Locality-Preserving L1-Graph and Its Application in Clustering

Shuchu Han, Hao Huang, Hong Qin
Computer Science Department
Stony Brook University (SUNY)
Stony Brook, NY 11794
{shhan,haohuang,qin}@cs.stonybrook.edu

Dantong Yu
Computational Science Center
Brookhaven National Lab
Upton, New York 11973
dtyu@bnl.gov

ABSTRACT

Constructing a good graph to represent data structures is critical for many important machine learning tasks such as clustering and classification. Recently, a nonparameteric graph construction method called $\mathcal{L}_1$-graph is proposed with claimed advantages on sparsity, robustness to data noise and datum-adaptive neighborhood. However, it suffers a lot from the loss of locality and the instability of performance. In this paper, we propose a Locality-Preserving $\mathcal{L}_1$-graph (LOP-$\mathcal{L}_1$), which preserves higher local-connections and at the same time maintains sparsity. Besides, compared with $\mathcal{L}_1$-graph and the succeeding regularization-based techniques, our LOP-$\mathcal{L}_1$ requires less amount of running time in the scalability test. We evaluate the effectiveness of LOP-$\mathcal{L}_1$ by applying it to clustering application, which confirms that the proposed algorithm outperforms related methods.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

theory

Keywords

locality, sparsity, $\mathcal{L}_1$-graph

1. INTRODUCTION

Among many techniques used in the machine learning society, graph-based mining mainly tries to accommodate the so-called cluster-assumption, which says that samples on the same structure or manifold tend to have large weight of connections in-between. But most of the time there is no explicit model for the underlying manifolds, hence most methods approximate it by the construction of an undirected/directed graph from the observed data samples. Therefore, correctly constructing a good graph that can best capture essential data structure is critical for all graph-based methods [16].

Ideally, a good graph should reveal the intrinsic relationship between data samples on manifold, and also preserve the strong local connectivity inside neighborhood (called as locality in the following paper). Traditional methods (such as k-nearest neighbors (kNN) [7], ϵ-neighborhood [7] and Gabriel graph (GG) [2]) mainly rely on pair-wise Euclidean distances to construct the locally-connected graph. The obtained graphs oftentimes fail to capture local structures and cannot capture global structures of the manifold [16]. Besides, these methods either cannot provide datum-adaptive neighborhoods because of using fixed global parameters [2], or are sensitive to the parameter setting or local noise especially on high-dimensional datasets [8].

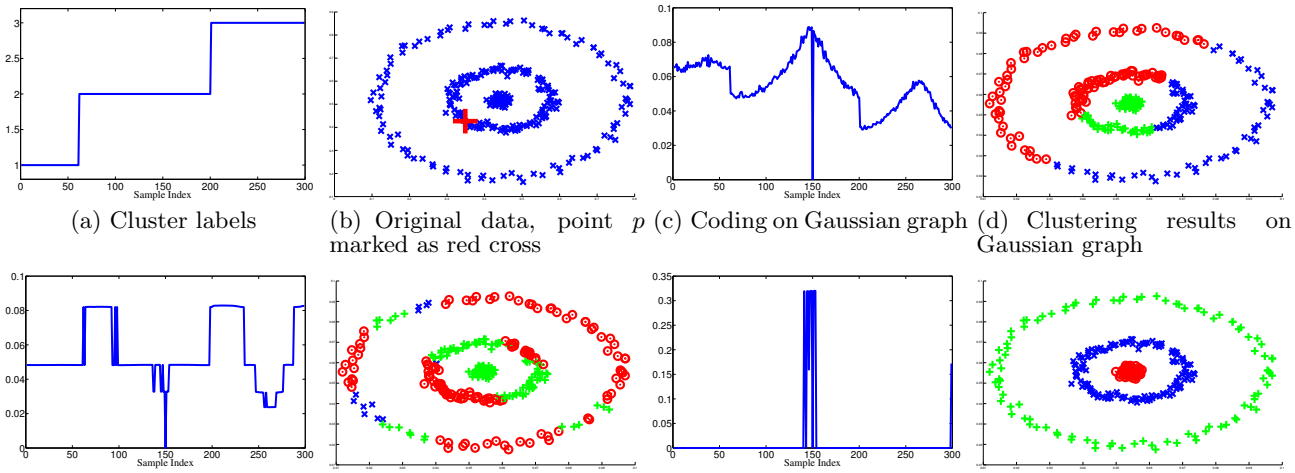
Recently, Cheng et al. [1] proposed to construct an $\mathcal{L}_1$-graph via sparse coding [12] by solving an $\mathcal{L}_1$ optimization problem. $\mathcal{L}_1$-graph is derived by encoding each datum as a sparse representation of the other samples (treated as basis or dictionary pool), and automatically selecting the most informative neighbors for each datum. The nice properties of $\mathcal{L}_1$-graph include: 1) sparsity, which leads to fast subsequent analysis and low requirement for storage [12], 2) datum-adaptive neighborhoods and 3) robustness to data noise as claimed in [1].

However, the constructing of classic $\mathcal{L}_1$-graph suffers from the loss in the locality of the samples to be encoded, which is a fundamental drawback from sparse coding [6]. Usually, the number of samples is much greater than the number of manifold dimensions, which means that the basis pool is "overcomplete" during the construction of $\mathcal{L}_1$-graph. Samples may be encoded with many basis (samples) with weak correlations with the object samples under such "overcomplete" basis pool. Thus, it results in the inaccuracy of $\mathcal{L}_1$-graph, and therefore impedes the quality of the consequent analysis tasks. As an illustration, Fig.1(e) shows that under classic $\mathcal{L}_1$-graph construction, the code of a sample point p (red cross in Fig.1(b)) involves many basis (samples) that do not belong to the same cluster with p. Such instability may hinder the robustness of the $\mathcal{L}_1$-graph based data mining applications, as shown in Fig.1(f). To address this issue, we propose a Locality-Preserving $\mathcal{L}_1$-graph (LOP-$\mathcal{L}_1$) to learn more discriminative sparse code and preserve the locality and the similarity of samples in the sparse coding process, and therefore the robustness of the data analysis result is enhanced. Our contributions are as follows:

1. LOP-$\mathcal{L}_1$ **preserves locality** in an datum-adaptive neighborhood, and at the same time maintains sparsity from

SAC'15 April 13-17, 2015, Salamanca, Spain.

http://dx.doi.org/10.1145/2695664.2695710



(a) Cluster labels (b) Original data, point p marked as red cross (c) Coding on Gaussian graph (d) Clustering results on Gaussian graph

(e) Coding on $\mathcal{L}_1$-graph (f) Clustering results on $\mathcal{L}_1$-graph (g) Coding on LOP-$\mathcal{L}_1$ (h) Clustering results on LOP-$\mathcal{L}_1$

Figure 1: Illustration of LOP-$\mathcal{L}_1$ effectiveness compared with Gaussian (similarity) graph and classic $\mathcal{L}_1$. The labels of sample in the original dataset (Fig.1(b)) are showed in Fig.1(a), and in this example we only focus on the coding of point p (the 150-th sample, marked as red cross in Fig.1(b)). Coding (similarity) of p on Gaussian graph (Fig.1(c)) is built upon Euclidean space, which leads to manifold non-awareness (Fig.1(d)). Classic $\mathcal{L}_1$ graph coding (Fig.1(e)) results in the loss of locality and therefore instable clustering result (Fig.1(f)). Comparatively, our LOP-$\mathcal{L}_1$ coding on p (Fig.1(g)) shows strongly locality-preserving characteristic and has the best performance in clustering, as shown in Fig.1(h).

classic $\mathcal{L}_1$.

2. The computation of LOP-$\mathcal{L}_1$ is **more scalable** than classic $\mathcal{L}_1$ graph and the succeeding regularization-based techniques.

3. We confirm the **effectiveness** of LOP-$\mathcal{L}_1$ in the application of clustering.

2. RELATED WORK

$\mathcal{L}_1$-graph is an informative graph construction method proposed by Cheng et al. [1]. It represents the relations of one datum to other data samples by using the coefficient of its sparse coding. The original $\mathcal{L}_1$-graph construction algorithm is a nonparametric method based on the minimization of a $\mathcal{L}_1$ norm-based object function. The details of $\mathcal{L}_1$-graph construction are documented in Algorithm 1.

The advantages of $\mathcal{L}_1$-graph are summarized as follows: (1) robustness to data noise; (2) sparsity for efficiency; and (3) datum-adaptive neighborhood. Because of these virtues, $\mathcal{L}_1$-graph has been applied to many graph based learning applications [1], for example, subspace learning [1], image classification [16] and semi-supervised learning [13] etc. However, classic $\mathcal{L}_1$-graph [1] is a purely numerical solution without physical or geometric interpretation of the data set [3]. Therefore, to better exploit the structure information of data, many research works have been proposed by adding a new regularization term in addition to the original Lasso penalty, for example, the elastic net regularization [3]. OSCAR regularization [3] and graph-Laplacian [14].

Another research focus of $\mathcal{L}_1$-graph is to reduce its high computational cost. For each datum, the $\mathcal{L}_1$-graph need to solve an $\mathcal{L}_1$ minimization problem within a large basis pool which is very slow. To reduce the running time, Zhou et al. [15] proposed a kNN Fused Lasso graph by using the

Algorithm 1: $\mathcal{L}_1$-Graph

Input : Data samples $X = [x_1, x_2, \cdots, x_N]$, where $x_i \in \mathbb{R}^m$

Output: Adjacency matrix W of $\mathcal{L}_1$ graph.

Normalize the data sample x_i with $\|x_i\|_2 = 1$;
for $x_i \in X$ **do**
  Solve: $\min_{\alpha_i} \|\alpha_i\|_1, \quad s.t. \quad x_i = B^i \alpha_i$;
  where
  $B^i = [x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N, I] \in \mathbb{R}^{m \times (m+N-1)}$
  and $\alpha_i \in \mathbb{R}^{m+N-1}$;
end
for $i = 1 : N$ **do**
  for $j = 1 : N$ **do**
    if $i > j$ **then**
      $W_{ij} = \alpha_i(j)$
    else if $i < j$ **then**
      $W_{ij} = \alpha_i(j-1)$
    else
      $W(i,j) = 0$
    end
  end
end

k-nearest neighbors idea in kernel feature space. With a similar goal, Fang et al. [3] proposed an algorithm which firstly transfers the data into a reproducing kernel Hilbert space and then projects to a lower dimensional subspace. By these projections, the dimension of the dataset is reduced and the computational time decreased.

In our research we evaluate the performance of different graph constructions in terms of clustering. Specifically we

integrate the constructed graph into the framework of spectral clustering, due to its popularity and its ability to discover embedding data structure.

Spectral clustering, as shown in Algorithm 2, starts with local information encoded in a weighted graph on input data, and clusters according to the global eigenvectors of the corresponding (normalized) affinity matrix. Particularly, to satisfy the input of spectral clustering algorithm, we transform the adjacency matrix of $\mathcal{L}_1$-graph into a symmetry matrix with the first step in Algorithm 2.

Algorithm 2: SpectralClustering

Input : Adjacency matrix W of $\mathcal{L}_1$-graph; K is the number of clusters;
Output: Cluster assignments.

1 Symmetrize the graph similarity matrix by setting the matrix $W = (W + W^T)/2$;
2 Set the graph Laplacian matrix $L = D^{-1/2}WD^{-1/2}$, where $D = [d_{ij}]$ is a diagonal matrix with $d_{ii} = \sum_j W_{ij}$;
3 Find $c_1, c_2, \cdots, c_K$, the eigenvectors of L corresponding to the K largest eigenvalues, and form the matrix $C = [c_1, c_2, \cdots, c_K]$ by stacking the eigenvectors in columns;
4 Treat each row of C as a point in R^K, and cluster them into K clusters via the k-means method;
5 Finally, assign x_i to the cluster j if the ith row of the matrix C is assigned to the cluster j;

3. ALGORITHM

The construction of classic $\mathcal{L}_1$ graph [1] is a global optimization which is short of local-structure awareness. Moreover, it has a high time complexity, since for each datum it needs to solve a $\mathcal{L}_1$-minimization problem:

$$\min_{\alpha_i} \|\alpha_i\|_1, \quad s.t. \quad x_i = B^i\alpha_i. \tag{1}$$

For each sample x_i, the global optimization aims at selecting as few basis functions as possible from a large basis pool, which consists of all the other samples (basis), to linearly reconstruct x_i, meanwhile keeping the reconstruction error as small as possible. Due to an overcomplete or sufficient basis pool, similar samples can be encoded as totally different sparse codes, which may bring about the loss of locality information of the samples to be encoded. To preserve such locality information, many researches add one or several regularization terms to the object Eq. 1 as in [5] [15] and etc. However, there is a lack of generality for these methods and the regularization-based approaches are, as widely known, very time consuming.

Here, we propose a much more general and concise approach, called Locality-Preserving $\mathcal{L}_1$-Graph (LOP-$\mathcal{L}_1$), by limiting the basis pool in a local neighborhood basis of the object sample. Our algorithm only uses the k nearest neighborhoods of the object sample as the basis pool, and the definition of the object function minimization is as follows:

Definition 1. *The minimizing object function of LOP-$\mathcal{L}_1$ is defined as:*

$$\min_{\alpha_i} \|\alpha_i\|_1, \quad s.t. \quad x_i = \Gamma^i\alpha_i, \tag{2}$$

where $\Gamma^i = [x_1^i, x_2^i, \cdots, x_k^i]$ is the k-nearest neighbors of x_i in the data set, with the constraint that all the elements in α_i are nonnegative.

The weights of edges in the LOP-$\mathcal{L}_1$ graph are obtained by seeking a nonnegative low-rank and sparse matrix that represents each data sample as a linear combination of its constrained neighborhood. The constructed graph can capture both the global mixture of subspaces structure (by the coding process) and the locally linear structure (by the sparseness brought by the constrained neighborhood) of the data, hence is both generative and discriminative. Furthermore, by introducing such a locality preserving constraint to the sparse coding process, the similarity of sparse codes between similar local samples can be preserved. Therefore, the robustness of the subsequent data analysis task (e.g. spectral clustering) is enhanced. Limiting the size of basis pool also leads to a benefit of reducing the running time of $\mathcal{L}_1$-graph construction.

The details of our proposed LOP-$\mathcal{L}_1$ is described in Algorithm 3. It is worth to point out that our proposed LOP-$\mathcal{L}_1$ doesn't prevent users to add specific regularization terms during the optimization for a special application.

Algorithm 3: LOP-$\mathcal{L}_1$-Graph

Input : Data samples $X = [x_1, x_2, \cdots, x_N]$, where $x_i \in \mathbb{R}^m$; Parameter t for scaling k-nearest neighborhood, where $k = t * m$ (check Section 5.1 for more details).
Output: Adjacency matrix W of $\mathcal{L}_1$ graph.

1 Normalize the data sample x_i with $\|x_i\|_2 = 1$;
2 **for** $x_i \in X$ **do**
3 Find k-nearest neighbors of x_i:$\Gamma^i = [x_1^i, \cdots, x_k^i]$;
4 Let $B^i = [\Gamma^i, I]$;
5 Solve: $\min_{\alpha_i} \|\alpha_i\|_1, \quad s.t. \quad x_i = B^i\alpha_i$;
6 **end**
7 **for** $i = 1 : N$ **do**
8 **for** $j = 1 : N$ **do**
  `/* get the sparse code for each` x_i `*/`
9 **if** $x_j \in \Gamma^i$ **then**
  `/*` $pos(x_j)$ `is the position of` x_j `in` nb^i `*/`
10 $W(i,j) = \alpha_i(pos(x_j))$
11 **else**
12 $W(i,j) = 0$
13 **end**
14 **end**
15 **end**

In our implementation, we select one gradient-project-based method called *truncated Newton interior-point method* (TNIPM) [9] as the $\mathcal{L}_1$ minimization solver, which has $O(N^{1.2})$ empirical complexity where N is the number of samples. The $\mathcal{L}_1$-minimization object function we used is:

$$\operatorname{argmin}_x \|Ax - b\|^2 + \lambda\|x\|_1, \tag{3}$$

where λ is the Lasso penalty parameter. We choose $\lambda = 1$ in our experiments as many methods also choose.

Analysis of Time Complexity. Here we analyze the time efficiency of LOP-$\mathcal{L}_1$ by comparing its running time with classic $\mathcal{L}_1$-graph. $\mathcal{L}_1$-graph with TNIPM solver has $O(N^{1.2})$ [10] empirical complexity. Our LOP-$\mathcal{L}_1$ algorithm reduces the size of basis pool from N to $k = t * m$, so the empirical complexity will be $O(Nk^{1.2})$. To demonstrate the

time reduction, we test the CPU time of LOP-$\mathcal{L}_1$ and (classic) $\mathcal{L}_1$ over a series of random data sets which have 50 attributes and sample size from 10^1 to 10^4. The result is presented in Fig.2, which shows our proposed LOP-$\mathcal{L}_1$ has much better scalability.

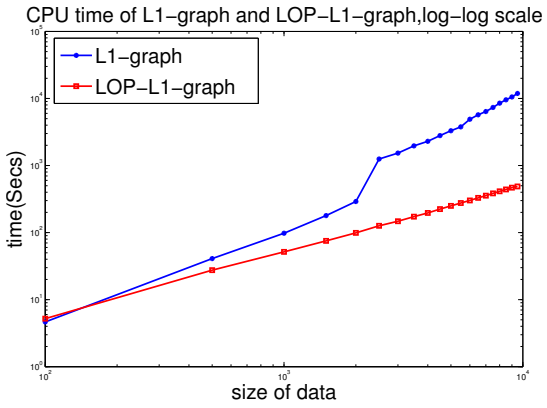


Figure 2: Scalability comparison between LOP-$\mathcal{L}_1$-graph and classic $\mathcal{L}_1$-graph.

4. ALGORITHM ANALYSIS AND CONNECTIONS

We now justify the LOP-$\mathcal{L}_1$ utility by briefly documenting its theoretic connections with a few existing methods, which also lays a solid foundation for LOP-$\mathcal{L}_1$'s attractive properties in practical use.

LOP-$\mathcal{L}_1$ vs Classic kNN-Graph. Compared with our proposed LOP-$\mathcal{L}_1$, the classic kNN graph [7] can be generated very fast, but they achieve this with a sacrifice on the quality. Classic kNN-graph-based methods can be easily affected by noises, especially those samples which are not in the same structure while being very close in the misleading high-dimensional Euclidean space. The fundamental difference between classic kNN graph and our proposed LOP-$\mathcal{L}_1$ is that the former is highly dependent on the pre-specified sample-sample similarity measure used to identify the neighbors, whereas the later generates an advanced similarity matrix W by solving the optimization problem of Equation 3. In this way, W can potentially encode rich and subtle relations across instances that may not be easily captured by conventional similarity metrics. This is validated by the experimental results in Section 5 that show the LOP-$\mathcal{L}_1$ substantially outperforms classic kNN graph in clustering application.

LOP-$\mathcal{L}_1$ vs Classic $\mathcal{L}_1$-Graph. Our proposed LOP-$\mathcal{L}_1$ is built upon classic $\mathcal{L}_1$, but has unique theoretical contributions and huge improvement on performance. As we mentioned earlier, the coding process of $\mathcal{L}_1$ suffers from the "overcomplete" basis pool. The optimization of $\mathcal{L}_1$ is solved by a straightforward numerical solution: every time the $\mathcal{L}_1$-minimization picks up the basis randomly from a group of "highly similar data samples" [17]. However, if the sample dimension is high, the similarity evaluation on Euclidean space would be highly misleading, which is a well-known problem. Therefore, together with a large-size basis pool, the basis $\mathcal{L}_1$ picks up are not guaranteed to be in the same manifold with the object sample. In our proposed LOP-$\mathcal{L}_1$, we restrain the coding process from picking up those samples outside certain neighborhood. In other words, the samples/basis are locally coded, and LOP-$\mathcal{L}_1$ brings a dramatic improvement of performance and stability on the subsequent analysis step. We will further confirm this in the Experiment Section 5.

LOP-$\mathcal{L}_1$ vs Regularization-based $\mathcal{L}_1$-Graph. Specifically, the idea of our LOP-$\mathcal{L}_1$ is close to the *kNN Fused Lasso graph* proposed by Zhou et al. [15]. However, our algorithm is different at: (1) there is no regularization term in our $\mathcal{L}_1$ minimization; (2) we process the data samples at original data space instead of at kernel feature space. Generally speaking, our LOP-$\mathcal{L}_1$ is designed in a more concise and efficient way compared with the regularization-based techniques such as [5] [15].

LOP-$\mathcal{L}_1$ vs Recommender Systems and Collaborative Filtering. Similar to the linear coding used in our proposed LOP-$\mathcal{L}_1$, Paterek [11] introduced a recommender system that linearly models each item for rating prediction, in which the rating of a user u_j on an item v_k is calculated as the aggregation of the ratings of u_j on all similar items (given by kNN graph). Intuitively, in our LOP-$\mathcal{L}_1$ we can treat $W(i,j)$ as a rating of sample x_i to sample x_j, which is derived by a subset of x_i's nearest neighbors, and prediction of $W(i,j)$ is generated based on a weighted aggregate of their ratings. In other words, LOP-$\mathcal{L}_1$ realizes the concept of collaborative filtering [11] within a constraint neighborhood that brings locality-preserving property, of which advantages in recommender systems has been analyzed and confirmed in [4].

5. EXPERIMENTS

5.1 Experiment Setup

Dataset. To demonstrate the performance of our proposed LOP-$\mathcal{L}_1$, we evaluate our algorithm on seven UCI benchmark datasets including three biological data sets (Breast Tissue(BT), Iris, Soybean), two vision image data set (Vehicle, Image,) and one chemistry data set (Wine) and one physical data set (Glass), whose statistics are summarized in Table 1. All these data sets have been popularly used in spectral clustering analysis research. These diverse combination of data sets are intended for our comprehensive studies.

Name	#samples	#attributes	#clusters
Iris	150	4	3
BT	106	9	6
Wine	178	13	3
Glass	214	9	6
Soybean	307	35	19
Vehicle	846	18	4
Image	2000	19	7

Table 1: Data Sets Statistics.

Baseline. To investigate the quality of the generated LOP-$\mathcal{L}_1$ graph, we compare its performance on spectral clustering applications with $\mathcal{L}_1$ graph. At the sample time, we also select a full-scale Gaussian similarity graph (Gaussian-graph), and a kNN Gaussian similarly graph (kNN-graph) as our competitors to understand the quality of LOP-$\mathcal{L}_1$ graph better. Since we have ground truth of labels for each data, we evaluate the spectral clustering performance with Normalized Mutual Information (NMI) and Accuracy (AC).

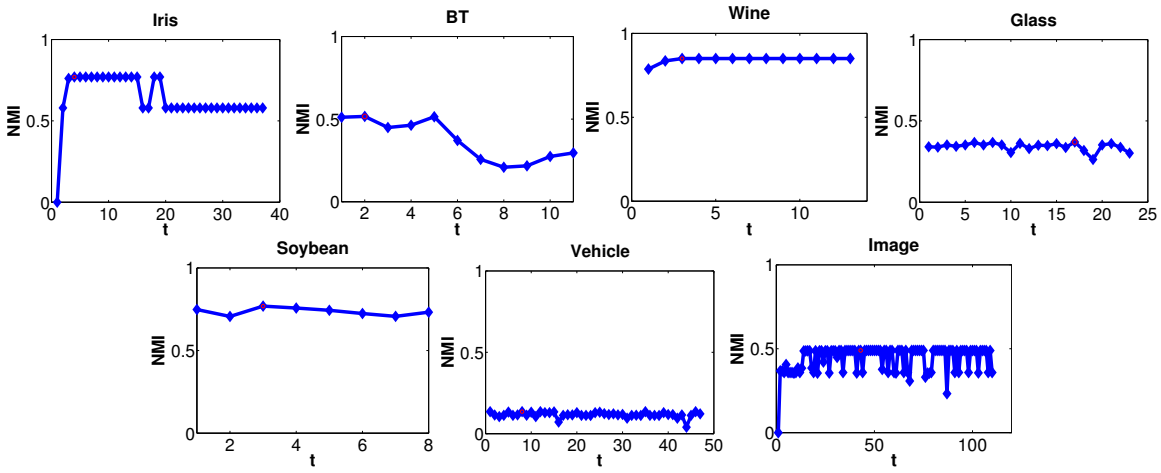


Figure 3: The change of NMI values w.r.t different selection of parameter t. Red dot in each subplot represent the maximal NMI. These experiments confirm that a basis neighborhood with certain size (with smaller t) provides better (or at least similar) performance than the overcomplete basis pool (with the maximal t in each subplot).

Parameter Setting. Our algorithm has one parameter named as *basis pool scaling parameter* t. It controls how many neighborhoods should be selected to the basis pool for each sample coding. We set t as a multiple value of attribute (or features) size w.r.t the data set.

$$2 \leq t \leq \frac{N}{m}, \tag{4}$$

where N is the number of samples and m is the sample dimensions. The reason we scale kNN neighborhood with Eq.4 is that we want to make it more adaptive to different context. In our experiments, we assign $t = 2, 3, 4$ and report the clustering performance results respectively. We will further analyze our selection of t in Section 5.3.

For Gaussian graph, the scaling parameter σ is configured as $\sigma = 0.1, 0.5, 1.0$. For kNN graph, we assign value of k as the size of basis pool of LOP-$\mathcal{L}_1$ graph with different t setting respectively. To obtain a fair comparison, we apply the same spectral clustering in Algorithm 2 to measure their performance.

5.2 Algorithm Performance Comparison

In this section, we evaluate our proposed LOP-$\mathcal{L}_1$-graph algorithm and other three graph construction algorithms. Table 2 and Table 3 document the comparison results (in NMI and AC) of clustering performance.

LOP-$\mathcal{L}_1$-graph vs $\mathcal{L}_1$-Graph. LOP-$\mathcal{L}_1$-graph has better average performance than $\mathcal{L}_1$-graph. LOP-$\mathcal{L}_1$-graph has average NMI value 0.5032 and AC value 0.5852 while $\mathcal{L}_1$-graph has average NMI value 0.4611 and AC value 0.5643. For each specific data set, the clustering performance of $\mathcal{L}_1$ graph beats average performance of LOP-$\mathcal{L}_1$-graph on Iris, BT, Image but lose on others. Moreover, we observe that the highest NMI value between them occurs at a specific t value of LOP-$\mathcal{L}_1$-graph, for example, the highest NMI values of Image data set is at $t = 2, 3$ of LOP-$\mathcal{L}_1$-graph.

LOP-$\mathcal{L}_1$-graph vs kNN-Graph. The average clustering performance of kNN-graph is the lowest one among Spectral Clustering with Gaussian similarity graph, $\mathcal{L}_1$-graph and LOP-$\mathcal{L}_1$-graph. Comparing to LOP-$\mathcal{L}_1$-graph, kNN-graph only have better performance (NMI: 0.4739, AC: 0.5346) than LOP-$\mathcal{L}_1$-graph (NMI: 0.4328, AC: 0.5189) on BT data set.

LOP-$\mathcal{L}_1$-graph vs Gaussian Similarity Graph. The spectral clustering with Gaussian similarity graph (fully connected graph) has lower average performance than LOP-$\mathcal{L}_1$-graph in our experiments. However, for specific data set, the maximum values of NMI and AC not always belong to LOP-$\mathcal{L}_1$-graph. For example, the highest NMI value for Iris data set is Gaussian similarity graph with $\sigma = 0.1$. The reason is that the spectral clustering based on Gaussian similarity graph is parameter sensitive. To obtain the best result, the user has to tune the parameter σ.

5.3 Analysis of Basis Pool Scaling

In our algorithm we argue that a constrained neighborhood as basis pool is not only enough but also provide locality property for the $\mathcal{L}_1$-graph construction. On the other hand, one of the most serious problem for kNN-based method is the over-sparsity where each sample only has a small amount of connected neighbors, which often results in that the derived graph is bias to some closely-connected "cliques" and the subsequent analysis is therefore unreliable.

We confirm the effectiveness of our strategy by recording the trend of NMI value with increasing size of t (up to the maximal t w.r.t each dataset) in Fig. 3 across different dataset. It once again confirms that we don't need all remain samples as the basis pool to construct an informative yet stable $\mathcal{L}_1$ graph.

6. CONCLUSION

Classic $\mathcal{L}_1$-graph exhibits good performance in many data mining applications. However, due to the overcomplete basis and the following lack of coding focus, the locality and the similarity among the samples to be encoded are lost. To preserve locality, sparsity and good performance in a concise and efficient way, we propose a Locality-Preserving $\mathcal{L}_1$-graph (LOP-$\mathcal{L}_1$). By limiting the coding process in a local neighborhood to preserve localization and coding stability, our

Name	Gaussian-graph				kNN-graph				$\mathcal{L}_1$-graph	LOP-$\mathcal{L}_1$-graph			
	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 1.0$	Avg.	$t = 2$	$t = 3$	$t = 4$	Avg.		$t = 2$	$t = 3$	$t = 4$	Avg.
Iris	0.8640	0.5895	0.7384	0.7306	0.4831	0.5059	0.3139	0.4343	0.7523	0.5794	0.7608	0.7696	0.7033
BT	0.4933	0.4842	0.4691	0.4822	0.4731	0.5335	0.4150	0.4739	0.3660	0.3912	0.4536	0.4536	0.4328
Wine	0.4540	0.7042	0.6214	0.5932	0.6647	0.7471	0.7031	0.7050	0.6537	0.8358	0.8500	0.8500	0.8453
Glass	0.3535	0.2931	0.3289	0.3252	0.2584	0.3475	0.3114	0.3058	0.3416	0.3533	0.3575	0.2988	0.3369
Soybean	0.6294	0.6814	0.6170	0.6426	0.6291	0.6120	0.5835	0.6082	0.7004	0.7265	0.7180	0.7267	0.7237
Vehicle	0.1248	0.0976	0.0958	0.1061	0.1101	0.0779	0.0667	0.0849	0.0726	0.1352	0.1019	0.1106	0.1159
Image	0.4800	0.4678	0.4740	0.4739	0.3256	0.4434	0.4548	0.4079	0.3410	0.3678	0.3678	0.3582	0.3646
Average				0.4791				0.4314	0.4611				0.5032

Table 2: NMI Comparison of LOP-$\mathcal{L}_1$-graph and other three graph construction methods.

Name	Gaussian-graph				kNN-graph				$\mathcal{L}_1$-graph	LOP-$\mathcal{L}_1$-graph			
	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 1.0$	Avg.	$t = 2$	$t = 3$	$t = 4$	Avg.		$t = 2$	$t = 3$	$t = 4$	Avg.
Iris	0.9600	0.7267	0.8600	0.8489	0.7533	0.6670	0.5800	0.6668	0.8867	0.6400	0.9933	0.9000	0.8111
BT	0.5472	0.4906	0.5189	0.5189	0.4717	0.6038	0.5283	0.5346	0.4434	0.4623	0.5472	0.5472	0.5189
Wine	0.6292	0.8876	0.8820	0.7996	0.8483	0.9101	0.9101	0.8895	0.8652	0.9551	0.9607	0.9607	0.9588
Glass	0.4112	0.3972	0.4299	0.4128	0.4299	0.5000	0.4860	0.4720	0.4579	0.4673	0.4907	0.4299	0.4626
Soybean	0.5081	0.5668	0.4300	0.5016	0.5049	0.4853	0.5016	0.4973	0.5244	0.5700	0.5668	0.6059	0.5809
Vehicle	0.3818	0.3582	0.3605	0.3668	0.3806	0.3475	0.3381	0.3554	0.3771	0.3936	0.3593	0.3676	0.3735
Image	0.5467	0.5124	0.5076	0.5222	0.4600	0.4838	0.4781	0.4740	0.3952	0.3919	0.3919	0.3881	0.3906
Average				0.5673				0.5556	0.5643				0.5852

Table 3: Accuracy Comparison of LOP-$\mathcal{L}_1$-graph and other three graph construction methods.

proposed LOP-$\mathcal{L}_1$ alleviates the instability of sparse codes and outperforms the existing works. We apply our proposed method on clustering application and the experiment result confirm the effectiveness of our proposed method.

7. ACKNOWLEDGMENT

This research is supported in part by National Science Foundation of USA (No. IIS-0949467, IIS-1047715, and IIS-1049448), and National Natural Science Foundation of China (No. 61190120, 61190121, and 61190125). It is also supported by United States Department of Energy, Grant No. DE-SC0003361, funded through the American Recovery and Reinvestment Act of 2009.

8. REFERENCES

[1] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang. Learning with l1-graph for image analysis. *IEEE Trans. of Image Processing*, 19(4):858–866, 2010.

[2] C. D. Correa and P. Lindstrom. Locally-scaled spectral clustering using empty region graphs. In *SIGKDD*, pages 1330–1338. ACM, 2012.

[3] B. Dai, X. Wu, et al. Graph-based learning via auto-grouped sparse regularization and kernelized extension. *TKDE*, page 1, 2014.

[4] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pages 107–144, 2011.

[5] Y. Fang, R. Wang, B. Dai, and X. Wu. Graph-based learning via auto-grouped sparse regularization and kernelized extension. 2013.

[6] S. Gao, I.-H. Tsang, and L.-T. Chia. Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. *PAMI*, 35(1):92–104, 2013.

[7] P. Hall, B. U. P. BU, and R. J. Samworth. Choice of neighbor order in nearest-neighbor classification. *Annals of Statistics*, 36(5):2135–2152, 2008.

[8] H. Huang, H. Qin, S. Yoo, and D. Yu. Local anomaly descriptor: a robust unsupervised algorithm for anomaly detection based on diffusion space. *ACM CIKM*, pages 405–414, 2012.

[9] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l 1-regularized least squares. *Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[10] K. Koh, S.-J. Kim, and S. P. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine learning research*, 8(8):1519–1555, 2007.

[11] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDD cup and workshop*, pages 5–8. ACM, 2007.

[12] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009.

[13] S. Yan and H. Wang. Semi-supervised learning by sparse representation. In *SDM*, pages 792–801. SIAM, 2009.

[14] Y. Yang, Z. Wang, J. Yang, J. Wang, S. Chang, and T. S. Huang. Data clustering by laplacian regularized l1-graph. In *AAAI*, 2014.

[15] G. Zhou, Z. Lu, and Y. Peng. L1-graph construction using structured sparsity. *Neurocomputing*, 120:441–452, 2013.

[16] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu. Non-negative low rank and sparse graph for semi-supervised learning. In *CVPR*, pages 2328–2335. IEEE, 2012.

[17] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.