Scalable Similarity Learning using Large Margin Neighborhood Embedding

Zhaowen Wang†‡, Jianchao Yang†, Zhe Lin†, Jonathan Brandt†, Shiyu Chang†, Thomas Huang†
†Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL
‡Adobe Research, San Jose, CA
†{wang308, chang87, huang}@ifp.uiuc.edu ‡{jiayang, zlin, jbrandt}@adobe.com

Abstract

Classifying large-scale image data into object categories is an important problem that has received increasing research attention. Given the huge amount of data, non-parametric approaches such as nearest neighbor classifiers have shown promising results, especially when they are underpinned by a learned distance or similarity measurement. Although metric learning has been well studied in the past decades, most existing algorithms are impractical to handle large-scale data sets. In this paper, we present an image similarity learning method that can scale well in both the number of images and the dimensionality of image descriptors. To this end, similarity comparison is restricted to each sample’s local neighbors and a discriminative similarity measure is induced from large margin neighborhood embedding. We also exploit the ensemble of projections so that high-dimensional features can be processed in a set of lower-dimensional subspaces in parallel. The efficiency and scalability of our proposed model are validated on several data sets with scales varying from tens of thousands to one million images.

1. Introduction

With the ever-increasing number of digital images generated and uploaded to the Internet, automatic categorization of large-scale image sets with diversified contents has become a popular research topic. As the vast amount of image samples populate the data space more densely, we can now afford classification models with higher capacities to capture the underlying data distribution. Non-parametric methods, which infer the label information of test images from similar database images, have demonstrated promising results on large data sets in many vision tasks. For example, the k-Nearest Neighbor (kNN) classifier [15] has been successfully applied on the classification of the ImageNet Challenge data [8].

A good distance or similarity measure is crucial to the performance of any non-parametric model. As most image feature descriptors have very high dimensions and mainly characterize the low level visual information, measuring their distance directly in the Euclidean space yields unsatisfactory results. In order to close the semantic gap, people have used supervised information [25] to learn distance metrics with promoted semantic similarity, famous examples include the large margin nearest neighbor (LMNN) [23]. The supervision is usually provided in the form of comparative constraints over image pairs or triplets, leading to a time complexity which grows polynomially with sample size. In addition, for large-scale data where multimodal distributions are usually observed, a single distance metric is insufficient to correctly measure the similarities between all image pairs throughout the space. To ameliorate the problem, multiple metrics have been applied to different parts of the data space, by either assigning a distinct metric to each discrete space partition or learning an adaptive metric parameterized based on the location of test sample [16, 20, 12]. However, the extra model complexity of local metrics makes them less suitable for large-scale applications.

Besides the size of data sets, the high dimensionality of image descriptors is another factor limiting the scalability of existing metric learning algorithms. The Mahalanobis distance, one of the most popular forms of distance metric, requires computation quadratic to data dimension in calculation and cubic to dimension in its learning when positive-semidefinite constraint is placed. Low-rank regularization can be imposed on the Mahalanobis metric to save computation, but this may result in a non-convex optimization problem [15] and cause performance loss in some cases.

In this paper, we propose a new similarity learning algorithm that features good scalability with respect to both sample size and dimensionality. First, motivated by the findings from manifold learning with neighborhood embedding [27], we restrict similarity comparison to sample pairs within the same local neighborhood, and try to capture the discriminative structure of local data manifold using large margin neighborhood embedding (Sec. 2). In this way, we can not only save a great amount of computation in
training and testing, but also gain robustness to outliers by focusing only on more relevant samples. On the other hand, we project the original high-dimensional data to a set of lower-dimensional subspaces, and use the ensemble of similarities learned from these subspaces as a surrogate to the similarity in original space (Sec. 3). The similarity for each subspace can be evaluated and optimized in parallel, which offers superior scalability to data dimension. The proposed method is validated on several image classification benchmarks with varying scales (Sec. 4), and both of its accuracy and efficiency are shown to scale up gracefully from tens of thousands to one million images. Conclusions are drawn in final (Sec. 5). In short, the main contributions of this paper are:

- a neighborhood embedding based similarity learning algorithm with improved classification performance and better scalability to the number of training samples;
- an ensemble of distributed similarities learning algorithm which scales well to data dimensionality.

2. Similarity Learning using Neighborhood Embedding

Graph embedding [27] is a family of dimensionality reduction algorithms which map data points from a manifold in high-dimensional space to low-dimensional space while preserving the intrinsic data structure represented by a weighted graph. In many cases such as Locally Linear Embedding (LLE) [18] and Laplacian eigenmap [1], a sparsely-connected graph is constructed based on neighborhood relationship, which we refer to as neighborhood embedding.

Since learning the transform of dimensionality reduction can be regarded as a special case of distance or similarity learning, it is easy to extend the concept of neighborhood embedding to similarity learning. Given a set of $N$ data samples $\{x_i\}_{i=1}^N$ and the associated class labels $\{y_i\}_{i=1}^N$, we define $N_i$ as the index set for $x_i$’s $k$ nearest neighbors (in Euclidean distance). $N_i$ can be divided into two mutually exclusive subsets $N_i^+$ and $N_i^-$, which denote the indices of $x_i$’s neighbors with and not with label $y_i$, respectively. An adjacency graph can be built in which each $x_i$ is a vertex, and there is an undirected edge with weight $wij$ linking $x_i$ and $x_j$ if $i \in N_j$ or $j \in N_i$. Generally, neighborhood embedding tries to find an optimal transform $f$ for all the data samples by minimizing the following loss function:

$$ \mathcal{L} = \sum_{i \in N_j \land j \in N_i} w_{ij} d(f(x_i), f(x_j)), $$

(1)

where $f$ is usually a linear transform and $d(\cdot, \cdot)$ is the Euclidean distance measure in the transformed space. For our purpose, we are interested in a good similarity function $s(x_i, x_j)$ defined as a mapping from a pair of samples $(x_i, x_j)$ to a real number that quantifies their semantic similarity. Here we adopt a bilinear similarity function parameterized by a matrix $M$:

$$ s_M(x_i, x_j) = x_i^T M x_j, $$

(2)

and $M$ is a symmetric matrix usually with a low rank constraint. Bilinear similarity is commonly used in place of distance metric for its compactness [6, 7]. Its performance in many recognition tasks is found to be similar as the Mahalanobis distance, which has better theoretical properties but requires a positive semidefinite parameter matrix. With the neighborhood embedding formulation, the objective for similarity learning can be cast as

$$ \min_M \mathcal{L}(M) = \sum_{i \in N_j \land j \in N_i} w_{ij} s_M(x_i, x_j). $$

(3)

We still need to define the weights $w_{ij}$, which encode the class label information. Binary values $\pm 1$ are commonly used to assign $w_{ij}$ based on whether $x_i$ and $x_j$ come from the same class or not, which essentially has the same effect as partitioning the adjacency graph into a within-class graph and a between-class graph [27, 5] to impose a pairwise constraint that the similarity should be high between samples from the same class and low otherwise. For kNN classifiers, as suggested in [23], we care more about the relative similarity defined over a triplet of samples; i.e., a higher similarity score should be assigned when a sample $x_i$ is compared with any of its target neighbor $x_j, j \in N_i^+ \lor i \in N_i^-$, than with any of its imposter neighbor $x_i, l \in N_i^- \lor i \in N_i^-$. To this end, we define the graph weights as

$$ w_{ij} = \begin{cases} 
-|\{l | l \in N_i^- \land i \in N_l^+\}|, & j \in N_i^+ \land i \in N_l^+ \\
+|\{l | l \in N_i^+ \land i \in N_l^-\}|, & j \in N_i^- \land i \in N_l^- \\
0, & \text{otherwise}
\end{cases} $$

(4)

where $|A|$ denotes the cardinality of set $A$. With the weights in (4), we can organize our objective function $\mathcal{L}(M)$ into a more interpretable form:

$$ \sum_i \sum_{j \in N_i^+ \land i \in N_j^+} \sum_{l \in N_i^- \land i \in N_l^-} -s_M(x_i, x_j) + s_M(x_i, x_l), $$

which enforces relative similarity constraint for each triplet $(x_i, x_j, x_l)$ from the same neighborhood. To apply more penalty to those triplets violating the constraint, we use a hinge function to promote large margin of relative similarity difference, leading to the final form of our objective:

$$ \min_M \mathcal{L}(M) = \sum_i \sum_{j \in N_i^+ \land i \in N_j^+} \sum_{l \in N_i^- \land i \in N_l^-} [b - s_M(x_i, x_j) + s_M(x_i, x_l)]_+, $$

(5)
where $[\cdot]_+ = \max(\cdot, 0)$ is the hinge loss function, and $b > 0$ is the minimum required margin by which $x_i$ should be more similar to a target neighbor $x_j$ than to an impostor neighbor $x_l$ as measured by $s_M(\cdot, \cdot)$.

2.1. Learning Algorithm

Optimal linear transforms for neighborhood embedding can be found by solving a generalized eigenvector decomposition problem using graph Laplacian. However, this approach is not applicable for large-scale data and the nonlinear objective in Eq. (5) due to the computational complexity. Instead, we use an online learning method based on stochastic gradient descent [4], which is similar to [15, 6]. Specifically, we iteratively go through the whole training set and randomly sample triplet $(x_i, x_j, x_l)$ which contributes non-zero cost to the objective in (5). The sub-gradient of the objective evaluated at the current triplet is then used to update the parameter $M$. The update is performed iteratively with a diminishing step size and terminates upon convergence.

There can be a huge number of triplets to be considered in the objective function (5), even though the similarity comparison is restricted to local neighbors. Thus, a good sampling strategy to generate candidate triplets is essential to the speed of convergence on large data sets. For each training sample $x_i$, we search for its target neighbor $x_j$ and impostor neighbor $x_l$ according to

$$\{j, l\} = \arg\max_{j', l'} s_M(x_i, x_{j'}) - s_M(x_i, x_{l'}), \quad s.t. \ j' \in N_i^+, \ l' \in N_i^- \quad \text{or} \quad j' \in N_i^-, \ l' \in N_i^+. \quad (6)$$

Optimizing (6) returns $x_i$’s most dissimilar target neighbor $x_j$ and most similar impostor neighbor $x_l$, so that the triplet violates the relative similarity constraint the most. This can be solved efficiently with $|N_i|$ similarity comparisons.

The overall procedure for Similarity Learning with Neighborhood Embedding (SL-NE) is summarized in Algorithm 1.

2.2. Relation to LMNN

The SL-NE introduced above has a similar objective function as the well-known LMNN method [23], which is also based on triplet relative constraint. The key difference is that in LMNN the $k$ nearest target samples of $x_i$ are required to be more similar to $x_i$ than all the impostor samples in the global space, which is an overly restrictive constraint that even exceeds the condition for correct prediction by kNN classifier. Such strong global constraints often conflict with each other for high-dimensional data with multi-modal distribution, and makes the learning result ineffective and more vulnerable to outliers. It is proved that focusing on local constraints can improve the generalization capability of metric learning algorithms [2]. From another perspective, it is also possible to formulate LMNN using the graph embedding framework as in Eq. (3), and the associated adjacency graph will be densely connected due to many non-zero weights $w_{ij}$, a stark contrast to our sparse graph whose edge connections are restricted inside local neighborhoods. It is widely concurred that sparse graphs are typically superior to or more robust than dense graphs [30].

More concretely, the advantage of SL-NE over LMNN can be illustrated through an example shown in Fig. 1 and the corresponding accuracy comparisons listed in Table 1. LMNN can discriminate well between two coarse-grained classes “bird” and “boat”, but fails to learn the subtle differences between fine-grained categories, i.e. “flamingo” vs “ibis” or “ketch” vs “schooner”. This is because a large part of LMNN’s effort is wasted on optimizing unimportant similarity constraints (e.g. “ibis” vs “ketch”), which makes the learning less effective. On the other hand, SL-NE finds the local discriminative structure due to its neighborhood embedding formulation, and achieves much higher accuracies than both kNN with Euclidean distance and LMNN. The strength of learning local metric for fine-grained classification has also been verified in [28, 21].

Besides the performance advantage, it is obvious to see the computational saving of SL-NE over LMNN. The total number of triplet constraints to be considered is on the order of $N^3k$ for LMNN which regards all the samples with labels other than $y_i$ as the impostors for $x_i$, while only around $N^2k^2$ for SL-NE which defines similarity exclusively for neighboring samples. Therefore, a significant reduction of computational complexity by a factor of $N/k$ can be achieved for large-scale data with $N \gg k$.

---

**Algorithm 1** Similarity Learning with Neighborhood Embedding (SL-NE)

**Input:** labeled data set $S = \{x_i \in \mathbb{R}^D, y_i \in \mathbb{Z}^+\}$, required margin $b$, initial step size $\rho_0$

**Output:** similarity parameter $M$

1: initialize $M \in \mathbb{R}^{D \times D}$ as identity matrix
2: set $t = 1$
3: while not converge do
4: randomly permute data set $S$
5: for each $(x_i, y_i) \in S$ do
6: choose $(x_j, x_l)$ from constraint-violating pairs according to (6)
7: set step size $\rho = \rho_0 / \sqrt{(t - 1) / |S| + 1}$
8: update $M \leftarrow M + \rho \cdot (x_j - x_l) (x_j - x_l)^T$
9: normalize the Frobenius norm of $M$ and project to symmetric/low rank space (optional)
10: $t \leftarrow t + 1$
11: end for
12: end while
13: return $M$


3. Ensemble of Distributed Similarities

Recent progress in image classification has witnessed the benefits of building features with very high dimensionality [17]. However, metric learning on such high-dimensional features can be prohibitively expensive. Many methods have tried to learn the metric on a subspace with much lower dimension by imposing a low rank constraint on the distance kernel [23, 15], which, unfortunately, makes the problem non-convex [15]. Although the bilinear similarity function in (2) takes a very simple form, it still has computation complexity quadratic to data dimension. To make our method also scalable to high-dimensional data in terms of computation and data I/O, we propose to use an ensemble of low-dimensional subspace projections so that the learning and evaluation of the similarity function can be conducted in a set of low-dimensional spaces distributely. Specifically, based on the similarity in (2), we define an ensemble of similarities as

\[ s_E(x_i, x_j) = \sum_{n=1}^{N_E} s_{M_n}(P_n x_i, P_n x_j) = \sum_{n=1}^{N_E} x_i^T P_n^T M_n P_n x_j, \]  

(7)

where \( \{P_n\}_{n=1}^{N_E} \) is a set of \( d \times D \) matrices that project data samples \( \{x_i\} \) from the original \( \mathbb{R}^D \) space to \( \mathbb{R}^d \) spaces, with \( d < D \). Note it is also possible to use different \( d \)'s for each subspace. \( N_E \) is the number of projections used in the ensemble. \( M_n \) is the parameter of the similarity function in the \( n \)-th projected space. Ideally, \( \{P_n\}_{n=1}^{N_E} \) should be a set of projections capturing complementary discriminative information, with which \( M \) can be transformed into a block matrix. However, learning these discriminative projections can be expensive and thus spoils the algorithm’s scalability. In practice, we find that projections built as consecutive partitions of PCA directions and random projections are both good candidates for \( \{P_n\} \). PCA projections decorrelate the data in different subspaces and thus guarantee their complementarity in certain sense. Random projections can well preserve the distance and recover the solution in high-dimensional space [29], and they can be obtained virtually at no cost.

From (7), we can see that learning the similarity function \( s_{M_n} \) in the \( n \)-th projected space is equivalent to learning a similarity function \( s_M \) in the original space, with low rank and subspace constraints induced by \( P_n \) and imposed on \( M \). From (7), we can further interpret that the ensemble of similarity functions tries to approximate the complete space parameter \( M \) with the summation of a set of parameters \( \{M_n\} \) constrained to subspaces. With sufficiently large \( N \), the ensemble of \( N \) similarity functions learned in their projected low-dimensional spaces have the same capacity as the similarity function learned in the original high dimensional space.

Ensemble learning of multiple metrics has been explored in literatures with different settings. Boosting algorithms are employed to select and combine multiple weak metrics in [13]. The metrics have to be learned sequentially, which is not efficient when the number of metrics is large. The ensemble method proposed here focuses on parallel learning of multiple metrics. Such computational advantage is also leveraged by the random forest metric [26], which regresses the distance function as the average of binary outputs from a set of decision trees.

3.1. Learning Algorithm

Learning the ensemble similarity \( s_E(\cdot , \cdot ) \) requires optimizing the objective function in (5) with the similarity defined in (7). Applying SL-NE in Algorithm 1 to find
all the $M_n$'s jointly induces a computational complexity of $O(N_E \cdot d^2)$, which may not offer too much gain over the original complexity of $O(D^2)$ if a large ensemble size $N_E$ is used. Instead, we propose a Similarity Learning with Distributed Ensemble (SL-DE) algorithm with the computational advantage that each single similarity function in the ensemble can be learned independently in a distributed manner. Given the projection matrix $P_n$, each $M_n$ is learned in parallel in the projected space of $P_n \cdot x$ using SL-NE, which has time complexity $O(d^2)$. The resulting similarity functions can be directly combined to approximate the optimal ensemble similarity $s_E(\cdot , \cdot )$ according to (7). In this way, we can potentially reduce the computational complexity from $O(D^2)$ to $O(d^2)$ if not considering the overhead in parallelization. When computation resource allows, we can further carry out an optional step that jointly optimizes $M_n$'s by minimizing the objective in (5) in a coordinate descent manner.

### 4. Experiments

We test the performance of the proposed SL-NE learning algorithm and its ensemble version SL-DE on several data sets. In all the experiments, we use Locality-constrained Linear Coding (LLC) [22] with 3,000 PCA dimensions to obtain the image feature representation. For the SL-NE and SL-DE methods, we set margin $b=0.02$ and initial step size $\rho_0=0.2$. We also add a local linear offset $w^T(x_i + x_j)$ to the bilinear similarity (2) as in [14], but found it does not help much. The training of SL-DE is performed on a number of distributed nodes in a computer cluster. The local neighborhood $N_i$ is approximately found using an efficient image retrieval system [19]. The neighborhood size is determined by cross-validation.

Our learned similarity functions are used with a soft-voting kNN classifier to make prediction for test samples, where the voting weight are given by the similarity scores. Specifically, for a test sample $x_t$ whose $k$ nearest training neighbors are indexed by a set $N_t$, its class label can be predicted as: $\hat{y}_t = \arg \max_{y \in N_t} \sum_{j \in N_t} s(x_t, x_j)$. The voting members for each test sample are selected from its local neighborhood instead of the entire training set so that the testing is also scalable.

Our SL-NE and SL-DE methods are compared with linear SVM, the Retrieval system [19] which generates the initial neighborhoods, and LMNN [23] whose code is publicly available. A rough comparison on training complexity for all these five methods is given in Table 2. Generally, metric learning methods are more scalable than one-versus-all SVM in terms of the number of classes. Our SL-NE is more scalable than LMNN in terms of the number of training samples due to its neighborhood embedding formulation, and SL-DE further improves over SL-NE on the scalability of data dimension by parallel computation.

| Method     | Sample number | Class number | Data dimension |
|------------|---------------|--------------|----------------|
| Linear SVM | linear constant (p) | linear constant (p) | linear |
| Retrieval [19] | $\approx$linear constant (p) | $\approx$linear constant (p) | linear |
| LMNN [23] | quadratic constant quadratic | quadratic constant quadratic | quadratic |
| SL-NE | $\approx$linear constant quadratic | $\approx$linear constant quadratic | quadratic |
| SL-DE | $\approx$linear constant constant (p) | $\approx$linear constant constant (p) | constant |

[4.1. Algorithm Analysis](#)

We first examine the optimization behavior for SL-NE algorithm in Fig. 2. Two types of triplet sampling strategy are compared: uniform sampling and the selective sampling defined by (6). With either sampling strategy, the performance of the learned similarity function improves a lot over the initial inner product similarity in the original space. Moreover, the proposed selective sampling can converge faster than uniform sampling and attains higher accuracy.

The performance of SL-DE is studied in Fig. 3 with different combinations of projection dimension $d$ and ensemble size $N_E$. Here we project image features from the original 3,000 dimensional space to random subspaces of dimensions ranging from 20 to 1,000. From Fig. 3 (a), we see that the accuracy of SL-DE increases with the size of ensemble, and converges to a certain bound determined by the projection dimension. An ensemble of 10 similarity functions in 1,000 dimensional projected spaces, for example, can achieve almost the same performance as the single similarity function learned in the original 3,000 dimensional space. The total computation for the
Table 3. Top \( n \) classification accuracies (%) for middle scale data sets.

| Data set | Top-\( n \) | SVM | Retrieval [19] | LMNN [23] | SL-NE | SL-DE |
|----------|-------------|-----|----------------|-----------|-------|-------|
| Caltech 101 | 1 | 73.46 | 63.49 | 69.16 | **75.28** | 75.14 |
|          | 3 | 84.59 | 77.87 | 80.83 | 85.29 | **85.30** |
| Caltech 256 | 1 | 43.61 | 37.28 | 37.47 | **46.02** | 45.39 |
|          | 3 | 56.97 | 48.83 | 46.84 | 55.97 | 55.33 |
| SUN      | 1 | 32.08 | 29.06 | – | 35.32 | **35.72** |
|          | 3 | 48.73 | 43.87 | – | 51.37 | 51.37 |

Figure 3. Test accuracy of SL-DE using random projections on the Caltech 101 data: (a) plotted as a function of ensemble size \( N_E \) for various projected dimension \( d \)'s; (b) plotted as a function of projected dimension \( d \) for \( N_E = 1 \) and \( N_E = 20 \) similarity function(s) used in ensemble.

4.2. Results on Middle Scale Data

The classification performance of our methods are validated on several benchmark data sets including the Caltech 101 [9] (9,144 images from 102 object classes), the Caltech 256 [10] (30,607 images from 257 object classes) and the SUN [24] (108,754 images from 397 scene categories). We randomly select 30 samples, 80 samples, and 70% of samples from each class as training set for the Caltech 101, Caltech 256 and SUN data set, respectively. In our SL-NE and SL-DE methods, the local neighborhood size \( |N| \) is chosen as 50/50/500 in both training and testing. The neighborhood size is selected to be about the same as the size of each class in the data sets. For the SL-DE method, an ensemble of similarity functions in 1,000/300/500 dimensional PCA and random subspaces is trained for the three data sets respectively. Note we just choose the ensemble parameters arbitrarily as long as the computation resource allows. The top-1 and top-3 classification accuracies are shown in Table 3, with comparison to several baseline approaches. On all the data sets, our SL-NE method is much better than the unsupervised Retrieval, and most of the time it also outperforms the popular linear SVM classifier by 2~3%. SL-NE also achieves much higher accuracies than LMNN (which cannot complete in a reasonable amount of time on the SUN set). This indicates that the neighborhood embedding formulation can not only reduce training complexity, but also improve the learning effectiveness by focusing on more relevant data. The accuracies attained by the SL-DE method are very close to those of SL-NE, even though SL-DE are learned based on features projected to subspaces with much lower dimensions. On the SUN data set, SL-DE even performs better than SL-NE, which implies that sometimes directly learning a similarity function in high-dimensional space may not be as effective as the ensemble approach.

The similarity functions learned by SL-NE and SL-DE can also be regarded as a rerank function for the similar images found by Retrieval. Therefore, we also evaluate their retrieval performance by plotting the precision-recall curves in Fig. 4. SL-NE consistently improves over Retrieval on all the operation points, and SL-DE achieves very similar performance as SL-NE on the Caltech 256 and SUN data sets.

\[^1\] \( 10 \times 1,000^2 \approx 3,000^2 \)
Table 4. Top-1 flat error rate (%) for ILSVRC’10. Results from [15] are indicated by *.

|            | SVM | Retrieval [19] | LMNN [23] | Mensink et al. [15] | PCA subspace | SL-NE | SL-DE |
|------------|-----|----------------|-----------|---------------------|--------------|-------|-------|
| 73.93/60.2* | 77.19 | 72.90*         | 65.10*    | 80.44               | 66.37        | 68.00 |

4.3. Results on Large Scale Data

To further demonstrate the scalability of our methods, we evaluate their performance on the ImageNet Large Scale Visual Recognition Challenge 2010 (ILSVRC’10) [8] data set, which contains 1.2M training images from 1,000 object classes, 50K validation images, and 150K test images. On this data set, the first 1,000 PCA dimensions of LLC features are used as our original feature. Neighborhood size $|N| = 500$ is used. The ensemble similarity functions for SL-DE are trained on 10 evenly block-partitioned PCA subspaces, each of 100 dimension. The PCA subspaces are preferred here because a small number of them can give a fair result in relatively short time.

We compare in Table 4 the top-1 flat classification error of the proposed methods with baseline methods including linear SVM, Retrieval, two metric learning based approaches [23, 15], and the average performance on the 10 PCA subspaces using SL-NE. All the methods use the same training and testing protocol. Using the same LLC feature, SL-NE achieves much smaller error rate than Retrieval as well as SVM with an online implementation [3], which implies that similarity learning is more advantageous than one-versus-all classification models on data sets with a huge number of classes. Our SL-NE also improves a lot over LMNN, and has a similar performance as Mensink et al. [15]. It should be noted that Mensink et al. have used fisher vector [17] as their feature representation, which gives an error rate more than 10% lower than what LLC achieves when both are used with a SVM classifier. Therefore, the similarity function learned with SL-NE has made up much of the performance loss due to our weaker feature. Given a better feature representation, our method has the potential to further reduce the classification error.

The SL-DE method has a significant improvement over the similarity function learned in each PCA subspace, which serves as its building block. Compared with SL-NE, SL-DE has an error rate less than 2% higher. However, it only takes SL-DE 2 hours (excluding the time to retrieve neighborhood, same below) to train on the 1.2M training set using 10 distributed computers; and this is much faster than SL-NE which needs almost 2 days to complete the same task.

5. Conclusions

A novel image similarity learning method is investigated for better scalability to data set size as well as feature dimensionality with the ideas of large margin neighborhood embedding and ensemble of similarities. The proposed method is validated on several image classification data sets, and achieves competitive accuracies with several existing methods but with much better scalability.

In future work, we will explore other possibilities to find local neighborhoods such as hash function. It is also of great interest to investigate efficient learning of
References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation, 15(6):1373–1396, 2003.

[2] A. Bellet. Supervised Metric Learning with Generalization Guarantees. PhD thesis, University of Saint-Etienne, 2012.

[3] L. Bottou. Stochastic gradient descent package (version 2). http://leon.bottou.org/projects/sgd.

[4] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT, pages 177–186, 2010.

[5] D. Cai, X. He, K. Zhou, J. Han, and H. Bao. Locality sensitive discriminant analysis. In International Joint Conferences on Artificial Intelligence, pages 708–713, 2007.

[6] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. The Journal of Machine Learning Research, 11:1109–1135, 2010.

[7] J. Deng, A. Berg, and L. Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In Proc. CVPR, pages 785–792, 2011.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In Proc. CVPR, pages 248–255, 2009.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding, 106(1):59–70, 2007.

[10] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. CIT Technical Report, 2007.

[11] M. T. Harandi, C. Sanderson, A. Willem, and B. C. Lovell. Kernel analysis over riemannian manifolds for visual recognition of actions, pedestrians and textures. In IEEE Workshop on Applications of Computer Vision, pages 433–439, IEEE, 2012.

[12] S. Hauberg, O. Freifeld, and M. J. Black. A geometric take on metric learning. In Advances in Neural Information Processing Systems, pages 2033–2041, 2012.

[13] D. Kedem, S. Tyree, K. Weinberger, F. Sha, and G. Lanckriet. Non-linear metric learning. In Advances in Neural Information Processing Systems, pages 2582–2590, 2012.

[14] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In Proc. CVPR, pages 3610–3617, 2013.

[15] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE Trans. PAMI, 35(11):2624–2637, 2013.

[16] Y.-K. Noh, B.-T. Zhang, and D. D. Lee. Generative local metric learning for nearest neighbor classification. In Advances in Neural Information Processing Systems, pages 1822–1830, 2010.

[17] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In Proc. ECCV, pages 143–156, 2010.

[18] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500):2323–2326, 2000.

[19] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking. In Proc. CVPR, pages 3013–3020, 2012.

[20] J. Wang, A. Kalousis, and A. Woznica. Parametric local metric learning for nearest neighbor classification. In Advances in Neural Information Processing Systems, pages 1610–1618, 2012.

[21] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In Proc. CVPR, June 2014.

[22] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In Proc. CVPR, pages 3360–3367, 2010.

[23] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res., 10:207–244, 2009.

[24] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In Proc. CVPR, pages 3485–3492, 2010.

[25] E. P. Xing, M. I. Jordan, S. Russell, and A. Ng. Distance metric learning with application to clustering with side-information. In Advances in Neural Information Processing Systems, pages 505–512, 2002.

[26] C. Xiong, D. Johnson, R. Xu, and J. J. Corso. Random forests for metric learning with implicit pairwise position dependence. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 958–966, 2012.

[27] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans. PAMI, 29(1):40–51, 2007.

[28] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In Proc. CVPR, June 2014.

[29] L. Zhang, M. Mahdavi, R. Jin, T. Yang, and S. Zhu. Recovering the optimal solution by dual random projection. In Proceedings of the 26th Annual Conference on Learning Theory, pages 135–157, 2013.

[30] X. Zhu. Semi-supervised learning literature survey. Computer Science, University of Wisconsin-Madison, 2,3, 2006.