Incremental Active Learning Method for Supervised ISOMAP

Guopeng Zhang, Rui Huang[✉], and Junli Chen

Shanghai University, Shanghai, China
huangr@shu.edu.cn

Abstract. The supervised Isomap (S-Isomap), a supervised version of Isomap, can achieve better recognition performance by considering the class label information. However, S-Isomap usually suffers from heavy computation burden especially when new data arrive and re-computation of distance matrix is needed. To address the limitation, an incremental strategy is applied to update the geodesic distances using the previous results. Meanwhile, informative samples are selected through active learning to exploit the added data information with less human labeling effort on incremental process. Therefore, an incremental active learning S-Isomap (IALSI) is proposed. Image retrieval and classification experiments are conducted to evaluate the performance of IALSI. The results show that the proposed method can improve the accuracy meanwhile reducing the running time greatly.

Keywords: Dimension reduction · Supervised Isomap (S-Isomap)
Manifold learning · Incremental learning · Active learning

1 Introduction

With the rapid growth of information technology, massive data sets, such as human gene distribution, human healthy report, climate patterns, commercial record, have been created. Dimension reduction (DR), as a key technique to extract the important formation from the high-dimensional data, has increasingly attracted considerable attentions and studies.

The major task of DR is to find a projection to map the high dimension data into a lower dimensional space according to some criterion. So far, many DR methods have been proposed, and they can be divided into linear and nonlinear groups. PCA and LDA [1] are most popular linear methods, which try to obtain linear projection matrices. In contrast to the linear DR methods, nonlinear approaches are able to discover the nonlinear structure data in real world. Most incremental DR methods are based on kernel function. For example, kernel principal component analysis (KPCA) [2] has been widely used to get an implicit nonlinear mapping. But the result may be worse once the inappropriate kernel function is chosen. Recently, some methods based on manifold learning, such as laplacian eigenmaps (LE) [3], locally linear embedding (LLE) [4] and Isomap [5] have drawn much concern. In particular, Isomap, which can

reveal the intrinsic geometric structure of manifold by preserving geodesic distance of all similarity pairs, has presented some encouraging results. Therefore, a series of Isomap methods have occurred, such as supervised Isomap (S-Isomap) [6], orthogonal constrained marginal Isomap (M-Isomap) [7] and multi-manifold discriminant Isomap (MMD-Isomap) [8]. Among them, S-Isomap has shown good performances in visualization, image retrieval and classification by modifying the distances between the pairwise points with the same and different class labels. Despite the good performances achieved by Isomap family, they suffer from heavy computation burden that is caused by calculating the geodesic distances of all point pairs. In particular, distance recalculation has to be done when new data arrive [9] and leads to more computational load.

Incremental learning, as a family of efficient of scalable machine learning methods, has been proposed to operate on a sequence of data instance. At each step, the incremental learning algorithms process an incoming example and update the learner through the instance. An incremental version of Isomap has been proposed, which can efficiently update the geodesic distances and re-estimate the eigenvectors using the previous computation results [10]. On the other hand, for supervised learning, the number of labeled samples is limited in practice due to cost and time. This issue has motivated the recent study of active learning [11], which queries only a subset of informative incoming instances to update the learner and aims to maximize learner performance using minimal human labeling effort at each iteration.

In order to cut down the computation amount of S-Isomap and use the most informative labeled samples, we present an incremental active learning S-Isomap, called by IALSI. The proposed method modifies the original S-Isomap to the incremental version and reduces the computational cost greatly. Furthermore, active learning is applied for high efficiency and scalability towards large-scale learning tasks.

The remainder of this paper is organized as follows: Sect. 2 describes our proposed incremental active learning S-Isomap. The experimental results are shown in Sect. 3. Finally, Sect. 4 gives the conclusion and future work.

2 Incremental Active Learning S-Isomap

In this section, we will elaborate our method. Before that, we first give a brief review of S-Isomap. Then we introduce the details of our method.

2.1 S-Isomap

Given a data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbf{R}^{r \times N}$ with N points, DR is to find a mapping function that maps these points to a new data set $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \mathbf{R}^{r' \times N}$ in a lower dimensional space $(r' \ll r)$. $d(\mathbf{x}_i, \mathbf{x}_j)$ denotes the Euclidean distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$.

S-Isomap is a supervised version of Isomap. Considering the class label information of samples, it defines the distance metric as follows:

$$\tilde{d}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \begin{cases} \sqrt{1 - \exp\left(-d^2\left(\mathbf{x}_i, \mathbf{x}_j\right)/\beta\right)} \text{ if } l\left(\mathbf{x}_i\right) = l\left(\mathbf{x}_j\right) \\ \sqrt{\exp\left(d^2\left(\mathbf{x}_i, \mathbf{x}_j\right)/\beta\right)} - \alpha \text{ if } l\left(\mathbf{x}_i\right) \neq l\left(\mathbf{x}_j\right) \end{cases} \tag{1}$$

where the parameters of α and β are used to control the range of $d\left(\mathbf{x}_i, \mathbf{x}_j\right)$. Actually, α is set to be a small positive value, and β is set to be the average Euclidean distance between all pairs of data points. $l(\mathbf{x}_i)$ is the class label of $\mathbf{x}_i$. After the modification of point distances, the same steps as Isomap are used to obtain the lower dimensional representations of samples. It can be summarized as follows:

1. Construct neighborhood graph $\mathbf{D}_G$. For every pair of data points, if $\tilde{d}\left(\mathbf{x}_i, \mathbf{x}_j\right)$. is smaller than the fixed us radius $\in$ or $\mathbf{x}_j \in \text{KNN}(\mathbf{x}_i)$ ($\text{KNN}(\mathbf{x}_i)$ means $\mathbf{x}_j$ is the K-nearest neighbors of $\mathbf{x}_i$), set edge with weight of $\tilde{d}\left(\mathbf{x}_i, \mathbf{x}_j\right)$.
2. Compute the shortest geodesic distances. Initial the distance $d_G\left(\mathbf{x}_i, \mathbf{x}_j\right) = \tilde{d}\left(\mathbf{x}_i, \mathbf{x}_j\right)$. if $\mathbf{x}_j$ and $\mathbf{x}_i$ are neighbors, otherwise, put $d_G\left(\mathbf{x}_i, \mathbf{x}_j\right) = \infty$. Estimate the geodesic distances between all pairs of points by computing their shortest path distances in $\mathbf{D}_G$ using Dijkstra's or Floyd's algorithm.
3. Construct r'-dimensional embedding. Define the optimization problem as follows:

$$\min_{\mathbf{Y}} \sum_{\mathbf{x}_i, \mathbf{x}_j} \left(d(\mathbf{y}_i, \mathbf{y}_j) - d_G(\mathbf{x}_i, \mathbf{x}_j)\right)^2 \tag{2}$$

Let $\mathbf{H} = \mathbf{I} - (1/N)\mathbf{e}\mathbf{e}^T$, where $\mathbf{I}$ is an $N \times N$ identity matrix, and $\mathbf{e}$ is the vector of all ones. Let $\mathbf{Q}$ is an $N \times N$ matrix with elements $\mathbf{Q}_{ij} = d_G^2\left(\mathbf{x}_i, \mathbf{x}_j\right)$. The lower-dimensional embedding $\mathbf{Y}$ is obtained as $\left[\sqrt{\lambda_1}\mathbf{v}_1, \sqrt{\lambda_2}\mathbf{v}_2, \ldots, \sqrt{\lambda_{r'}}\mathbf{v}_{r'}\right]^T$., where $\{\mathbf{v}_i\}_{i=1}^{r'}$ denotes the eigenvectors corresponding to the first r' leading eigenvalues of $\mathbf{R} = -\mathbf{HQH}/2$.

2.2 Proposed IALSI

In IALSI, incremental learning is applied to update the shortest distance matrix based on Floyd's algorithm en new training data arrive in each iteration. These new incoming samples are chosen by active learning according to a given criterion.

The main task of active learning is to select information and representative samples for classification. According to different criteria for sample selection, active learning methods can be roughly divided into two groups: committee-based approaches and uncertainty-based approaches. Committee-based approaches, based on committee of classifiers, select the data whose classification results have the greatest disagreements among the classifiers. While in the uncertainty-based approaches, the most uncertain samples determined by some uncertainty scheme are chosen and labeled.

In the proposed IALSI, the uncertainty of sample $\mathbf{x}_i$ are measured as follows:

$$w(\mathbf{x}_i) = \max_{1 \leq c \leq C} d(\mathbf{x}_i, \overline{\mathbf{x}}_c) - \min_{1 \leq c \leq C} d(\mathbf{x}_i, \overline{\mathbf{x}}_c), \tag{3}$$

where $\bar{\mathbf{x}}_c$ is the mean value of samples belonging to class c, and C is the number of classes. A smaller $w(\mathbf{x}_i)$ means that the distances between sample $\mathbf{x}_i$ and each class center are comparable and thus there is ambiguity to label $\mathbf{x}_i$ based on the distances. Therefore, $\mathbf{x}_i$ is more informative and to be labeled and added to the training set.

Suppose the available training data set is $\{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^{N}$, the unlabeled data set is $\{\mathbf{x}_j\}_{j=1}^{M}$. We will select m samples from the unlabeled set through active learning. The details of active learning algorithm are summarized in Table 1.

Table 1. Active learning

Input: Training data set $\{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^{N}$, unlabeled data set $\{\mathbf{x}_j\}_{j=1}^{M}$, m.
Output: $\{\mathbf{x}_j, l(\mathbf{x}_j)\}_{j=1}^{m}$.
    1. Compute $w(\mathbf{x}_i)$ $(i = 1, 2, \ldots, M)$ using Eq. (3).
    2. Rank $w(\mathbf{x}_i)$ in an ascend order.
  3. Choose first m smallest samples from $\{\mathbf{x}_j\}_{j=1}^{M}$ and assign them true class labels to get
    $\{\mathbf{x}_j, l(\mathbf{x}_j)\}_{j=1}^{m}$.

In the beginning of IALSI, distances between points in $\{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^{N}$. are calculated using Eq. (1), and the shortest distance matrix $\mathbf{D}_0$ is obtained. In the t th iteration, m samples selected by active learning algorithm are added to the training set as $\{\mathbf{x}_j, l(\mathbf{x}_j)\}_{j=1}^{m}$, and used to update the geodesic distances between points $(1 \leq t \leq T)$. The extended distance matrix is given as:

$$\mathbf{D}_t = \begin{bmatrix} \mathbf{D}_{t-1} & \mathbf{D}_E \\ \mathbf{D}_E^T & \mathbf{D}_A \end{bmatrix} \tag{4}$$

where $\mathbf{D}_A$ denotes the distance matrix among the new arriving data $\{\mathbf{x}_j, l(\mathbf{x}_j)\}_{j=1}^{m}$, $\mathbf{D}_E$ denotes the distances between $\{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^{N}$ and $\{\mathbf{x}_j, l(\mathbf{x}_j)\}_{j=1}^{m}$. Instead of re-computing the shortest distances for all points, only new data points are considered. For new point $\mathbf{x}_i$, the distance between it and point $\mathbf{x}_j \in \{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^{N+m}$. is initialized as $d_G(\mathbf{x}_i, \mathbf{x}_j) = \tilde{d}(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_j$ and $\mathbf{x}_i$ are neighbors, otherwise, put $d_G(\mathbf{x}_i, \mathbf{x}_j) = \infty$. Then Floyd's algorithm is adopted to dynamically update the shortest distance matrix in all-pairs data. Hence, for each value of $p = N+1, \cdots, N+m$. in turn, replace all entries $d_G(\mathbf{x}_i, \mathbf{x}_j)$ by $\min\{d_G(\mathbf{x}_i, \mathbf{x}_j), d_G(\mathbf{x}_i, \mathbf{x}_p) + d_G(\mathbf{x}_p, \mathbf{x}_j)\}$. After the iteration, the final matrix is denoted $\mathbf{D}_t$. At last, we use the classic MDS algorithm to construct r'-dimensional embedding. During the iterations, the k value in neighborhood graph changes allows

$$k(t) = k_0 + k_1 t \tag{5}$$

where k_0 is the initial value and k_1 acts as a factor controlling the change range. Table 2 lists the details of IALSI.

Table 2. IALSI

Input: $\{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^N$, unlabeled data set $\{\mathbf{x}_j\}_{j=1}^M$, m, k_0, k_1
Output: $\mathbf{Y}$
1. Initialization: $k := k_0$; compute $\mathbf{D}_0$.
2. Incremental active learning loop:
for $t = 1$ to T
Obtain $\{\mathbf{x}_j, l(\mathbf{x}_j)\}_{j=1}^m$ using algorithm in Table 1
$k := k(t)$ using Eq. (5)
Compute $\mathbf{D}_A$ and $\mathbf{D}_E$;
Construct $\mathbf{D}_t$ using Eq. (4);
$\{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^{N+m} := \{\mathbf{x}_i, l(\mathbf{x}_i)\}_{i=1}^N \cup \{\mathbf{x}_j, l(\mathbf{x}_j)\}_{j=1}^m$;
$\{\mathbf{x}_j\}_{j=1}^{M-m} := \{\mathbf{x}_j\}_{j=1}^M - \{\mathbf{x}_j\}_{j=1}^m$;
$N := N + m$; $M := M - m$;
$\mathbf{D}_{t-1} := \mathbf{D}_t$;
end for
3. Obtain $\mathbf{Y}$ using Eq. (2).

Remark on computational complexity. We focus on the space complexity of construction of distance matrix and neighborhood graph, as the steps is the major computing load in practice. The space complexity of original S-Isomap algorithm is $O(N^2)$ [8]. When the new data points are added, the space complexity of batch algorithm reaches to $O\left((N+m)^2\right)$. In IALSI, only the shortest distances involving new data need to be computed. Hence, the space complexity of IALSI is simply $O(2N \times m + m^2)$.

3   Experiments and Discussions

In this section, experiments are carried out to evaluate the performance of IALSI. Three publicly available image data sets, including Corel [12], UC Merced LULC [13], and MNIST [14], are used for image retrieval and classification. Two batch methods including original Isomap and S-Imap, two incremental methods including incremental S-Isomap (ISI) and IALSI are compared. It is noted that, in ISI, samples to be added into training set are randomly selected. All the algorithms are implemented in MATLAB and run on an Intel(R) core(TM) i5-3470 PC at 3.2 GHz with 12 GB RAM.

3.1   Data Sets and Experimental Setting

Two datasets, the UC Merced LULC and Corel, are used for image retrieval. The LULC data set, obtained from aerial imagery, consists of images from 20 classes, with a pixel resolution of 30 cm. Each class contains 100 images of size 256 by 256 pixels. The Corel data set is from the real-world photo from COREL Photo Gallery. It has 20 categories with 500 images per category. We extract basic color features and wavelet texture features to describe images [15]. The features include color histogram (32 dimensions), color moment (64 dimensions), color auto correlogram (6 dimensions), wavelet moment (40 dimensions) and Gabor transform where the number of scales was set 4 and orientation was set 6 (48 dimensions). All the features are concatenated into a long vector as an image feature and each image is represented by a 190-dimensional vector. The new feature is normalized to zero mean and unit variance. The MNIST handwritten digit database is used for classification. It has 70000 hand written digit images with size of 28×28 pixels. We randomly choose 4000 images from digits "0–9" for evaluation. Figure 1 gives some example images of the three data sets.

In our experiments, each dataset is randomly divided into three partitions: training set, candidate set and test set. Among them, the candidate set is for sample selection where the informative samples are selected and added into the training set in each iteration. For the LUCL, Corel and MNIST data sets, the numbers of training and test samples are 200 and 300, 500 and 500, 500 and 500, respectively. The rest acts as candidate samples. For supervised DR methods, a BP neural network is constructed to approximate the mapping from the high-dimensional space to the lower-dimensional space. In the reduced dimension space, L_2 distance metric is employed for image retrieval to measure the similarities between the query image in the test set and the training images. For classification task, KNN $(K = 1)$ classifier is used to predict their class labels for the test samples.

In our experiments, the dimension of data is reduced to the number of classes of each data set. The parameters of each method are carefully adjusted to get the best performance. In particular, for Isomap and S-Isomap, the k value in KNN is assigned to 25 for LULC, 80 for COREL, 60 for MNIST. For the incremental learning methods, k_0 is set 15 for LUCL, 60 for Corel and MNIST, k_1 is 30 for the three data sets. Besides, we set $m = 300$ for all datasets, $T = 3$ for LULC and $T = 6$ for others.

3.2   Experimental Results and Analysis

Table 3 summarizes the retrieval precisions of 4 different methods when top 5 to 25 (with 5 step intervals) images are retrieved. In addition, classification accuracies on MNIST are also listed. It can be seen: (1) supervised methods perform better than unsupervised Isomap; (2) more training samples help to improve the performance; (3) IALSI with active learning outperforms the other methods, as more informative samples are chosen.
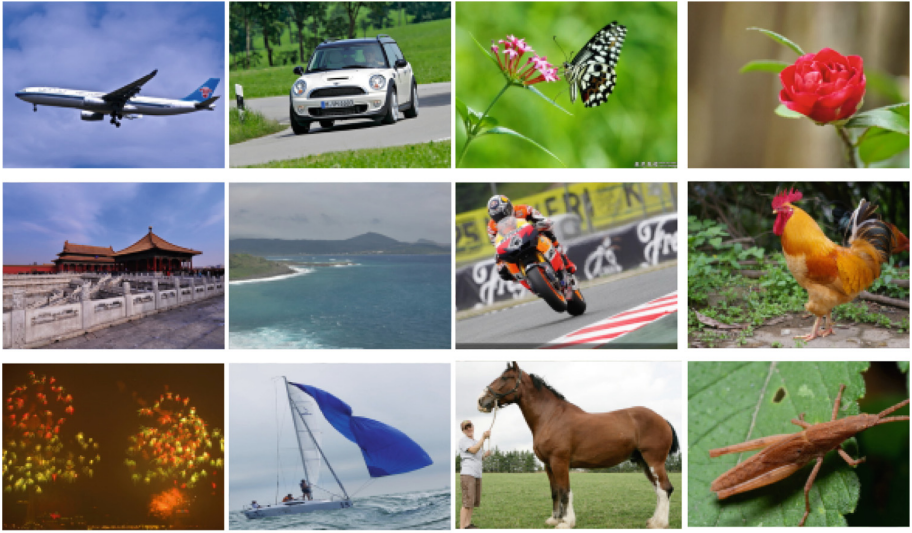
(a) Corel dataset

(b) LULC dataset

(c) MNIST dataset

Fig. 1. Example images of the three datasets used in our experiments.

Table 3. Performance comparison of different methods

|		Isomap	S-Isomap	ISI	IALSI
COREL	5	0.2335	0.3639	0.4367	**0.4628**
	10	0.2159	0.3553	0.4252	**0.4464**
	15	0.2036	0.3412	0.4154	**0.4335**
	20	0.1939	0.3400	0.4073	**0.4199**
	25	0.1852	0.3312	0.3996	**0.4043**
	AVE	0.2064	0.3463	0.4168	**0.4334**
LULC	5	0.1593	0.2680	0.4240	**0.4607**
	10	0.1333	0.2653	0.4037	**0.4374**
	15	0.1153	0.2238	0.3936	**0.4211**
	20	0.1055	0.2033	0.3863	**0.4077**
	25	0.0967	0.1777	0.3789	**0.3921**
	AVE	0.1220	0.2274	0.3973	**0.4238**
MNIST		0.7940	0.7660	0.8840	**0.8920**
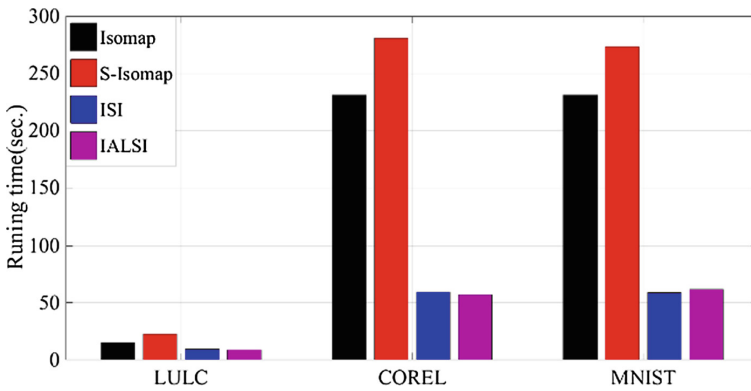
Fig. 2. Running time comparison of different methods

The running time of different methods are shown in Fig. 2. Compared with the batch methods, incremental learning methods are more efficient. Figure 3 takes the Corel data set as an example, and further depicts the running time during iterations. For the batch methods, re-computation of distance matrix is necessary when new samples are added and thus the computation amount rises dramatically while a smooth increase can be observed in the incremental methods.
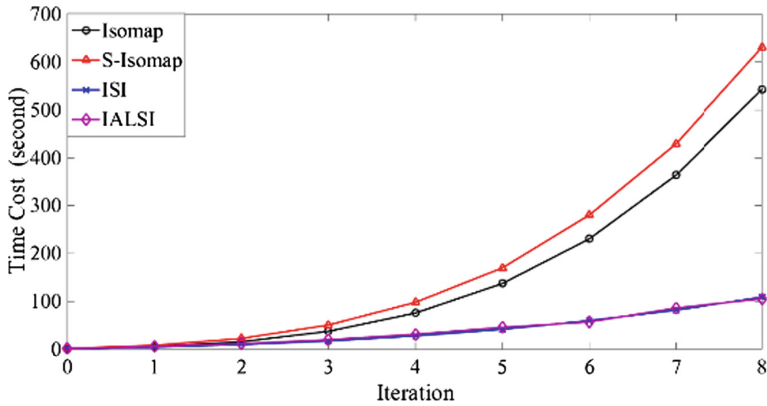
Fig. 3. Running time during iterations on Corel data set

4   Conclusions

An incremental active learning method for S-Isomap, IALSI, is proposed in the paper. IALSI aims to modify the original S-Isomap into incremental learning version meanwhile selecting informative samples through active learning in each iteration. The core idea of IALSI is to update the geodesic distance matrix based on the existing computation results and select informative samples according to an uncertainty criterion. The encouraging experimental results show that IALSI not only achieves better performance in image retrieval and classification tasks, but reduces the running time greatly.

References

1. Cai, D., He, X., Han, J.: Training linear discriminant analysis in linear time. In: 24th IEEE International Conference on Data Engineering, Cancun, pp. 209–217 (2008)
2. Zhang, D., Zhou, Z.H., Chen, S.: Adaptive kernel principal component analysis with unsupervised learning of kernels. In: 6th International Conference on Data Mining, Hong Kong, pp. 1178–1182 (2006)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv. Neural Inf. Process. Syst. **14**(6), 585–591 (2002)
4. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
5. Tenenbaum, J.B., De, S.V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
6. Geng, X., Zhan, D.C., Zhou, Z.H.: Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Trans. Syst. Man Cybern. B Cybern. **35**(6), 1098–1107 (2005)
7. Zhang, Z., Chow, T.W., Zhao, M.: M-Isomap: orthogonal constrained marginal Isomap for nonlinear dimensionality reduction. IEEE Trans. Syst. Man Cybern. B Cybern. **43**(1), 180–191 (2012)

8. Yang, B., Xiang, M., Zhang, Y.: Multi-manifold discriminant Isomap for visualization and classification. Pattern Recogn. **55**(11), 215–230 (2016)
9. Hoi, S.C.H., Wang, J., Zhao, P.: LIBOL: a library for online learning algorithms. J. Mach. Learn. Res. **15**(1), 495–499 (2014)
10. Law, M.H.C., Jain, A.K.: Incremental nonlinear dimensionality reduction by manifold learning. IEEE Trans. Patter Anal. Mach. Intell. **28**(28), 377–391 (2006)
11. Huang, S.J., Jin, R., Zhou, Z.H.: Active learning by querying informative and representative examples. In: Advance in Neural Information Processing System, pp. 892–900 (2010)
12. Hoi, S.C.H, Liu, W., Lyu, M.R., Ma, W.Y.: Learning distance metrics with contextual constraints for image retrieval. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2072–2078 (2006)
13. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, USA, pp. 270–279 (2010)
14. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
15. Xia, H., Hoi, S.C., Jin, R.: Online multiple kernel similarity learning for visual search. IEEE Trans. Pattern Anal. Mach. Intell. **36**(3), 536–549 (2013)