Sparsity Induced Similarity Measure for Label Propagation

Hong Cheng
Carnegie Mellon University
Pittsburgh, USA
hongc@cs.cmu.edu

Zicheng Liu
Microsoft Research
Redmond, WA, USA
zliu@microsoft.com

Jie Yang
Carnegie Mellon University
Pittsburgh, USA
jie.yang@cs.cmu.edu

Abstract

*Graph-based semi-supervised learning has gained considerable interests in the past several years thanks to its effectiveness in combining labeled and unlabeled data through label propagation for better object modeling and classification. A critical issue in constructing a graph is the weight assignment where the weight of an edge specifies the similarity between two data points. In this paper, we present a novel technique to measure the similarities among data points by decomposing each data point as an L_1 sparse linear combination of the rest of the data points. The main idea is that the coefficients in such a sparse decomposition reflect the point's neighborhood structure thus providing better similarity measures among the decomposed data point and the rest of the data points. The proposed approach is evaluated on four commonly-used data sets and the experimental results show that the proposed **Sparsity Induced Similarity (SIS)** measure significantly improves label propagation performance. As an application of the SIS-based label propagation, we show that the SIS measure can be used to improve the Bag-of-Words approach for scene classification.*

1. Introduction

Many pattern recognition techniques require labeled data which are often expensive and time consuming to obtain. On the other hand, it is much cheaper to obtain unlabeled data. Therefore, how to combine unlabeled data with labeled data is an important problem, which is the focus of semi-supervised learning techniques. In the past several years, the graph-based semi-supervised learning approach has attracted a lot of attention due to its elegant mathematical formulation and its demonstrated effectiveness in combining labeled and unlabeled data through label propagation [31, 13, 2, 3, 24, 22, 13, 25, 14].

The performance of graph-based semi-supervised learning depends on the weights which are assigned to the edges of the graph. The weight on each edge specifies the similarities between the two nodes that are adjacent to the edge. The simplest method for the weight assignment is to use the Euclidean distances between the feature vectors. A straightforward extension is the K-Nearest Neighbor (KNN) approach where only the edges between a data point and its K-nearest neighbors have non-zero weights. Another extension is to use a Gaussian Kernel Similarity (GKS) [2, 3] as the edge weights. As pointed out by [3, 24], the main drawback with the GKS approach is that its performance is sensitive to the parameter variance and there is no reliable way to determine the optimal variance value especially when the amount of labeled data is small. Wang and Zhang [24] proposed to first approximate a graph by a set of overlapped linear neighborhood patches, and the edge weights in each patch are then computed by neighborhood linear projection. While this method improves the traditional KNN approach by re-adjusting the weight between a point and its k nearest neighbors, it relies on the traditional Euclidean distance to pre-determine its k nearest neighbors. In other words, it does not address the fundamental problem of how to determine the true neighbors in the first place.

In this paper, we propose a new technique to compute the similarities among the data points based on sparse decomposition in L_1 norm sense. We call it **Sparsity Induced Similarity measure (SIS)**. The main idea is that the sparse decomposition of a data point reflects its true neighborhood structure and provides a similarity measure between the data point and its neighbors. In contrast to the Linear Neighborhood Propagation (LNP) [24], the proposed method does not need a separate phase to estimate the neighborhood patches before measuring similarities. In other words, we do not need to reply on the Euclidean distance to pre-determine its k nearest neighbors. Our approach is loosely related to distance metric learning approaches which need more data though these approaches can explore local structures among data [27]. In addition, Shakhnarovich et. al [19] measured patch similarities using sparse overcomplete code coefficients. This technique requires training data to learn the basis vectors. In contrast, our technique does not require any training data for similarity measure.

We evaluate the proposed approach on four data sets, Cedar Buffolo binary Digits data set [12] which is commonly used for evaluating graph-based semi-supervised learning methods, UIUC car data set [1], ETH-80 object data set [16], and scene-15 data set [11, 18, 15], which are commonly-used data sets for object/scene recognition. The experimental results indicate that the proposed SIS measure significantly improves label propagation performance. As an application of the SIS-based label propagation, we show that the SIS measure is useful for codeword assignment in a Bag-of-Words (BoW) approach and its performance is evaluated on the scene-15 data set.

The rest of the paper is organized as follows. We review the label propagation framework in Section 2. Section 3 describes the Sparsity Induced Similarity measure. The experiment results are presented in section 4. We conclude in Section 5.

2. The Framework of Label Propagation

In this section, we review the label propagation framework as described in [31]. We choose to use this method to evaluate our similarity measure because it is a representative graph-based semi-supervised technique which is closely related to other graph-based methods including random walk approach [21], spectral clustering [20] and graph cuts [29].

Label propagation is a way to propagate labels from labeled data to unlabeled data for different applications, for example, patch labelling [4], image annotation [13]. The basic idea is, given a small number of labeled data, to propagate the labels through dense unlabeled regions and find more data with the similar properties as the labeled data, and use these selected unlabeled data to enhance a certain performance of a system. A straightforward solution is to compute pairwise similarities among all the data points, and then formulate the problem as a harmonic energy minimization problem [31] which has a closed-form solution. This technique is briefly summarized below.

Suppose there are K classes. Let $C = \{1, 2, \cdots, K\}$ denote the set of class labels. Let $F_l = [f_1, f_2, \cdots, f_n]$ denote the labeled data. Let $F_u = [f_{n+1}, f_{n+2}, \cdots, f_{n+m}]$ denote the Unlabeled data. Typically $n \ll m$. We use g_i to denote the label of f_i, $i = 1, ..., n+m$. We assume $g_1, ..., g_n$ are known, and the task is to compute $g_{n+1}, ..., g_{n+m}$.

Consider a graph $G = (V, E)$ with nodes corresponding to $N = n + m$ feature vectors. There is an edge for every pair of the nodes. We assume there is an $N \times N$ symmetric weight matrix $W = [w_{ij}]$ on the edges of the graph. The weight for each edge indicates the similarity between the two nodes that are adjacent to the edge. Intuitively, similar unlabeled samples should have similar labels. Thus, the label propagation can be formulated as minimizing the quadratic energy function [31]

$$E(f) = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2. \tag{1}$$

One commonly used similarity measure is the Gaussian Kernel Similarity based weight matrix defined as

$$w_{ij} = \exp\left(-d_{\sigma^2}(f_i, f_j)\right) = \exp\left(-\frac{\|f_i - f_j\|^2}{\sigma^2}\right), \tag{2}$$

where σ is a hyperparameter. As pointed out in [31] and [24], it is hard to determine the optimal value of σ, which causes the instability of label propagation process.

Let D denote an $N \times N$ diagonal matrix with $d_{ii} = \sum_j w_{ij}$. Denote $P = D^{-1}W$. We split matrix W into 4 blocks

$$W = \begin{bmatrix} W_{nn} & W_{nm} \\ W_{mn} & W_{mm} \end{bmatrix}, \tag{3}$$

where W_{nn} is the top left $n \times n$ sub-matrix of W. We split D and P in the same way.

Denote $G_n = (g_1, ..., g_n)^{\mathrm{T}}$, and $G_m = (g_{n+1}, ..., g_{n+m})^{\mathrm{T}}$. It can be shown [31] that given W and G_n, the solution to the energy minimization problem of Eqn. (1) is given by the following formula:

$$G_m = (D_{mm} - W_{mm})^{-1} W_{mn} G_n = (I - P_{mm})^{-1} P_{mn} G_n. \tag{4}$$

In summary, we can propagate labels from the labels G_n of the labeled samples to labels G_m of unlabeled samples using weight matrix W. The performance of such a graph-based label propagation technique relies on the weight matrix, that is, the similarity measure between the nodes. Even though there have been extensive studies on the label propagation techniques, little research has been reported on how to measure the similarities. The most commonly used similarity measure is the Gaussian Kernel Similarity based measure Eqn. (2) whose performance is sensitive to the parameter variance setting. In the next section, we propose a new technique to measure the similarities.

3. The Sparseness Induced Similarity Measure

One main drawback of most of the existing similarity measures such as the Euclidean distance and Gaussian Kernel Similarity measure is that the similarity measurement completely ignores the class structure. In image classification and object recognition, people usually use high dimensional feature vectors while assuming that the feature vectors for each class belong to a lower dimensional subspace. The subspace structure can be discovered when there is enough training data, and researchers have shown that the subspace representation is effective for image classification and object recognition. However, when there is little training data available such as in semi-supervised training or unsupervised training, it is impossible to compute the subspace structure. Consequently, the similarity measurement between feature points are usually based on pairwise Euclidean distance while the subspace structure is ignored. How to leverage the hidden subspace structure for similarity measurement has not been addressed before.

3.1. Sparseness Representation Assumptions

We observe that the subspace assumption is closely related to sparseness representation assumption, and we propose to use sparseness decomposition as a way to define the similarity measurement that takes into consideration of the subspace structure. In particular, our technique is based on the following sparseness representation assumptions on the feature vectors in each class.

Linearity: Any feature vector in a class can be represented as a linear combination of some other feature vectors in the same class.

Sparsity: Given a feature vector, its sparsest representation is achieved when all the basis feature vectors belong to the same class as the feature vector.

The linearity assumption has been used extensively in various computer vision tasks [30, 28, 17]. Note that for a data set with sufficient amount of data (regardless of whether they are labeled or not), the linear representation of a feature vector is usually far from unique. For example, a feature vector may be represented as a linear combination of a number of feature vectors from a different class or from multiple classes. The sparsity assumption states that when a feature vector is represented as a linear combination of feature vectors in a different class, the representation tends to be less sparse. The sparsity assumption is the basis for many sparse sensing researches [28], and it was used in [26] for face recognition. In this paper, we propose to use sparsity assumption as a way to obtain similarity measurement that reflects the subspace structure of classes.

Note that if the sparsity assumption is strictly satisfied, the sparseness decomposition will provide a simple method for unsupervised clustering. For each feature vector V, we decompose it as a sparse linear combination of the rest of the feature vectors. The feature vectors that have non-zero coefficients in the decomposition will be in the same class of V. After performing this decomposition for every feature vector V, we will be able to group them into connected components, and it is guaranteed that the feature vectors in each connected components belong to the same class.

In practice, the data are noisy. Thus the sparsity assumption may not be strictly satisfied. In fact, a random noise vector in general has a long tail (i.e., many small non-zero coefficients) in their sparse decomposition. Therefore making binary decisions does not work well. Thus, we instead use the coefficients as soft similarity measures.

3.2. The Definition of SIS

More formally, we propose the following *Sparseness Induced Similarity Measure*. Let $F = \{f_1, f_2, ..., f_N\}$ denote all the feature vectors of a data set regardless of whether they are labeled or not, where $f_k \in \mathbb{R}^D$. For any given $f_k \in F$, we first decompose f_k as a sparse linear combination of the rest of the feature vectors in F. Let G_k denote the matrix that consists of f_1,...,f_{k-1}, f_{k+1},..., f_N as its columns, that is, $G_k = (f_1, ..., f_{k-1}, f_{k+1}, ..., f_N)$. Let $X = (x_1, ..., x_{k-1}, x_{k+1}, ..., x_N)^T$ denote the coefficients of the sparse decomposition. Given $F = (f_k, G_k)$, X is defined by the following optimization problem:

$$\min_X \|X\|_{\ell_0}, \text{ s.t. } G_k X = f_k, \tag{5}$$

where $\|X\|_{\ell_0}$ is the ℓ_0 norm of X.

This decomposition is different from the most common sparse decomposition problem in that the basis vectors are not necessarily orthogonal. In fact, strictly speaking, vectors in G_k may not form a basis. But Donoho and Elad [9, 8] showed that such non-orthogonal sparse decomposition problem can still be solved through ℓ_1 minimization. That is, X can be obtained by solving the following linear programming problem:

$$\min_X \|X\|_{\ell_1}, \text{ s.t. } G_k X = f_k, \tag{6}$$

where $\|X\|_{\ell_1}$ is the ℓ_1 norm of X.

We convert it to a standard linear programming problem by introducing variables x_i^+ and x_i^-, and setting $x_i = x_i^+ - x_i^-$ and $|x_i| = x_i^+ + x_i^-$, $1 \leq i \leq N, i \neq k$, In addition, we add constraints $x_i^+ \geq 0$ and $x_i^- \geq 0$. The resulting linear programming problem is then solved by a simplex algorithm [5] [1].

If the amount of data is large, it becomes expensive to solve the linear programming problem of Eqn. (6) for each feature vector. It has been shown [10, 5] that for ℓ_1-norm based signal reconstruction, the number of basis vectors required to recover a sparse signal is only a small fraction of the signal's dimension. However, it is impossible to know a priori what vectors should be selected as basis vectors. A heuristics that we use in our experiments is the following. Given a feature vector f_k, we choose the first CD vectors that are closest to f_k in terms of Euclidean distance where D is the feature vector dimension and C is a user-specified parameter which is set to 1.5 in our experiments. The value of C is a compromise between computation cost and the quality of the sparse decomposition. In general, C needs to be greater than 1 to prevent the linear system from being over constrained due to noises in the feature vectors. The larger the C, the more expensive the computation and the better the

[1] www.l1-magic.org

quality of sparse decomposition. Empirically, we find that C=1.5 provides a good tradeoff in our experiments.

The similarity between f_k and f_i, $1 \leq i \leq N, i \neq k$, is defined as

$$s_{ki} = \frac{\max\{x_i, 0\}}{\sum_{j=1, j\neq k}^{N} \max\{x_j, 0\}}. \tag{7}$$

After we repeat this procedure for every $f_k \in F$, $k = 1, ..., N$, we obtain a matrix s_{ij}, $1 \leq i, j \leq N$. Note that this matrix is not necessarily symmetric. To ensure symmetry, the final similarity between f_i and f_j is defined as $w_{ij} = \frac{s_{ij}+s_{ji}}{2}$. We set $w_{ii} = 1$.

Prior to computing the SIS, we need to normalize all the feature vectors so that their L_2 norms are 1. Normalization is necessary because otherwise the decomposition coefficients would be sensitive to the magnitudes of the feature vectors.

We would like to note that the sparse coefficients in Eqn. (5) are related to non-orthogonal projection coefficients onto the sparse basis. Let $X^* = (x_1^*, ..., x_N^*)^{\mathrm{T}}$ denote the solution of Eqn. (5). Let $x_{u_1}^*, ..., x_{u_k}^*$ denote the nonzero coefficients. Denote $\hat{X}_k^* = (x_{u_1}^*, ..., x_{u_k}^*)^{\mathrm{T}}$, and $\hat{G}_k = (f_{u_1}, ..., f_{u_k})$. Then Eqn. (5) becomes

$$\hat{G}_k \hat{X}_k^* = f_k. \tag{8}$$

The columns of $\hat{G}_k$ must be linearly independent because otherwise there would be a solution sparser than X^*. Therefore,

$$\hat{X}_k^* = \left((\hat{G}_k)^{\mathrm{T}}\hat{G}_k\right)^{-1} (\hat{G}_k)^{\mathrm{T}} f_k. \tag{9}$$

In other words, the sparse coefficients $(x_{u_1}^*, ..., x_{u_k}^*)$ are non-orthogonal projection coefficients of f_k onto the vectors $(f_{u_1}, ..., f_{u_k})$.

3.3. A Toy Problem

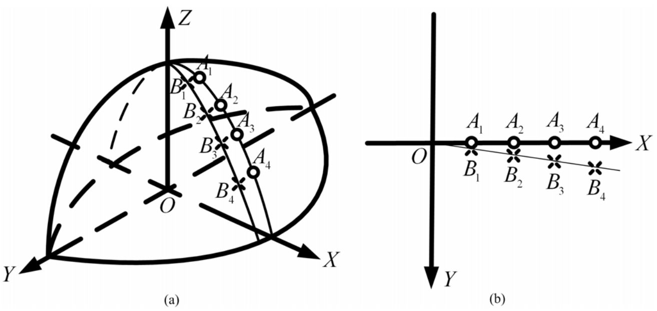

Figure 1. An illustration of a two-class classification problem: (a) Points on a $3D$ sphere; (b) Points projected to a $2D$ plane only for better illustration of spatial relationship among points.

Let us use a toy problem to illustrate how sparsity representation can be used to improve similarity measure. Figure 1 (a) shows a two class classification problem. The points in each class belong to a linear subspace. Points A_1, A_2, A_3, and A_4 belong to class A. Points B_1, B_2, B_3, and B_4 belong to class B. Note that all the points are on the unit sphere because we assume they are normalized feature vectors. Figure 1(b) is obtained by projecting the $3D$ points to $2D$ XY plane for better visualization of the spatial relationship among the points. The coordinates of these points are

$$\begin{array}{llll} A_1 & = [0.1, 0, 0.9950], & B_1 & = [0.1, 0.025, 0.9947] \\ A_2 & = [0.2, 0, 0.9798], & B_2 & = [0.2, 0.050, 0.9785] \\ A_3 & = [0.3, 0, 0.9539], & B_3 & = [0.3, 0.075, 0.9510] \\ A_4 & = [0.4, 0, 0.9165], & B_4 & = [0.4, 0.100, 0.9110]. \end{array}$$

It can be easily verified that for each point A_i, $i = 1, ..., 4$, its closest point is B_i according to the Euclidean distance. Similarly, point B_i's closest point is A_i. Figure 2(b) shows the label propagation result obtained by computing similarities based on the Euclidean distance and using A_4 and B_1 as the labeled data. We can see that A_1 and A_2 are incorrectly labeled as in class B while B_3 and B_4 are incorrectly labeled as in class A.

On the other hand, let us represent point A_2 as a sparse linear combination of the rest of the points. That is, we seek coefficients, $x_1, x_3, x_4, y_1, y_2, y_3, y_4$, so that

$$A_2 = \sum_{i=1, i\neq 2}^{4} x_i A_i + \sum_{j=1}^{4} y_j B_j, \tag{10}$$

and the number of non-zero coefficients is the smallest. It can be verified that the sparsest decomposition is given by

$$A_2 = 0.5079 A_1 + 0.4974 A_3. \tag{11}$$

In this representation, A_1 has the largest coefficient, and A_3 has the second largest coefficient. The rest of the coefficients are all zero. Based on the coefficients, we conclude that A_2 is most similar to A_1 and A_3. We can see that this similarity measure is more consistent with the class structure. Figure 2(a) shows the label propagation result obtained by using SIS measure and using A_4 and B_1 as the labeled data. We can see that all the points are correctly labeled.

Note that if we use linear decomposition without sparsity constraints, the resulting coefficients do not provide a good similarity measure. Again, let us consider A_2 in the above example. Since there are multiple ways to represent A_2 as a linear combination of the rest of the vectors, one possibility, as suggested by Wang and Zhang [24], is to choose a small number of nearest neighbors (in terms of Euclidean distance). If we choose the 2-nearest neighbors of A_2, which are A_1 and B_2, we obtain the following least-square solution:

$$A_2 \approx 0.1361 A_1 + 0.8639 B_2. \tag{12}$$

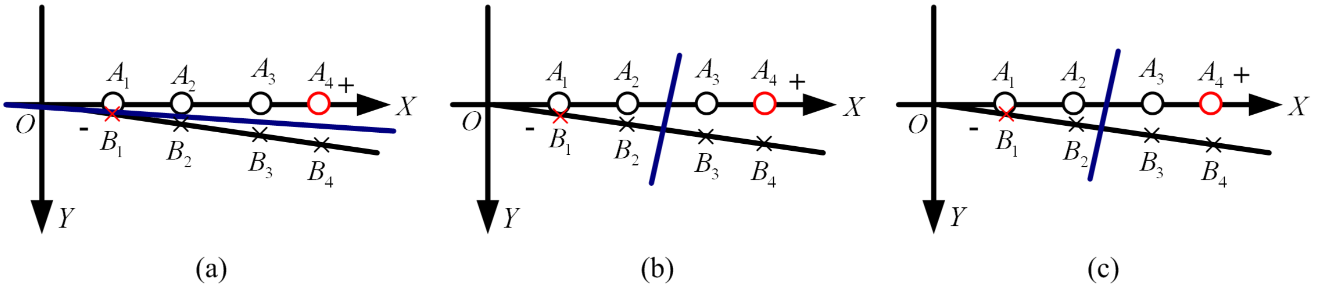

Figure 2. Label propagation using 3 different similarity measures: (a) SIS; (b) Euclidean distance; (c) Linear neighbors in [24], where the number of nearest neighbors is 2. Red points A_4 and B_1 are labeled data, and the blue lines are class boundaries.

According to this representation, B_2 would be considered as the most similar to A_2. Figure 2(c) shows the label propagation result obtained by using Wang and Zhang's method for the similarity measure. Again, A_4 and B_1 are used as the labeled data. We can see that the result is the same as what is obtained by computing similarities based on Euclidean distance.

4. Experimental Results and Analysis

In this section, we present four sets of experiments to validate the proposed approach. Section 4.2, 4.3, 4.4, and 4.5 describe experiments on Cedar Buffalo digits data set, UIUC car data set, ETH-80 object data set, and scene-15 data set, respectively.

4.1. Experimental Setup

Data sets: We used four data sets in our experiments, which are commonly used for semi-supervised learning and object/scene recognition experiments. We evaluated the proposed SIS measure on the Cedar Buffalo binary digit data set [12]. The digits are preprocessed to reduce the size of each image down to a 16×16 by down-sampling and Gaussian smoothing, and the value of each pixel ranges from 0 to 255. Each digit is thus represented by a 256-dimensional vector.

For evaluating the performance of label propagation for object data sets, we used both ETH-80 data set [16] and UIUC car data set [1]. ETH-80 contains 8 object categories. In each category there are 10 different objects, and for each object there are 41 different poses. There are $8 \times 10 \times 41 = 3,280$ images in total. Here, similar to [16], we use the histogram of the first derivatives D_xD_y with 48 dimensions over 3 different scales to represent each image, and then all features are normalized. The UIUC car training data set consists of 1050 images of cars in side views with resolution 40(H)×100(W) pixels. For this data set, we use dense grids of histogram-of-gradient features to represent each image [7], where 20×20 pixel blocks, block stride of 10 pixels, and 8 orientation bins are used to obtain the feature vector of 240 dimensions for each image.

Furthermore, we conducted experiments on scene classification task based on the data set of scene-15 data set [11, 18, 15] which consists of 4485 images with different resolutions over 15 categories. Each category has 200 to 400 images. As in [15], we use dense SIFT descriptors of 128 dimensions on 16×16 pixel patches and spacing pixels are 8, and skip the usual SIFT normalization procedure when the overall gradient magnitude of the patch is too small. To eliminate the effect of SIFT feature vectors whose L_2 norms are less than 1 on L_1 decomposition, we normalize SIFT feature vectors $f_k \in \mathbb{R}^D$ by increasing one dimension

$$f_k(D+1) = 1 - \sqrt{f_k(1)^2 + \cdots + f_k(D)^2}. \quad (13)$$

Similar to [23], we choose 100 random images per category as a training set and the remaining images as testing images.

Labeled and unlabeled samples: Similar to [31], we randomly sample labeled samples from the entire samples of each class, and the rest of the samples of this class are used as unlabeled data to evaluate different similarity measure approaches in the first three experiments. However, we fix codewords as labeled samples and dense patches as unlabeled samples in the fourth experiment due to application requirement.

Evaluation Criterion: For the first three experiments, We employed recognition accuracy to evaluate the performance of the proposed SIS on label propagation. Each recognition accuracy curve is obtained by averaging the results over 10 different trials. For each trial, again, we randomly select the labeled and unlabeled samples for the semi-supervised label propagation. For the scene classification experiment, similar to [23, 15], we use the classification accuracy of final scene classification to evaluate the performance of two patch labeling approaches.

4.2. The Cedar Buffalo Binary Digits Data Set

This experiment compared the proposed SIS measure with the other three similarity measures, GKS, Linear Neighbor Similarity(LNS) [24], and KNN on digits data set.

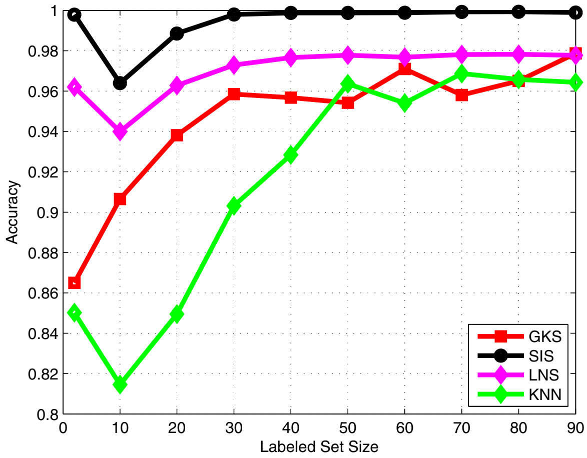

Figure 3. The accuracy of label propagation on digits ‘1’ and ‘2’ based on semi-supervised learning.

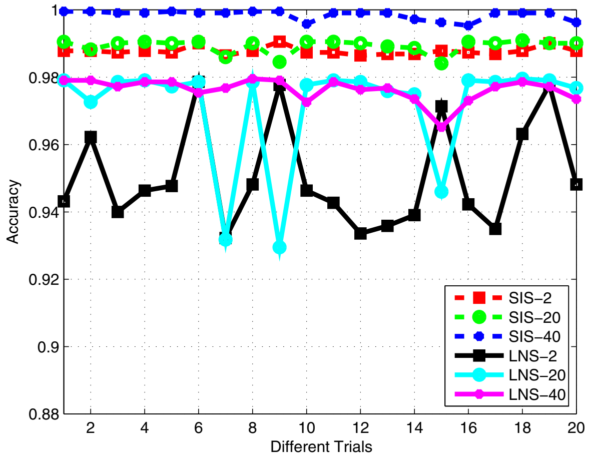

Figure 4. The performance stability of different similarity measures for the different number of labeled samples.

For GKS, we used Eq. (2) to measure similarity and the variance σ is set to 380 which achieves the best performance. For LNS, we obtain the weight matrix by optimizing an objective function based on a linear neighbor assumption in the LNP approach [24] while the label propagation scheme is similar to [31]. The number of nearest neighbors K is set to 10 (Both 5 and 10 were used in [24]. We found that 10 works better for this example). As for KNN, We use inner-product similarity to find the K nearest neighbors. Then, the similarity values between a sample and its K nearest neighbors are their correlation coefficients while those between the sample and the rest are set to 0. The value of K is set to 10. Note that, in this paper, we consider KNN just as a type of similarity measure, not as a classifier as in [31, 24]. In our SIS approach, we normalize all feature vectors so that their L_2 norms are 1 before computing weight matrix.

Figure 3 shows the propagation accuracies of four different similarity measures: GKS, SIS, LNS, and KNN on digits ‘1’ and ‘2’. The x-axis is the number of labeled data ranging from 2 to 90. The y-axis is label propagation accuracy. We can see that GKS (labeled as ‘GKS’) works better than KNN (labeled as ‘KNN’), LNS (labeled as ‘LNS’) works better than GKS, and SIS (labeled as ‘SIS’) works the best.

We also use the digit data set to study the performance stability when the amount of labeled samples varies. Since Wang and Zhang [24] showed that the performance of LNS is better than that of GKS in performance stability, we only compare SIS with LNS. In Figure 4, the x-axis is the index of different trials ranging from 1 to 20, and y-axis is the accuracy resulted from label propagation. We compare with three different labeled data sizes: 2, 20, and 40 . The curve ‘SIS-2’ denotes the label propagation accuracy obtained by using SIS with 2 labeled samples, and the curve ‘SIS-20’ denotes the accuracy obtained by using SIS with 20 labeled samples, and so on. Similarly, the curve ‘LNS-2’ denotes the labeling accuracy using LNS with 2 labeled samples, etc. We can see that the proposed similarity measure has very stable performance for different labeling data sizes, while the LNS approach exhibits large performance fluctuations.

4.3. The UIUC Car Data Set

In this experiment, we investigate the SIS performance for label propagation of UIUC car data set. Two classes, 1050 images of cars and backgrounds, are used to validate the proposed approach. For GKS, the variance σ is set to 0.5 and the value of K is set to 10 in KNN. Similarly, the number of nearest neighbors is set to 10 in LNS.

Figure 5 compares the label propagation accuracy resulted from four different similarity measures. We select the labeled samples randomly from 1000 images of cars and backgrounds, and use 2, 10, 20, 30, $\cdots$, 100 labeled samples to evaluate the effects of different labeled data sizes. Again, the proposed similarity measure significantly outperforms the other approaches.

4.4. The ETH-80 Object Data Set

In this experiment, we evaluate the SIS performance for label propagation of multi-class objects. 3 types of objects, apples, pears and tomatoes are used to evaluate the proposed similarity measure for multi-class label propagation since it is comparatively difficult to distinguish those three categories in this data set. For GKS, the variance σ is set to 0.15 and the value of K is set to 10 in KNN. Similarly, the number of nearest neighbors is set to 10 in LNS.

Figure 6 compares four different similarity measures on ETH-80. We select the labeled samples randomly from 1,230 images of apples, cars, and cows. We use 3, 9, 18, 27, $\cdots$, 81 labeled samples to evaluate the effect on different labeled data sizes. Again, the proposed similarity measure

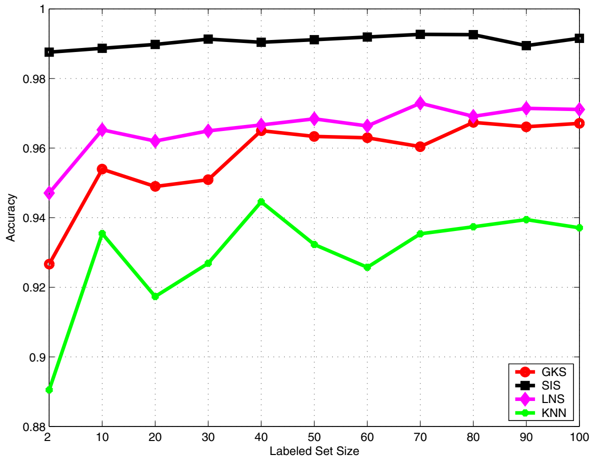

Figure 5. The accuracy of label propagation on UIUC data set based on semi-supervised learning.

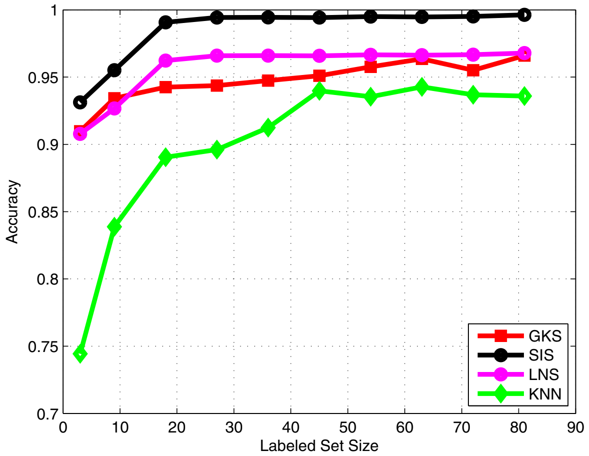

Figure 6. An accuracy comparison of GKS, SIS, LNS, KNN on the ETH-80 data set.

outperforms the rest.

4.5. The Scene-15 Data Set

This experiment evaluates the proposed similarity measure in the framework of BoW for scene classification on scene-15 data set. The basic idea of BoW is to sample a representative set of patches from each image, compute a feature descriptor for each patch, characterize the resulting distributions and finally classify images based on the distributions. In this experiment, we focus on the third issue, that is, how to characterize the resulting distributions. Basically, it consists of codebook generation and codeword assignment. Similar to traditional codebook generation, we use K-means approaches to generate a codebook. For codeword assignment, two popular approaches, hard assignment and soft assignment, are used to assign codewords to the patches thus forming the patch distributions (i.e., histogram). In the hard assignment approach, each patch is assigned with a single codeword. In contrast, the soft assignment allows multiple codewords (with weights) to be assigned to a patch. A Gaussian-kernel soft assignment approach was proposed in [23]. Essentially, it uses GKS measure to determine the similarities between the patches and the codewords.

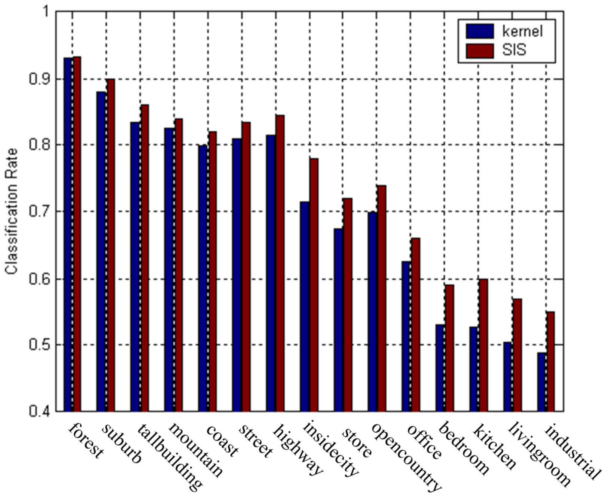

Figure 7. A performance comparison on scene-15 data set of two patch labeling approaches: the kernel-based approach and the SIS-based approach.

As an alternative to kernel-based codeword assignment [23], we use the SIS-based label propagation technique to determine the soft assignment. Given h_1 patches for one image and h_2 codewords for a codebook, we first obtain feature vector matrix F in Eq. (5), where the total number of feature vectors is $N = h_1 + h_2$. Second, we compute the weight matrix using the approach introduced in Sect. 3.2. Third, we use the codewords in F to represent the labeled samples and use the patch descriptors to represent the unlabeled samples. The label propagation result of Eq. (4) is used as the soft assignment for the patches. Similar to [23], for each image we accumulate the soft assignment values of its patches to form the resulting histogram for the image. The histograms are then used for classification.

In Figure 7, we compare the scene classification performance obtained by using SIS-based patch labeling approach (labeled as ‘SIS’) with what is obtained by using the Gaussian-kernel patch labeling approach [23] (labeled as ‘kernel’). To ensure a fair comparison with the kernel based patch labeling approach, we closely follow [23] in the experiment setup. As in [23], we perform K-means clustering to form the codebook of 200 codewords, and repeat the experiment 10 times to obtain reliable results. For classification, libSVM [6] and a histogram intersection kernel are used in one-versus-one multi-class classification. Figure 7 shows the classification rates obtained from using the two patch labeling approaches. We can see that our approach outperforms the kernel-based approach for every category. On average, we achieve 74.94% classification rate on scene-15 data set with an absolute improvement of 3.5% over kernel based approach. Furthermore, SIS based scene classification with 200 codewords achieves nearly the same performance as the kernel-based approach with 1600 codewords.

5. Conclusions

In this paper, we have proposed a novel similarity measure for propagating labels form labeled samples to unlabeled samples. The proposed SIS measure takes into account the hidden class structure by using the sparse decomposition. We have compared the proposed similarity measure with the traditional similarity measures including GKS and KNN, and the experiment results demonstrated the superiority of the proposed method. In addition, we showed that the proposed similarity measure can be used to improve the codeword assignment in the BoW framework for scene classification.

Acknowledgments

This research was partially supported by the gift grants from the General Motors, Microsoft and the grant from NSFC (No. 60705024). We also thank the anonymous reviewers for their valuable suggestions, and the discussion of BoF methods with Lei Yang. The third author was partially supported by NSF.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. ECCV*, 2002. 2, 5

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2002. 1

[3] M. Belkin and P. Niyogi. Learning with local and global consistency. In *NIPS*, 2004. 1

[4] C. M. Bishop and I. Ulusoy. Object recognition via local patch labelling. In *Proceedings 2004 Workshop on Machine Learning*, 2005. 2

[5] E. J. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207 – 1223, 2008. 3

[6] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. 7

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, 2005. 5

[8] D. Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Comm. Pure and Applied Math*, 59(6):797–829, 2006. 3

[9] D. L. Donoho and M. Elad. Maximal sparsity representation via l1 minimization. In *the Proc. Nat. Aca. Sci. 100*, pages 2197–2202, 2003. 3

[10] D. L. Donoho and J. Tanner. Counting faces of randomly-projected polytopes when the projection radically lowers dimension. *Journal of the America Mathematical Society*, 22(1):1–53, 2009. 3

[11] L. Feifei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE CVPR*, 2005. 2, 5

[12] J. J. Hull. A database for handwritten text recognition research. *IEEE PAMI*, 16(5):550–554, 1994. 2, 5

[13] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *IEEE CVPR*, 2006. 1, 2

[14] T. Kato, H. Kashima, and M. Sugiyama. Robust label propagation on multiple networks. *IEEE TNN*, 20(1):35–44, 2009. 1

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE CVPR*, 2006. 2, 5

[16] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *IEEE CVPR*, 2003. 2, 5

[17] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *Proc. ECCV*, 2008. 3

[18] A. Oliva and A. Torralba. Modeling the shapes of the scene: a holistic representation of the spatial envelope. *IJCV*, 42:145–175, 2001. 2, 5

[19] G. Shakhnarovich. Learning task-specific similarity, 2006. PhD Thesis, MIT. 1

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, 2000. 2

[21] J. Shi and J. Malik. Partially labeled classification with markov random walks. In *NIPS*, 2000. 2

[22] A. Singh, R. D. Nowak, and X. Zhu. Unlabeled data: now it helps, now it doesn't. In *NIPS*, 2008. 1

[23] J. C. van Gemert and J. Geusebroek. Kernel codebooks for scene categorization. In *Proc. ECCV*, 2008. 5, 7

[24] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *ICML*, 2007. 1, 2, 4, 5, 6

[25] J. Wang, S. F. Chang, X. Zhou, and S. T. C. Wong. Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. In *IEEE CVPR*, 2008. 1

[26] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE PAMI*. 3

[27] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. 2002. 1

[28] J. Yang, J. Wright, Y. Ma, and T. Huang. Image super-resolution as sparse representation of raw image patches. In *IEEE CVPR*, 2008. 3

[29] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *IEEE CVPR*, 2004. 2

[30] X. Zhang, L. Liang, X. Tang, and H. Shum. L_1 regularized projection pursuit for additive model learning. In *IEEE CVPR*, 2008. 3

[31] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *IEEE ICML*, 2003. 1, 2, 5, 6