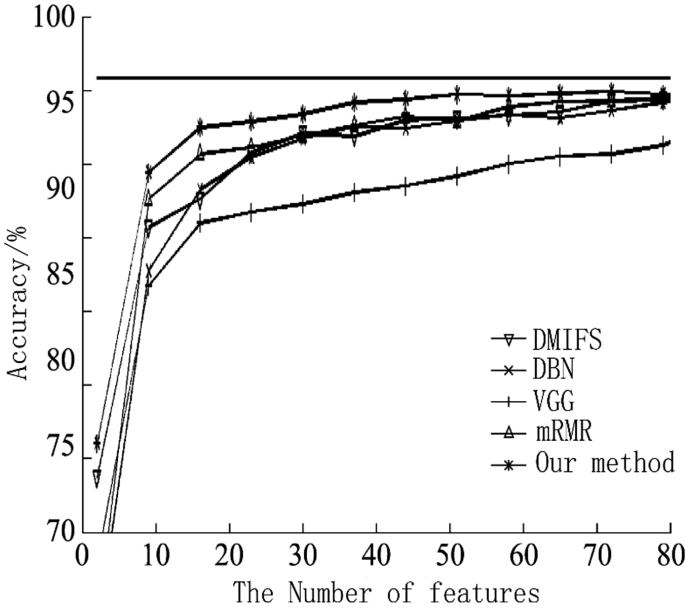

Fig. 3. Accuracy comparison for dataset CBCI 3000

The final experiment is simulated in the dataset USPSclassifyAll. Compared with the former three experiment, this experiment has most samples. Different from the former experiments, the five methods seem to have poorer performance in this experiment. Our method and mRMR can only obtain an accuracy at 80% when the number of the features is nearly 30. DMIFS, VGG and DBN have the accuracy 0 when the number of the samples is 38, 43 and 55. They can get an accuracy of 85% after the number of the samples reaches 80. This may be caused by that the number of the samples are far more than the former experiments. Therefore, it needs more features to represent the samples and then get a good learning model (Fig. 4).

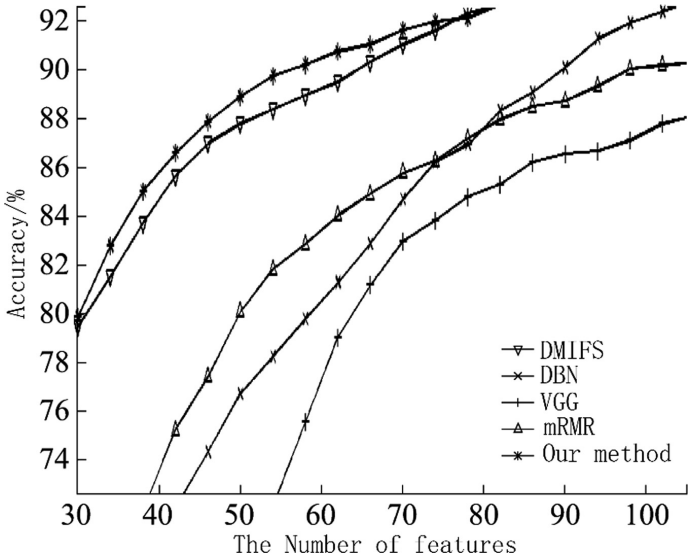

Fig. 4. Accuracy comparison for dataset COIL-100

The recall rate of the five methods for the dataset COIL-100 is also simulated, and the result is shown as Table 1.

Tab 1. Performance comparisons

Performance	Precision	Recall	F1
DMIFS	0.86	0.81	0.83
DBN	0.88	0.75	0.81
VGG	0.87	0.74	0.80
mRMR	0.90	0.78	0.84
Our method	0.92	0.81	0.86

From the Table 1, we can see our method has the best performances on precision rate, recall rate and F1.

5 Conclusion

To improve the feature extraction from the data, we propose a novel method that combines the KL divergence with feature extraction. The features are initially extracted from the primitive data by matrix decomposition. Then the features are optimized by using KL divergence. The KL divergence is introduced to the loss function to make the goal function has the shortest KL distance. The experiment is implemented in four

datasets. The result shows that our method has the best performance on accuracy compared with the other methods such as SVM, DMIFS, mRMR, VGG, DBN. Our method always can get a good accuracy when the number of the features is small, which proves that our method has the better ability in extracting better feature.

The next work will be applying our method to more practical applications with enormous samples.

Acknowledgments. This work was financially Project supported by the Education Department of Hai-nan Province, project number: hnjg2017ZD-17. The Hainan Provincial Department of Science and Technology under Grant No. ZDKJ201602.

References

1. Saidi, R., Aridhi, S., Nguifo, E.M., et al.: Feature extraction in protein sequences classification: a new stability measure. Med. Phys. **6**, 683–689 (2018)
2. Chu, Q.C., Wang, Q.G., Qiao, S.B., et al.: Feature analysis and primary causes of pre-flood season "cumulative effect" of torrential rain over South China. Theoret. Appl. Climatol. **131**, 91–100 (2018)
3. Robnik, S.M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach. Learn. **53**, 23–69 (2003)
4. Liu, H., Sun, J., Liu, L., et al.: Feature selection with dynamic mutual information. Pattern Recogn. **42**, 1330–1339 (2009)
5. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the 7th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, pp. 359–366 (2000)
6. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. In: Proceedings of the IEEE Computer Society Conference on Bioinformatics, pp. 523–528. IEEE Computer Society Press, Washington DC (2003)
7. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. **5**, 1205–1224 (2004)
8. Tsamardinos, I., Aliferis, C., Statnikov, A.: Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation. J. Mach. Learn. Res. **11**, 171–234 (2010)
9. Yang, L., Wen, K., Gao, Q., et al.: SVM based multi-label learning with missing labels for image annotation. Pattern Recogn. **78**, 307–317 (2018)
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
11. Szegedy, C., Liu, W., Jia, Y., et al.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
12. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
13. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
14. Han, Q.L., Liang, S., Zhang, H.L.: Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world. IEEE Netw. **29**(2), 40–45 (2015)
15. Song, Y., Wang, H., Li, J., Gao, H.: MapReduce for big data analysis: benefits, limitations and extensions. In: Che, W., et al. (eds.) ICYCSEE 2016. CCIS, vol. 623, pp. 453–457. Springer, Singapore (2016). https://doi.org/10.1007/978-981-10-2053-7_40