Balancing large margin nearest neighbours for imbalanced data

Xiaotian Zhang¹, Nan Han², Shaojie Qiao³, Yongqing Zhang⁴, Ping Huang⁵, Jing Peng⁵, Kai Zhou⁶.
Chang-an Yuan⁶

¹School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, People’s Republic of China
²School of Management, Chengdu University of Information Technology, Chengdu 610103, People’s Republic of China
³School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, People’s Republic of China
⁴School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, People’s Republic of China
⁵Sichuan Provincial Department of Public Security, Chengdu 610014, People’s Republic of China
⁶School of Computer and Information Engineering, Nanning Normal University, Nanning 530299, People’s Republic of China

E-mail: hannan@cuti.edu.cn

Abstract: It is critical to learn and obtain a good distance metric that can precisely measure the distance between samples in imbalanced data. However, traditional metric learning algorithms, e.g. large margin nearest neighbour (LMNN), information-theoretic metric learning, neighbourhood component analysis, do not take imbalanced distributions of classes into consideration. The traditional methods are apt to be affected by the majority samples, so those important minority samples are often ignored during the learning phase of distance metrics matrix, this may greatly confuse decision-making systems on classifying samples. In order to resolve this problem, the authors propose a novel metric-learning method named balancing large margin nearest neighbour (BLMNN) for imbalanced data. BLMNN can improve the objective function according to the distribution of classes, which treats the minority and majority classes equally during the optimisation process. Thus, the contribution of minority class is taken into full consideration, which can greatly improve the accuracy of classification. Substantial experiments were performed on real-world imbalanced datasets. The experiments result in various evaluation indexes of the proposed method comparing it with other metric-learning methods show the advantages of the proposed method.

1 Introduction

Most datasets in real life are imbalanced, which means the number of samples from one class heavily exceeds the others, and most classification algorithms assume that the data is balanced. The performance may drop significantly when imbalanced distribution occurs in datasets. In general, these algorithms have a tendency to favour the samples belong to majority classes and ignore the samples belong to minority classes in the phase of training. Especially those real-world applications that are designed to recognise infrequent but crucial instances, such as network intrusion detection [1], fraud detection [2] and disease detection [3]. Taking cancer detection as an example, people who have cancer only hold a tiny percentage in the world. Classifying a patient with cancer to a normal person will cause precarious consequences for this patient, which could affect the treatment and put their lives in danger. Therefore, accurately measuring the similarities between samples from imbalanced datasets is very important.

In the past decades, various metric-learning methods have been proposed [4–9]. Also distance metric learning has attracted much attention in various machine learning tasks because of its extensive applications [10–13].

However, conventional metric-learning methods do not have the ability to learn proper distance metrics which can precisely capture the similarities of samples in imbalanced datasets. This is because these metric-learning methods focus more on the distances between samples while do not take into full consideration the distribution of imbalanced classes. This will severely affect the accuracy of measuring the similarities between samples in imbalanced datasets.

2 Related work

Metric learning is used to learn a better distance measurement via training. To achieve this goal, it tries to minimise the distance between similar samples and maximise the distance of dissimilar samples. Weinberger and Saul [5] proposed a distance metric of large margin nearest neighbour (called LMNN). It tries to learn a distance metric on the basis of k-nearest neighbours (kNN) classification. In the learning process, it tries to keep k-nearest similar neighbours in the same class while keeping dissimilar samples separated by a large margin. Furthermore, Weinberger and Tesauro [6] proposed a metric-learning method for kernel regression (called MLKR). Xing et al. [7] proposed a convex objective function which aims to learn a distance metric for clustering (called MMC) with side-information by minimising the distances between all similarly labelled input data while maximising the distances between the differently labelled input data. Davis et al. [8] proposed a method called information-theoretic metric learning (ITML). It tries to learn a distance metric by combining metric learning with information theory. Goldberger et al. [9] proposed a supervised learning algorithm known as neighbourhood component analysis (called NCA). It aims to minimise the leave-one-out classification error from a stochastic variant of KNN classification which uses a Mahalanobis distance metric.

Recently, some deep metric-learning methods are proposed for classification with the development of deep learning. Yu et al. [13] proposed a deep multimodal distance metric-learning method, which combines auto-encoder and metric learning to obtain optimal weight. Song et al. [14] proposed a structured objective function for deep metric learning. Wang et al. [15] proposed a metric-learning method combining deep learning and regression method.

The aforementioned approaches do not take into account handling the imbalanced data. However, in reality, the distributions of data are often imbalanced. The study aims to handle the real-world problems beyond imbalanced data.
3 Problem definition

In this section, we will introduce the important concepts relating to metric learning and the LMNN algorithm which is the basis of the proposed method.

3.1 Problem statement

Assuming there are a set of samples in the amount of $N$ in a $d$-dimensional space $X = \{x_1, x_2, ..., x_N\} \subset \mathbb{R}^d$, along with labels $Y = \{Y_1, Y_2, ..., Y_N\}, y_i \in Y, 1 \leq i \leq N$ belongs to the class set $C = \{1, 2, ..., c\}$ where $c$ is the total number of classes in a dataset. For each sample $x_i$ and $x_j$, we can calculate the Mahalanobis distance between these two samples by the following equation:

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)}$$

(1)

where $M \in \mathbb{R}^{d \times d}$ represents a distance metric matrix, $d_M(x_i, x_j)$ is the Mahalanobis distance between $x_i$ and $x_j$ based on the distance metric matrix $M$.

What metric learning actually do is to design an algorithm to learn a Mahalanobis distance matrix $M$. Also we can accurately measure the distance between samples by (1) with this matrix $M$. For any distance matrix $M$, it should satisfy the following properties:

(i) $d_M(x_i, x_j) + d_M(x_j, x_k) \geq d_M(x_i, x_k)$ (triangular inequality)
(ii) $d_M(x_i, x_j) \geq 0$ (non-negativity)
(iii) $d_M(x_i, x_j) = d_M(x_j, x_i)$ (symmetry)
(iv) $d_M(x_i, x_j) = 0 \leftrightarrow x_j = x_i$ (distinguishability)

It is straightforward that $M$ is positive semi-definite. Therefore, the matrix $M$ could be decomposed into $M = LL^T$. $L \in \mathbb{R}^{N \times d}, 1 \leq d$. It is identical for a distance metric to find a mapping function to transform samples from original feature space which unable to accurately represent similarities between samples to a new feature space which able to. Moreover the transformed samples would be calculated by $x'_i = L \times x_i$. The Mahalanobis distance is defined as follows:

$$d_M(x_i, x_j) = \sqrt{\| L(x_i - x_j) \|^2_{\xi}}$$

(2)

The distance $d_M(x_i, x_j)$ equals the Euclidean distance between the projected sample vectors in new feature space $x'_i = L \times x_i$ and $x'_j = L \times x_j$.

In summary, we aim to find an appropriate transformation such that the distance between similar samples is small while the distance between dissimilar samples is relatively large. We can intuitively understand this process by Fig. 1.

3.2 Large margin nearest neighbour

LMNN trains a distance metric with the goal to minimise the distance between the input samples and their similar $k$-nearest neighbours and dissimilar samples are separated by a large margin. On the basis of hinge loss, LMNN formulates a convex optimisation by using the large margin criterion similar to support vector machines (SVM) [16]. The objective function of LMNN is defined as follows:

$$\min_{M} \left(1 - \mu \right) \sum_{i,j \rightarrow i} (x_i - x_j)^T M (x_i - x_j) + \mu \sum_{i,j \rightarrow i,l} \left(1 - y_{ij}\xi_{ij}\right)$$

s. t. $\left(x_i - x_i\xi_M (x_i - x_i) - (x_i - x_j)^T M (x_i - x_j) \geq 1 - \xi_{ij}\right)$

(3)

where $\mu$ is a hyperparameter which controls the weight between the two terms in the objective function of LMNN. $j \rightarrow i$ indicates that $x_j$ is the target neighbour of input $x_i$. Importantly, $y_{ij} = 1$ indicates that $x_i$ and $x_j$ belong to the same class, and $y_{ij} = 0$ otherwise. $\xi_{ij}$ is a slack variable which monitors the margin violation. $M \succeq 0$ indicates that the matrix $M$ is semi-definite.

4 Balancing large margin nearest neighbour

For the purpose of training a proper distance metric which can deal with imbalanced data, we proposed an improved metric-learning method called BLMNN.

4.1 Construction of loss function

To balance the importance of samples in each class, a weight variable is introduced which can be computed by the following equation:

$$\eta_i = \frac{N}{n_i}$$

(4)

where $\eta_i$ is the weight of sample $x_i$, $N$ indicates the total number of samples belong to the majority class, $n_i$ indicates the total number of samples belong to the same class with $x_i$.

There are two terms in the loss function of BLMNN in (3), one of which tries to pull target neighbours closer together, and another one tries to push samples from different classes further apart as we can see in Fig. 2.

The first term in loss function will minimise the distances between each input sample and its target neighbours. The target neighbours of $x_i$ are those that should be closest to $x_i$, which are $KNN$ that belong to the same class with $x_i$. After the transformation $L$ of the original feature space along with the consideration of the weight of each sample the sum of these squared distances can be formulated as follows:

$$\epsilon_{pull}(L) = \sum_{i,j \rightarrow i} \eta_i \| L(x_i - x_j) \|_2$$

(5)

where $x_i$ indicates the input sample, $x_j$ indicates the target neighbour of $x_i$, $\eta_i$ indicates the weight of class which $x_i$ belongs to.

The second term penalises the occurrences that samples from the different classes are close to each other. The goal is to put a large margin between impostors and perimeters established by target neighbours. An impostor is any $x_i$ with label $y_j \neq y_i$ such that the following inequality does hold:
$$\| L(x_i - x_l) \|^2 \leq \| L(x_i - x_j) \|^2 + 1$$

(6)

The second term can be described as follows:

$$\varepsilon_{\text{push}}(L) = \sum_{i \neq j} \sum_{l} (1 - y_{il}) \eta_l \left[ 1 + \| L(x_i - x_j) \|^2 - \| L(x_i - x_l) \|^2 \right].$$

(7)

where $y_{jl} = 0$ indicates that $x_i$ and $x_l$ belong to different class, and $y_{jl} = 1$ otherwise. [$a_i$] = $\max(a, 0)$. It is worthwhile to note that this hinge loss will make no contribution to the overall loss function if (7) does not hold, i.e. the input $x_l$ keeps a relatively large distance away from $x_i$.

These two terms could have competing effects. A weighing parameter $\mu \in [0, 1]$ balances these goals. So the final combined loss function is defined as follows:

$$\varepsilon(L) = (1 - \mu)\varepsilon_{\text{pull}}(L) + \mu\varepsilon_{\text{push}}(L)$$

(8)

### 4.2 Optimisation

To minimise the loss function in (8), let $M = L^TL$, then substitute the squared Mahalanobis distance $d_{\text{M}}^2(x_i, x_j)$ (defined in (11)) into (8), we can obtain the loss function:

$$\varepsilon(M) = (1 - \mu) \sum_{i \neq j} \eta \cdot d_{\text{M}}^2(x_i, x_j) + \mu \sum_{i \neq l, l} (1 - y_{il}) \eta_l \left[ 1 + d_{\text{M}}^2(x_i, x_j) - d_{\text{M}}^2(x_i, x_l) \right].$$

(9)

With this substitution, the loss function is now expressed over a positive semi-definite matrix $M \succeq 0$. Also $M \succeq 0$ must be added to the phase of optimisation to ensure that we learn a well-defined distance metric. To reform the optimisation process of (9) as a case of semi-definite programming (SDP), a slack variable $\xi_{ijl}$ is introduced. This variable $\xi_{ijl} \geq 0$ is used to measure the number of sample triplets violating the inequality in (6). Eventually, we can obtain the optimisation as

$$\min_M \left( 1 - \mu \right) \sum_i \eta_i (x_i - x_j)^T M (x_i - x_j) + \mu \sum_{i \neq l, i, l} (1 - y_{il}) \eta_l \xi_{ijl}$$

s.t. $(x_i - x_j)^T M (x_i - x_j) - (x_i - x_l)^T M (x_i - x_l) \geq 1 - \xi_{ijl}$

$$\xi_{ijl} \geq 0$$

$$M \succeq 0$$

(10)

It is worth to point out that slack variables $\xi_{ijl}$ are very sparse because most input $x_l$ are distant to $x_i$ relative to its target neighbour $x_j$. Such case will not trigger a positive loss. Also, we can use a standard SDP solver package to optimise the objective function in (10).

### 5 Experiments

To prove the performance of BLMNN, we performed substantial experiments on various benchmark datasets by comparing our method with five representative methods including LMNN [5], MLKR [6], MMC [7], ITML [8] and NCA [9].

### 5.1 Datasets

There are ten datasets from UCI Machine Learning Repository utilised in experiments, including Wine, Ecoli, Spectrometer, car_eva_34, us_crime, CTG, scene, Abalone, Coil_2000 and Letter_img. Table 1 presents some information on these datasets in detail. Target Class represents the number of minority classes or the label of class (e.g. imU). Ratio represents the ratio of the number of negative samples to the positive samples.

### 5.2 Classification performance on imbalanced datasets

Most metric-learning methods try to increase the accuracy of $k$NN classification, therefore we choose $k$NN to be the classifier for all methods, and set $k$ to 3. For LMNN and BLMNN, we set the regularised parameter $\mu$ to 0.5. For ITML, we set the parameter $\gamma$ for slack variables to 1. For all these comparing metric learning algorithms, we set the initial metric matrix to be an identity matrix whose diagonal elements are 1. For all the datasets, we normalise each feature column.

For those high dimensional datasets (like Spectrometer, us_crime, scene and Coil_2000), we use the principal component analysis (PCA) approach to reduce the dimensionality of the input.

![Fig. 2](image_url) When $x_i$ is input sample, the rest $k$ (3 in this case) nearest similar samples (denoted with the same shape) are seen as target neighbours, we want to pull its target neighbours closer and push its impostors away.

### Table 1 Description of the datasets

| Datasets        | Size  | Target class | Dimensions | Ratio |
|-----------------|-------|--------------|------------|-------|
| Wine            | 178   | 2            | 13         | 1.5:1 |
| Ecoli           | 336   | imU          | 7          | 8.6:1 |
| Spectrometer    | 531   | > = 44       | 93         | 11:1  |
| car_eval_34     | 1728  | good, v good | 21         | 12:1  |
| us_crime        | 1994  | >0.65        | 100        | 12:1  |
| CTG             | 2126  | 2            | 21         | 6.2:1 |
| scene           | 2407  | >one label   | 294        | 13:1  |
| Abalone         | 4177  | 7            | 10         | 9.7:1 |
| Coil_2000       | 9822  | minority     | 85         | 16:1  |
| Letter_img      | 20,000| Z            | 16         | 26:1  |
samples to 30. Normalisation and PCA on data help shorten the computational time and get rid of the phenomena of over-fitting. Then, we conduct fivefold cross-validation on each dataset which means 4/5 of samples are used to train and 1/5 for testing. Five evaluation measurements including precision, recall, F1 score, geometric-mean and area under ROC curve (AUROC) are calculated to reveal the performance of six metric learning algorithms on imbalanced data. Experimental results are given by taking the average of ten executions.

There are three classes of the Wine dataset. We treat the second class as minority and the remaining are treated as majority. The imbalance ratio is 1.5:1. For the Ecoli dataset, there are eight classes, we take class ‘imU’ as positive class with 35 samples, and the rest are negative classes with 301 samples. The imbalance ratio is 8.6:1. Six metric-learning algorithms are used to train the distance metrics, then we use 3-NN with different distance metrics to evaluate the performance. The results are given in Fig. 3 and Table 2.

From Fig. 3 and Table 2, we can see that BLMNN outperforms other metric-learning methods on all evaluation measurements. Since BLMNN takes full consideration the imbalanced distributions of classes, while others do not.

For car_eval_34 dataset, the classes ‘good’ and ‘v good’ are assigned to be positive class. The imbalance ratio is 12:1. For CTG dataset, there are 2126 samples in total, we take the class ‘2’ as positive class and the imbalance ratio is 6.2:1. For the Abalone dataset, there are 29 classes and Class ‘7’ is assigned to be positive class. The imbalance ratio is 9.7:1. For the Letter_img dataset, the class ‘z’ is viewed as positive class. The imbalance ratio is 26:1. We perform fivefold cross-validation for ten times and take the average in each dataset. The results are given in Tables 3–6.

As we can see from Tables 3 to 6, BLMNN outperforms others in most cases. Since the imbalanced distribution of classes severely affects the construction of the distance metric in conventional metric-learning methods.

Furthermore, we conduct experiments on the Spectrometer, us_crime, scene and Coil_2000 datasets to evaluate the performances of these methods. To avoid over-fitting and shorten the computational time, we use PCA to reduce the dimension on these datasets. 3-NN is utilised after learning of all distance metrics. Five measurements are calculated in Tables 7–10. It is straightforward to see that BLMNN achieves the best performance in most cases. As BLMNN takes imbalanced data into full consideration. In addition, we find ITML is also a good distance metric when compared to other methods.
In this study, we proposed a novel metric-learning method named BLMNN. Different from conventional metric-learning methods, BLMNN is designed to handle imbalanced data, which takes into full consideration the distribution of imbalanced classes. Importantly, BLMNN improves the objective function according to the distribution of classes, which treats the minority and majority classes equally during the phase of optimisation. The experiment results conducted on imbalanced datasets show that BLMNN outperforms the other metric-learning methods. However, the transformation of samples in BLMNN is linear, which is

### Table 5 Experimental results on the Abalone dataset

| Methods | Precision | Recall | F1   | GM   | AUROC |
|---------|-----------|--------|------|------|-------|
| LMNN    | 0.2273    | 0.1282 | 0.1639 | 0.3499 | 0.6426 |
| MLKR    | 0.2564    | 0.1282 | 0.1709 | 0.3511 | 0.6535 |
| MMC     | 0.2708    | 0.1667 | 0.2063 | 0.3087 | 0.6356 |
| ITML    | 0.2609    | 0.1538 | 0.1935 | 0.3833 | 0.7083 |
| NCA     | 0.2524    | 0.1331 | 0.1741 | 0.3545 | 0.6638 |
| BLMNN   | 0.3636    | 0.2051 | 0.2623 | 0.4445 | 0.7208 |

### Table 6 Experimental results on the Letter_img dataset

| Methods | Precision | Recall | F1   | GM   | AUROC |
|---------|-----------|--------|------|------|-------|
| LMNN    | 0.7542    | 0.6054 | 0.6717 | 0.7752 | 0.8786 |
| MLKR    | 0.6764    | 0.323  | 0.4333 | 0.5615 | 0.8078 |
| MMC     | 0.0833    | 0.0338 | 0.0432 | 0.1726 | 0.5029 |
| ITML    | 0.7954    | 0.5476 | 0.6471 | 0.7376 | 0.8765 |
| NCA     | 0.5556    | 0.3061 | 0.3947 | 0.5507 | 0.7581 |
| BLMNN   | 0.7931    | 0.6259 | 0.6996 | 0.7886 | 0.8912 |

### Table 7 Experimental results on the Spectrometer dataset

| Methods | Precision | Recall | F1   | GM   | AUROC |
|---------|-----------|--------|------|------|-------|
| LMNN    | 0.8679    | 0.6711 | 0.7453 | 0.8064 | 0.8927 |
| MLKR    | 0.8571    | 0.5933 | 0.6852 | 0.7526 | 0.8734 |
| MMC     | 0.8571    | 0.6667 | 0.7500 | 0.8123 | 0.9353 |
| ITML    | 0.8884    | 0.6822 | 0.7636 | 0.8187 | 0.8923 |
| NCA     | 0.7402    | 0.4844 | 0.5621 | 0.6702 | 0.8387 |
| BLMNN   | 0.9417    | 0.7333 | 0.8213 | 0.8520 | 0.8953 |

### Table 8 Experimental results on the us_crime dataset

| Methods | Precision | Recall | F1   | GM   | AUROC |
|---------|-----------|--------|------|------|-------|
| LMNN    | 0.4939    | 0.2893 | 0.3611 | 0.5252 | 0.7676 |
| MLKR    | 0.3797    | 0.1444 | 0.2006 | 0.3623 | 0.6692 |
| MMC     | 0.4423    | 0.08   | 0.1329 | 0.2793 | 0.5767 |
| ITML    | 0.4695    | 0.2422 | 0.3155 | 0.4833 | 0.7497 |
| NCA     | 0.2908    | 0.1022 | 0.148  | 0.2986 | 0.6108 |
| BLMNN   | 0.5029    | 0.32   | 0.3885 | 0.555  | 0.7763 |

### Table 9 Experimental results on the Coil_2000 dataset

| Methods | Precision | Recall | F1   | GM   | AUROC |
|---------|-----------|--------|------|------|-------|
| LMNN    | 0.1748    | 0.0273 | 0.0469 | 0.1627 | 0.5573 |
| MLKR    | 0.162     | 0.0192 | 0.0342 | 0.1353 | 0.5269 |
| MMC     | 0.1615    | 0.033  | 0.0416 | 0.1592 | 0.5076 |
| ITML    | 0.1872    | 0.0393 | 0.0646 | 0.1935 | 0.5594 |
| NCA     | 0.134     | 0.0337 | 0.0537 | 0.1746 | 0.5262 |
| BLMNN   | 0.2207    | 0.0461 | 0.0759 | 0.2093 | 0.586 |

### Table 10 Experimental results on the scene dataset

| Methods | Precision | Recall | F1   | GM   | AUROC |
|---------|-----------|--------|------|------|-------|
| LMNN    | 0.1885    | 0.0565 | 0.0855 | 0.225  | 0.5633 |
| MLKR    | 0.2285    | 0.0791 | 0.1173 | 0.2778 | 0.627 |
| MMC     | 0.0833    | 0.0169 | 0.0282 | 0.1292 | 0.5325 |
| ITML    | 0.3134    | 0.1243 | 0.178  | 0.3475 | 0.6167 |
| NCA     | 0.0987    | 0.0282 | 0.0435 | 0.1642 | 0.507 |
| BLMNN   | 0.3378    | 0.1186 | 0.1744 | 0.3388 | 0.6569 |

6 Conclusions

In this study, we proposed a novel metric-learning method named BLMNN. Different from conventional metric-learning methods, BLMNN is designed to handle imbalanced data, which takes into full consideration the distribution of imbalanced classes. Importantly, BLMNN improves the objective function according to the distribution of classes, which treats the minority and majority classes equally during the phase of optimisation. The experiment results conducted on imbalanced datasets show that BLMNN outperforms the other metric-learning methods. However, the transformation of samples in BLMNN is linear, which is
challenging to train a proper distance metric matrix for complex datasets, and we will focus on coping with this problem in future.

7 Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 61772091, 61802035, 61562008, 61962006; the Sichuan Science and Technology Program under Grant Nos. 2018JY0448, 2019YFG0106, 2019YFS0067, 2020YJ0481, 2020YFS0466, 2020YJ0430, 2020JDR0164; the Innovative Research Team Construction Plan in Universities of Sichuan Province under Grant No. 18TD0027; the Natural Science Foundation of Guangxi under Grant No. 2018GXNSFA138005, 2016GXNSFAA380209; the Soft Science Foundation of Chengdu under Grant No. 2017-RK00-00053-ZF; the Scientific Research Foundation for Young Academic Leaders of Chengdu University of Information Technology under Grant No. J201701; the Scientific Research Foundation for Advanced Talents of Chengdu University of Information Technology under Grant Nos. KYZ201715, KYZ201750; the Project of Scintific Research and Technology Development in Guangxi under Grant Nos. AA18118047, AD18126015, AB16380272; Guangdong Key Laboratory Project under Grant No. 2017B030314073; the support by the BAGUI Scholar Program of Guangxi Zhuang Autonomous Region of China.

8 References

[1] Parsaei, M.R., Rostami, S.M., Javidan, R.: ‘A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset’, Int. J. Adv. Comput. Sci. Appl., 2016, 7, (6), pp. 20–25
[2] Dal Pozzolo, A., Boracchi, G., Caelen, O., et al.: ‘Credit card fraud detection: a realistic modeling and a novel learning strategy’, IEEE Trans. Neural Netw. Learning Syst., 2018, 29, (8), pp. 3784–3797
[3] Wang, J., Ding, H., Bidgoli, F.A., et al.: ‘Detecting cardiovascular disease from mammograms with deep learning’, IEEE Trans. Med. Imaging, 2017, 36, (5), pp. 1172–1181
[4] Sugiyama, M.: ‘Local fisher discriminant analysis for supervised dimensionality reduction’. Proc. of the 23rd Int. Conf. on Machine Learning, Pittsburgh, Pennsylvania, USA, 2006, pp. 905–912
[5] Weinberger, K.Q., Saul, L.K.: ‘Distance metric learning for large margin nearest neighbor classification’, J. Mach. Learn. Res., 2009, 10, pp. 207–244
[6] Weinberger, K.Q., Tesauro, G.: ‘Metric learning for kernel regression’. Artificial Intelligence and Statistics, San Juan, Puerto Rico, 2007, pp. 612–619
[7] Xing, E.P., Jordan, M.I., Russell, S.J., et al.: ‘Distance metric learning with application to clustering with side-information’. Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2002, pp. 521–528
[8] Davis, J.V., Kulis, B., Jain, P., et al.: ‘Information-theoretic metric learning’. Proc. of the 24th Int. Conf. on Machine Learning, Corvalis, Oregon, USA, 2007, pp. 209–216
[9] Goldberger, J., Hinton, G.E., Roweis, S.T., et al.: ‘Neighbourhood components analysis’. Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2005, pp. 513–520
[10] Cheng, G., Yang, C., Yao, X., et al.: ‘When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs’, IEEE Trans. Geosci. Remote Sens., 2018, 56, (5), pp. 2811–2821
[11] Liao, S., Hu, Y., Zhu, X., et al.: ‘Person re-identification by local maximal occurrence representation and metric learning’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 2197–2206
[12] Meyer, B.I., Harwood, B., Drummond, T.: ‘Deep metric learning and image classification with nearest neighbour Gaussian kernels’. Proc. of the 25th IEEE Int. Conf. on Image Processing, Athens, Greece, 2018, pp. 151–155
[13] Yu, J., Yang, X., Gao, F., et al.: ‘Deep multimodal distance metric learning using click constraints for image ranking’, IEEE Trans. Cybernet., 2017, 47, (12), pp. 4014–4024
[14] Oh Song, H., Xiang, Y., Jegelka, S., et al.: ‘Deep metric learning via lifted structured feature embedding’. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 4004–4012
[15] Wang, Q., Wan, J., Yuan, Y.: ‘Deep metric learning for crowdswedness regression’, IEEE Trans. Circuits Syst. Video Technol., 2018, 28, (10), pp. 2633–2643
[16] Suykens, J.A., Vandewalle, J.: ‘Least squares support vector machine classifiers’, Neural Process. Lett., 1999, 9, (3), pp. 293–300