Semi-Supervised Learning with Manifold Fitted Graphs

Tongtao Zhang[†]
[‡]Xiamen University, China
rrji@xmu.edu.cn

Rongrong Ji[†‡]
[†]Columbia University, USA
tz2163@caa.columbia.edu

Wei Liu[♮]
[♮]IBM Research, USA
weiliu@us.ibm.com

Dacheng Tao[§]
[§]University of Technology
Sydney, Australia
dacheng.tao@uts.edu.au

Gang Hua[♮♯]
[♯]Stevens Inst. of Technology, USA
ghua@stevens.edu

Abstract

In this paper, we propose a locality-constrained and sparsity-encouraged manifold fitting approach, aiming at capturing the locally sparse manifold structure into neighborhood graph construction by exploiting a principled optimization model. The proposed model formulates neighborhood graph construction as a sparse coding problem with the locality constraint, therefore achieving simultaneous neighbor selection and edge weight optimization. The core idea underlying our model is to perform a sparse manifold fitting task for each data point so that close-by points lying on the same local manifold are automatically chosen to connect and meanwhile the connection weights are acquired by simple geometric reconstruction. We term the novel neighborhood graph generated by our proposed optimization model $\mathcal{M}$*-Fitted Graph* since such a graph stems from sparse manifold fitting. To evaluate the robustness and effectiveness of $\mathcal{M}$-fitted graphs, we leverage graph-based semi-supervised learning as the testbed. Extensive experiments carried out on six benchmark datasets validate that the proposed $\mathcal{M}$-fitted graph is superior to state-of-the-art neighborhood graphs in terms of classification accuracy using popular graph-based semi-supervised learning methods.

1 Introduction

In this paper, we investigate the problem of *Graph-Based Semi-Supervised Learning* [Zhu and Goldberg, 2009] which is an emerging machine learning topic having received broad research attention and also triggered a variety of practical applications including document and image ranking [Zhou *et al.*, 2003b], web-scale image search [Jing and Baluja, 2008][Liu *et al.*, 2011a], image and video annotation [Jiang *et al.*, 2012], protein classification [Weston *et al.*, 2003], etc. To accomplish sensible semi-supervised learning, the key challenge arises, *i.e.*, how to build a robust neighborhood graph which is capable of capturing the inherent manifold structure underlying input data samples. After constructing such a graph, representative semi-supervised learning methods, *e.g.*, [Zhu *et al.*, 2003][Zhou *et al.*, 2003a][Belkin *et al.*, 2006][Liu and Chang, 2009], can be readily deployed through engaging this graph in proper smoothness regularization terms. A typical kind of graphs are kNN graphs which have been intensively used in a number of learning problems such as dimensionality reduction [Belkin and Niyogi, 2003] and spectral clustering [Ng *et al.*, 2001] besides semi-supervised learning studied in this paper. To construct a kNN graph on an input dataset $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$ with cardinality $|\mathcal{X}| = n$, edges are established to connect each individual data point and its k nearest neighbors. The edge weights are then obtained by either parametric kernel functions (typically the Gaussian kernel) or nonparametric optimization procedures [Jebara *et al.*, 2009]. However, kNN graphs do not explicitly explore or capture intrinsically low-dimensional manifolds on which high-dimensional data samples reside, though they can approximate some manifolds existing in simple datasets.

Towards capturing geometric structure hidden in data into graph construction mechanisms, one recent trend is to explore the sparse subspace structure [Elhamifar and Vidal, 2009][Elhamifar and Vidal, 2010][Elhamifar and Vidal, 2011][Gao *et al.*, 2010], which has turned out to be very effective for discovering the clusters formed in high-dimensional data such as face images. Specifically, [Elhamifar and Vidal, 2009][Elhamifar and Vidal, 2010] proposed to reconstruct each individual data point $\boldsymbol{x}_i$ using the rest of points in $\mathcal{X}$ in a least squares sense. Very importantly, for each instance $\boldsymbol{x}_i$ a sparsity constraint is imposed to make such a geometric reconstruction find the most fitted subspace spanned by a few instances in $\mathcal{X}$. In another words, the found subspace holds the shortest distance to the point $\boldsymbol{x}_i$ meanwhile keeping the subspace dimension as low as possible. This sparse subspace fitting problem can be relaxed to an ℓ_1 minimization problem [Candés and Tao, 2005] and thus solved by conventional convex optimization techniques. In addition, the seminal work [Gao *et al.*, 2010] and [Elhamifar and Vidal, 2011] suggested similar geometric reconstruction schemes in kernel-induced feature spaces and on manifolds, respectively. In the work, the sparsity constraint is enforced as well.

The key computational challenge of the aforementioned sparse subspace fitting scheme [Elhamifar and Vidal, 2009]

and its variants is that they require all, or at least a part, of the input dataset $\mathcal{X}$ to reconstruct *every* data point $\boldsymbol{x}_i$, that is, minimizing an ℓ_1 problem ($\|\cdot\|_1$ denotes ℓ_1 norm) of $n-1$ variables

$$\min_{\boldsymbol{a}_i \in \mathbb{R}^{n-1}} \|\boldsymbol{a}_i\|_1 \quad \text{s.t.} \quad \boldsymbol{x}_i = \mathbf{X}_{\bar{i}}\boldsymbol{a}_i. \tag{1}$$

In reconstructing $\boldsymbol{x}_i$, this ℓ_1 problem involves a basis $\mathbf{X}_{\bar{i}} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \cdots, \boldsymbol{x}_n]$ that is composed of $n-1$ instances in $\mathcal{X}$ except the target instance $\boldsymbol{x}_i$. Above all, the convex relaxation from ℓ_0 norm to ℓ_1 norm cannot guarantee that the resulting solution $\boldsymbol{a}_i$ (*i.e.*, the subspace reconstruction coefficient vector) is always meaningful. The ℓ_1 minimization may lead to a dense $\mathbf{a}_i$ and thus violate the sparsity expectation of the subspace reconstruction coefficients. Second, solving the ℓ_1 problem in eq. (1) is nontrivial and time-consuming for high-dimensional or large-scale data. Third, the sparse subspace fitting scheme neglects the locality of data and is hence likely to introduce irrelevant instances that are outliers or from other classes with respect to $\boldsymbol{x}_i$. Finally and most critically, sparse subspace fitting assumes that a *multi-subspace* structure exists in the data, which is not always true for real-world data. It is noted that [Cheng *et al.*, 2009] considered the locality constraint when performing sparse subspace fitting for neighborhood graph construction, but it still adopted the multi-subspace assumption.

In this paper, we argue that a *multi-manifold* structure is more natural than the multi-subspace structure for realistic datasets. The latter can be regarded as an extreme case of the multi-manifold structure.

In order to overcome the potential issues associated with sparse subspace fitting, in this work we propose a locality-constrained and sparsity-encouraged reconstruction approach, namely *Sparse Manifold Fitting*, which can capture the locally sparse manifold structure and thus discover the global multi-manifold structure. The proposed approach yields a novel neighborhood graph called by *$\mathcal{M}$-Fitted Graph*. In sparse manifold fitting, we directly design an ℓ_0 norm-based optimization model to perform sparse coding with the locality constraint that restricts the coding within the scope of a limited number of neighbors, thereby ensuring the optimization efficiency.

Our $\mathcal{M}$-fitted graphs can benefit vast applications which previously used kNN graphs. To validate the robustness and effectiveness of $\mathcal{M}$-fitted graphs, we adopt graph-based semi-supervised learning as the testbed since its classification accuracy heavily depends on the quality of used graphs. Through extensive experiments, we find that our $\mathcal{M}$-fitted graphs are very suitable for semi-supervised learning and exhibit superior performance over state-of-the-art neighborhood graphs when cooperating with representative semi-supervised learning techniques, *e.g.*, *Local and Global Consistency* (LGC) [Zhou *et al.*, 2003a], *Linear Neighborhood Propagation* (LNP) [Wang and Zhang, 2008], and *Gaussian Fields and Harmonic Functions* (GFHF) [Zhu *et al.*, 2003].

The remainder of this paper is organized as follows: we first introduce $\mathcal{M}$-fitted graphs, subsequently present the sparse manifold fitting approach for constructing $\mathcal{M}$-fitted graphs, then illustrate the usage in the context of graph-based semi-supervised learning, and finally show the quantitative experimental results.

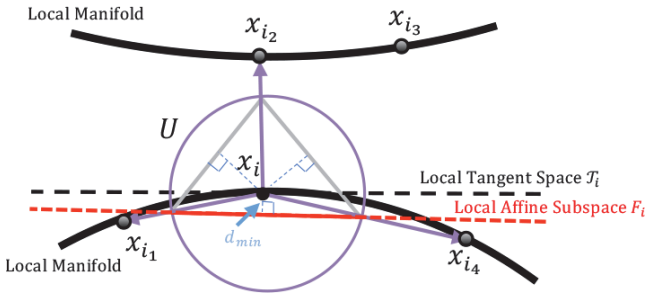


Figure 1: The best fitted local affine subspace $\mathbf{F}_i$ (red dashed line) consists of two local directions $\frac{\boldsymbol{x}_{i_1}-\boldsymbol{x}_i}{\|\boldsymbol{x}_{i_1}-\boldsymbol{x}_i\|}$ and $\frac{\boldsymbol{x}_{i_4}-\boldsymbol{x}_i}{\|\boldsymbol{x}_{i_4}-\boldsymbol{x}_i\|}$, and holds the minimal distance to $\boldsymbol{x}_i$.

2 $\mathcal{M}$-Fitted Graphs

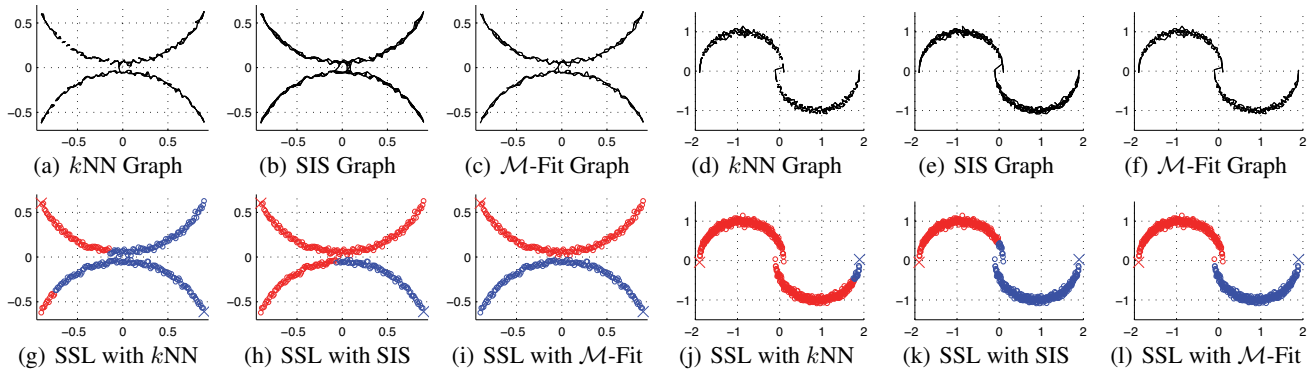
Notations. We define the input data matrix as $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \subset \mathbb{R}^{d\times n}$ of d feature dimensions and n instances. We formulate the neighborhood graph to be constructed as an affinity matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ where w_{ij} represents the edge weight between vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We define by $\mathcal{M}$ the manifold(s) underlying the input dataset $\mathcal{X}$, $\{\mathcal{T}_i\}_{i=1}^n$ the local tangent spaces, $\{\mathbf{U}_i\}_{i=1}^n$ the local direction basis matrices, and $\{\mathbf{F}_i\}_{i=1}^n$ the local affine subspaces.

Geometric Idea. Following the well-known manifold learning approach *Local Tangent Space Alignment* [Zhang and Zha, 2004], we assume that the manifold(s) (possibly multiple manifolds) underlying the input data collection $\mathcal{X}$ are composed of and merged by a series of *local tangent spaces*. Each tangent space, denoted by $\mathcal{T}_i$, lies under each data point $\boldsymbol{x}_i$. In mathematics, we can write $\mathcal{M} = \bigcup_{i=1}^n \mathcal{T}_i$ if $n \to \infty$, where a local tangent space $\mathcal{T}_i$ is actually a small patch of a local manifold from $\mathcal{M}$. Then our goal is to seek such a tangent space located at each point. We employ the geometric sparsity idea proposed in [Elhamifar and Vidal, 2011] to approximately seek the local tangent spaces. Specifically, we reconstruct each data point $\boldsymbol{x}_i$ using sparse bases from a local direction basis $\mathbf{U}_i$, and the minimal reconstruction distortion is pursued to yield the optimal local directions that determine the optimally fitted local affine subspace $\mathbf{F}_i$ for $\boldsymbol{x}_i$. As a result, this sparse reconstruction gives rise to the local affine subspace $\mathbf{F}_i$ with the minimal *angle* to the target local tangent space $\mathcal{T}_i$, therefore discovering the best fitted local manifold.

Through sparse reconstruction, we can build a neighborhood graph in which the graph edges correspond to the solved local directions. Under such a circumstance, the edges connecting to $\boldsymbol{x}_i$ are closely parallel to the local tangent space $\mathcal{T}_i$ and thus closely reside on the local manifold at $\boldsymbol{x}_i$.

Formulation. Our proposed sparse reconstruction starts from setting up a relatively large neighborhood of m ($m < n$) neighbors (record $\boldsymbol{x}_i$'s neighbor indices into $[i_1, ..., i_m]$), and then eliminates the unqualified ones.

We normalize the difference vectors between $\mathbf{x}_i$ and its m neighbors to remove distance variations while preserving



(a) kNN Graph (b) SIS Graph (c) $\mathcal{M}$-Fit Graph (d) kNN Graph (e) SIS Graph (f) $\mathcal{M}$-Fit Graph

(g) SSL with kNN (h) SSL with SIS (i) SSL with $\mathcal{M}$-Fit (j) SSL with kNN (k) SSL with SIS (l) SSL with $\mathcal{M}$-Fit

Figure 2: Graph-based semi-supervised learning on synthetic datasets. Points on the same semi-circle are considered to take the same class label. The big $\times$ denotes initially labeled points (the blue color represents the positive label, while the red color represents the negative label). The upper row visualizes the edge connections in kNN graphs, SIS graphs, and $\mathcal{M}$-fitted graphs. We set $k = 3$ for kNN graphs and $m = 6, s = 3$ for our $\mathcal{M}$-fitted graphs, respectively. The lower row shows the classification results of the semi-supervised learning method GFHF using different graphs.

structural cues, and then form a local direction basis matrix

$$\mathbf{U}_i = \left[\frac{\boldsymbol{x}_{i_1} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{i_1} - \boldsymbol{x}_i\|}, \ldots, \frac{\boldsymbol{x}_{i_m} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{i_m} - \boldsymbol{x}_i\|} \right]. \tag{2}$$

Now, our task is to find the local affine subspace $\mathbf{F}_i$ that fits $\boldsymbol{x}_i$ as much as possible. As intuitively illustrated in Fig. 1, we pursue the minimal distance between $\mathbf{x}_i$ and the ambient affine subspace

$$\mathbb{F}_i = \{\boldsymbol{x}_i + \mathbf{U}_i \boldsymbol{c}_i | \mathbf{1}^\top \boldsymbol{c}_i = 1, \boldsymbol{c}_i \in \mathbb{R}^m\}. \tag{3}$$

That is, $\min_{\mathbf{1}^\top \boldsymbol{c}_i = 1} \|\boldsymbol{x}_i - \boldsymbol{x}_i - \mathbf{U}_i \boldsymbol{c}_i\| = \|\mathbf{U}_i \boldsymbol{c}_i\|$. Critically, a sparsity constraint $\|\mathbf{c}_i\|_0 \leq s$ ($s < m$) is imposed to reduce the risk of introducing outliers. In doing so, the nonzero elements in the reconstruction coefficient vector $\boldsymbol{c}_i$ automatically select sparse local directions from $\mathbf{U}_i$ to generate a feasible local affine subspace. So far, we formulate the sparse reconstruction task as follows

$$\begin{aligned} \min_{\boldsymbol{c}_i \in \mathbb{R}^m} \; & Q(\boldsymbol{c}_i) = \|\mathbf{U}_i \boldsymbol{c}_i\|^2 \\ \text{s.t.} \; & \mathbf{1}^\top \boldsymbol{c}_i = 1, \; \|\boldsymbol{c}_i\|_0 \leq s, \end{aligned} \tag{4}$$

whose core idea is to seek the best fitted local affine subspace $\mathbf{F}_i$ subject to the affine and sparsity constraints on the coefficient vector $\boldsymbol{c}_i$. Fig. 1 displays that the local affine subspace holding the minimal distance to the point $\boldsymbol{x}_i$ also holds the minimal angle to the local tangent space $\mathcal{T}_i$.

Graph Construction. The next step is to translate the coefficient vectors $\{\boldsymbol{c}_i\}_{i=1}^n$, solved in sparse reconstruction, into the graph edge weights $\{w_{ij}\}_{i,j=1}^n$. Let $\boldsymbol{c}_i = [c_{1i}, \cdots, c_{mi}]^\top$. For each vertex (*i.e.*, data point) $\boldsymbol{x}_i$, we obtain the edge weights $w_{i_j,i}$'s via re-normalizing c_{ji}'s:

$$w_{i_j,i} = \frac{\frac{c_{ji}}{\|\boldsymbol{x}_{i_j} - \boldsymbol{x}_i\|}}{\sum_{j'=1}^m \frac{c_{j'i}}{\|\boldsymbol{x}_{i_{j'}} - \boldsymbol{x}_i\|}}, \; j = 1, \ldots, m. \tag{5}$$

It is worth mentioning that although we eliminate the distance information in eq. (2), we take it back in eq. (5) to make the neighbors closer to $\boldsymbol{x}_i$ contribute larger edge weights.

Like [Cheng *et al.*, 2009], we deal with the negative weights as follows

$$\boldsymbol{w}_i \leftarrow \max(\boldsymbol{w}_i, \mathbf{0}), \; \boldsymbol{w}_i \leftarrow \frac{\boldsymbol{w}_i}{\mathbf{1}^\top \boldsymbol{w}_i}, \tag{6}$$

where $\boldsymbol{w}_i$ is the i-th column vector of the graph affinity matrix $\mathbf{W}$. Finally, we conduct a symmetrizing step

$$\mathbf{W} \leftarrow \frac{\mathbf{W} + \mathbf{W}^\top}{2}. \tag{7}$$

We term the graph of the affinity matrix $\mathbf{W}$ *$\mathcal{M}$-Fitted Graph*, which is yielded through the sparse reconstruction in eq. (4) and the graph construction steps in eqs. (5)(6)(7).

In contrast to *Sparse Manifold Clustering and Embedding* (SMCE) [Elhamifar and Vidal, 2011], our proposed $\mathcal{M}$-fitted graph not only eliminates the incorrect edges connecting points not lying on the same local manifold, but also keeps the sparse edge connections. The sparsity originates from directly solving an ℓ_0 problem (see Section 3) instead of the ℓ_1 problem in [Elhamifar and Vidal, 2011]. Note that in SMCE the edges connecting to some outliers are tolerable, as SMCE is designed for clustering which is somewhat insensitive to the sparsity level of the graph affinity matrix $\mathbf{W}$. However, in the scenario of graph-based semi-supervised learning, erroneous edge connections are unacceptable, so we discard negative reconstruction coefficients and enforce explicit sparsity (via ℓ_0 norm) in producing $\mathbf{W}$.

Fig. 2 visualizes the semi-supervised learning results produced by kNN graphs, *Sparsity Induced Similarity* (SIS) graphs [Cheng *et al.*, 2009], and $\mathcal{M}$-fitted graphs on two synthetic datasets. The results indicate that 1) kNN graphs lead to disconnected graphs, and that 2) SIS graphs are denser and include more erroneous edges than $\mathcal{M}$-fitted graphs. On the contrary, our $\mathcal{M}$-fitted graphs successfully achieve the balance between keeping connected graphs and eliminating incorrect edges.

3 Sparse Manifold Fitting

We propose an iterative optimization algorithm to solve the *Sparse Manifold Fitting*[1] problem with ℓ_0 norm formulated by eq. (4). The key idea is to progressively fit the local tangent space $\mathcal{T}$ by iteratively selecting s bases with the index (support) set $\mathcal{S} \subset [1:m]$ from the local direction basis $\mathbf{U}$. These sparse bases wrapped in $\mathbf{U}_{\mathcal{S}}$ will be refined such that the local affine subspace $\mathbf{F}$ spanned by $\mathbf{U}_{\mathcal{S}}$ has a shorter distance from the point $\boldsymbol{x}$, equivalently a smaller angle from $\mathcal{T}$. At each iteration we evaluate the current bases in $\mathbf{U}_{\mathcal{S}}$ and subsequently update them. Concretely, a more proper base $\mathbf{U}_j$ from $\mathbf{U}_{[1:m]\setminus\mathcal{S}}$ is augmented firstly, the most unreliable base in $\mathbf{U}_{\mathcal{S}}$ is removed later, and the support set $\mathcal{S}$ is updated accordingly. The above iterative procedure which consecutively updates the support set $\mathcal{S}$ is halted until the decrease of the raw objective function $Q(\boldsymbol{c})$ in eq. (4) stops or becomes insignificant. We call such an iterative algorithm as *Forward-Backward Fitting*.

Before presenting our algorithm, we provide some notations and derivations which are vital in clarifying the forward-backward fitting mechanism. Given a support set $\mathcal{S}$, we consider the following simpler problem

$$\begin{aligned} \min_{\boldsymbol{c}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}} \ & \|\mathbf{U}_{\mathcal{S}}\boldsymbol{c}_{\mathcal{S}}\|^2 \\ \text{s.t.} \ & \mathbf{1}^\top \boldsymbol{c}_{\mathcal{S}} = 1, \end{aligned} \tag{8}$$

which enjoys a closed-form solution

$$\boldsymbol{c}^\star_{\mathcal{S}} = g(\mathbf{U}, \mathcal{S}) = \frac{(\mathbf{U}_{\mathcal{S}}^\top \mathbf{U}_{\mathcal{S}})^{-1}\mathbf{1}}{\mathbf{1}^\top (\mathbf{U}_{\mathcal{S}}^\top \mathbf{U}_{\mathcal{S}})^{-1}\mathbf{1}}. \tag{9}$$

Eq. (9) is exploited to estimate $\boldsymbol{c}$ when we have a guess for the support set $\mathcal{S}$. Another key component that our algorithm uses is the hard-thresholding operator:

$$\begin{aligned} &\text{supp}_s(\boldsymbol{c}) \\ &= \{j \,||c_j| \text{ is among largest } s \text{ absolute values of elements of } \boldsymbol{c}\}, \end{aligned} \tag{10}$$

which prunes the elements with small absolute values in a vector $\boldsymbol{c}$ and returns the support set with cardinality s.

Forward-Backward Fitting Algorithm

The algorithm targeting for sparse manifold fitting is presented in Algorithm 1 and elaborated below.

Initialization. Before performing a forward selection step, an initialized support set $\mathcal{S}$ should be determined. To satisfy the locality constraint, $\boldsymbol{x}$'s s nearest neighbors may have a higher probability of forming a well fitted affine subspace. Hence, we start with the left first s vectors of $\mathbf{U}$ to initialize the support set $\mathcal{S} = [1:s]$, and then acquire the initial solution $\boldsymbol{c}^0_{\mathcal{S}}$ by means of eq. (9). In addition, we write the residue vector $\boldsymbol{r} = \boldsymbol{x} - (\boldsymbol{x} + \mathbf{U}_{\mathcal{S}}\boldsymbol{c}_{\mathcal{S}}) = -\mathbf{U}_{\mathcal{S}}\boldsymbol{c}_{\mathcal{S}}$ given the current support set $\mathcal{S}$ and the corresponding reconstruction coefficient vector $\boldsymbol{c}_{\mathcal{S}}$. Consequently, the current cost function is derived as $Q(\mathbf{c}) = \|\mathbf{U}_{\mathcal{S}}\boldsymbol{c}_{\mathcal{S}}\|^2 = \|\mathbf{r}\|^2$.

[1]Without loss of generality, in this section we ignore the subscript i when referring to $\boldsymbol{x}_i$, $\mathcal{T}_i$, $\mathbf{U}_i$, $\mathbf{F}_i$ and $\boldsymbol{c}_i$.

Algorithm 1: Forward-Backward Fitting Algorithm

Input: A local direction basis matrix $\mathbf{U} \in \mathbb{R}^{d\times m}$, a sparsity level s ($s < m$).
Output: The sparse reconstruction coefficient vector $\boldsymbol{c}^* \in \mathbb{R}^m$ with $\|\boldsymbol{c}^*\|_0 \le s$.
Initialize: $\mathcal{I} = [1:m]$, $\mathcal{S} = [1:s]$, $\boldsymbol{c}^0_{\mathcal{S}} = g(\mathbf{U}, \mathcal{S})$, $\boldsymbol{c}^0_{\mathcal{I}\setminus\mathcal{S}} = \mathbf{0}$, $\boldsymbol{r} = -\mathbf{U}_{\mathcal{S}}\boldsymbol{c}^0_{\mathcal{S}}$, $Q^0 = \|\boldsymbol{r}\|^2$, $\epsilon = 10^{-6}$, $t = 0$.
while (true) **do**
  forward step:
  $\boldsymbol{b}_{\mathcal{S}} = \mathbf{0}$, $\boldsymbol{b}_{\mathcal{I}\setminus\mathcal{S}} = \mathbf{U}^\top_{\mathcal{I}\setminus\mathcal{S}}\boldsymbol{r}$,
  $\mathcal{S} \leftarrow \text{supp}_1(\boldsymbol{b}) \cup \mathcal{S}$,
  $\boldsymbol{c}^{t+1}_{\mathcal{S}} \leftarrow g(\mathbf{U}, \mathcal{S})$, $\boldsymbol{c}^{t+1}_{\mathcal{I}\setminus\mathcal{S}} \leftarrow \mathbf{0}$;
  backward step:
  $\mathcal{S} \leftarrow \text{supp}_s(\boldsymbol{c}^{t+1})$;
  updating step:
  $\boldsymbol{c}^{t+1}_{\mathcal{S}} \leftarrow g(\mathbf{U}, \mathcal{S})$, $\boldsymbol{c}^{t+1}_{\mathcal{I}\setminus\mathcal{S}} \leftarrow \mathbf{0}$,
  $\boldsymbol{r} = -\mathbf{U}_{\mathcal{S}}\boldsymbol{c}^{t+1}_{\mathcal{S}}$, $Q^{t+1} \leftarrow \|\boldsymbol{r}\|^2$;
  if $(Q^{t+1} > Q^t)$ $\boldsymbol{c}^* \leftarrow \boldsymbol{c}^t$, **break**;
  elseif $(Q^t - Q^{t+1} \le \epsilon)$ $\boldsymbol{c}^* \leftarrow \boldsymbol{c}^{t+1}$, **break**;
  else $t \leftarrow t + 1$.
end

Forward Selection Step. We intend to select a new base from $\mathbf{U}_{[1:m]\setminus\mathcal{S}}$, which has the maximum correlation to the current residue vector $\boldsymbol{r}$. For a convenient expression, we define an indicator vector $\boldsymbol{b}$ as follows

$$\begin{cases} \boldsymbol{b}_{\mathcal{S}} = \mathbf{0}, \\ \boldsymbol{b}_{[1:m]\setminus\mathcal{S}} = \mathbf{U}^\top_{[1:m]\setminus\mathcal{S}}\boldsymbol{r}. \end{cases}$$

Afterwards, we augment the support set as

$$\mathcal{S} \leftarrow \text{supp}_1(\boldsymbol{b}) \cup \mathcal{S},$$

and immediately update $\boldsymbol{c}_{\mathcal{S}}$ by eq. (9). Note that after this step $\boldsymbol{c}_{\mathcal{S}} \in \mathbb{R}^{s+1}$.

Backward Pruning Step. We simply prune $\boldsymbol{c}_{\mathcal{S}} \in \mathbb{R}^{s+1}$ by applying the hard-thresholding operator in eq. (10), which amounts to updating $\mathcal{S} \leftarrow \text{supp}_s(\boldsymbol{c})$. After applying eq. (9) again with the updated support set, $\boldsymbol{c}_{\mathcal{S}}$ is reduced from $s+1$ dimensions to s dimensions.

Stopping Criterion. The presented iterative procedure integrating both forward and backward steps will be halted once the cost function Q increases or its decrease is not significant, that is,

$$Q(\boldsymbol{c}^{t+1}) > Q(\boldsymbol{c}^t) \quad \text{or} \quad 0 \le Q(\boldsymbol{c}^t) - Q(\boldsymbol{c}^{t+1}) \le \epsilon,$$

where $\epsilon > 0$ is a small constant.

Analysis

Our forward-backward fitting algorithm is specially devised to tackle the sparse reconstruction (or sparse coding) problem with ℓ_0 norm, namely sparse manifold fitting. In the sense of forward-backward exploring nonzero entries of a sparse solution, our algorithm is similar to the feature selection algorithm proposed in [Zhang, 2011]. Nonetheless, our algorithm

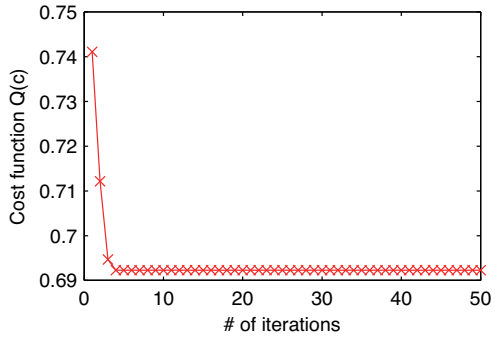


Figure 3: The cost function $Q(\boldsymbol{c}) = \|\mathbf{U}\boldsymbol{c}\|^2$ converges within five iterations for an image sample of the **USPS** dataset.

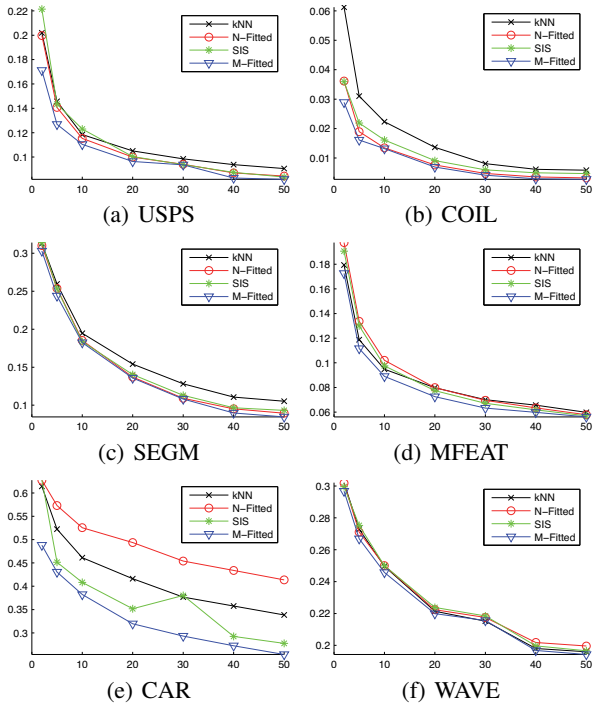
differs from [Zhang, 2011] in terms of augmenting and eliminating only one nonzero entry in each iteration, while the latter added and pruned multiple nonzero entries. The computational efficiency of sparse coding algorithms is especially crucial when being applied to neighborhood graph construction, since one needs to solve an ℓ_0 or ℓ_1 problem for each individual data point. Notably, our algorithm executing simple "push" and "pop" operations is substantially fast, as validated later by our experiments. The reason is that the proposed optimization algorithm driven by ℓ_0 norm may result in a faster convergence[2] than the optimization algorithms using ℓ_1 norm. Compared with the state-of-the-art graph construction methods [Wang and Zhang, 2008][Cheng *et al.*, 2009], the proposed $\mathcal{M}$-fitted graph construction through applying the forward-backward fitting algorithm runs faster. The speed of $\mathcal{M}$-fitted graphs is most comparable with that of kNN graphs.

4 Graph-Based Semi-Supervised Learning

We employ graph-based semi-supervised learning (GSSL) as the testbed for evaluating our proposed $\mathcal{M}$-fitted graph. In particular, we use three representative GSSL methods: Local and Global Consistency (LGC) [Zhou *et al.*, 2003a], Linear Neighborhood Propagation (LNP) [Wang and Zhang, 2008], and Gaussian Fields and Harmonic Functions (GFHF) [Zhu *et al.*, 2003]. The GFHF method also includes two versions: GFHF without postprocessing and GFHF with the postprocessing of Class Mass Normalization (CMN). Given a graph affinity matrix $\mathbf{W}$ as well as its degree matrix $\mathbf{D} = \mathrm{diag}(\mathbf{W1})$, LGC uses the symmetrically normalized graph Laplacian matrix $\hat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, LNP uses the asymmetrically normalized graph Laplacian matrix $\check{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$, and GFHF (two versions) uses the standard graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

5 Experiments

Datasets. We conduct our experiments on six benchmark datasets: (1) **USPS** [Hastie *et al.*, 2009] contains handwritten digital images from 0-9; (2) **COIL** [Nene *et al.*, 1996] includes 20 object classes, each of which contains 72 images; (3) **SEGM** [Frank and Asuncion, 2010] contains seven outdoor categories of images; (4) **MFEAT** [van Breukelen and Duin, 1998] contains 2,000 handwritten digital images from 0-9, which are extracted from the Dutch utility maps; (5) **CAR** [Bohanec and Rajkovič, 1988] contains 1,728 textual records for evaluating cars based on price, safety, comfort and maintenance, which are categorized into four classes; (6) **WAVE** [Breiman *et al.*, 1984] contains 5,000 textual instances sampled from three classes.



(a) USPS (b) COIL (c) SEGM (d) MFEAT (e) CAR (f) WAVE

Figure 4: Average error rate vs. l for GFHF+CMN.

Compared Graphs. We compare four graph construction approaches including kNN graphs, neighborhood-fitted ($\mathcal{N}$-fitted) graphs [Wang and Zhang, 2008], Sparsity Induced Similarity (SIS) graphs [Cheng *et al.*, 2009], and our proposed $\mathcal{M}$-fitted graphs. In kNN graphs, we use the Gaussian RBF kernel for weighting edges. For kNN and $\mathcal{N}$-fitted graphs, we empirically set the number of nearest neighbors to $k = 6$. To build SIS graphs, for each instance we select its $\frac{n}{4c}$ (c is the total number of classes) nearest neighbors to carry out ℓ_1 minimization. For simplicity, we set $m = \frac{n}{4c}$ and $s = 6$ to construct our $\mathcal{M}$-fitted graphs.

Quantitative Results. To set up semi-supervised learning (SSL) trials, we uniformly sample instances at random from each dataset, ensuring that every class covers at least one instance. We denote by l the average number of labeled instances per class. On each dataset, we conduct 50 SSL trials. The SSL outputs are the classification error rates, averaged over 50 trials, achieved by different combinations of graph construction and SSL methods. We also report the average runtime[3] that every combination of graph construction and

[2]Typically, our proposed algorithm converges within five iterations on most datasets. Due to the page limit, we only report the convergence curve on the **USPS** dataset, as shown in Fig. 3.

[3]All of our experiments are run on a workstation with INTEL

Table 1: Average error rate and runtime on **USPS**, **COIL** and **SEGM** datasets.

SSL Method	Graph	USPS			COIL			SEGM		
		$l=2$ Error/%	$l=5$ Error/%	Time/s	$l=2$ Error/%	$l=5$ Error/%	Time/s	$l=2$ Error/%	$l=5$ Error/%	Time/s
LGC	kNN	29.96	20.89	0.3580	9.65	4.91	1.5452	39.11	31.12	0.2785
	$\mathcal{N}$-Fitted	31.29	22.28	7.8536	9.04	4.45	26.3985	38.58	30.97	8.5385
	SIS	31.19	22.42	19.1036	8.18	5.89	281.5685	37.13	29.30	13.6618
	$\mathcal{M}$-Fitted	**29.80**	**21.20**	5.0395	**5.69**	**2.88**	14.8532	**36.33**	**28.81**	4.3460
LNP	kNN	24.53	17.96	0.7478	6.68	3.22	1.7054	33.11	27.36	0.7131
	$\mathcal{N}$-Fitted	22.35	16.63	8.2643	4.96	4.19	26.1052	37.26	28.17	6.4655
	SIS	**20.71**	16.51	19.4061	9.25	3.28	289.6741	31.62	26.10	9.3933
	$\mathcal{M}$-Fitted	**20.71**	**16.16**	3.6660	**4.68**	**2.91**	20.0269	**30.66**	**25.55**	3.3524
GFHF	kNN	38.05	25.02	0.3333	8.18	3.97	1.5627	36.31	28.71	0.2579
	$\mathcal{N}$-Fitted	38.65	22.77	8.4252	5.08	2.25	27.4328	38.55	28.66	7.1161
	SIS	45.50	28.72	13.3950	4.85	2.62	279.6863	37.53	28.48	7.3029
	$\mathcal{M}$-Fitted	**29.50**	**19.13**	3.8513	**3.77**	**1.97**	11.6737	**36.21**	**27.76**	3.5579
GFHF + CMN	kNN	20.19	14.58	0.3355	6.13	3.10	1.5509	31.28	25.96	0.2628
	$\mathcal{N}$-Fitted	19.97	14.07	8.5836	3.61	1.90	26.2992	31.01	25.38	7.2184
	SIS	22.15	14.35	15.8672	3.59	2.19	283.8451	31.41	25.33	6.9735
	$\mathcal{M}$-Fitted	**17.12**	**12.69**	3.3283	**2.89**	**1.61**	12.2562	**30.30**	**24.38**	3.4812

Table 2: Average error rate and runtime on **MFEAT**, **CAR** and **WAVE** datasets.

SSL Method	Graph	MFEAT			CAR			WAVE		
		$l=2$ Error/%	$l=5$ Error/%	Time/s	$l=20$ Error/%	$l=50$ Error/%	Time/s	$l=2$ Error %	$l=5$ Error/%	Time/s
LGC	kNN	25.50	16.70	0.4126	21.01	17.50	0.2589	34.12	30.67	1.6048
	$\mathcal{N}$-Fitted	27.53	18.20	11.2863	26.48	22.86	5.5986	32.59	28.59	34.0939
	SIS	25.26	16.98	40.2452	25.37	20.80	4.0573	33.27	29.69	32.8699
	$\mathcal{M}$-Fitted	**24.31**	**16.45**	8.9988	**18.93**	**13.77**	2.9807	**32.30**	**28.27**	26.8215
LNP	kNN	18.01	11.72	0.7811	32.96	28.02	0.4416	41.69	33.21	5.0517
	$\mathcal{N}$-Fitted	19.13	12.37	7.6312	36.69	29.77	4.9133	39.64	31.60	18.2306
	SIS	17.78	11.61	34.7508	30.07	26.32	4.5560	34.75	29.38	33.8275
	$\mathcal{M}$-Fitted	**16.99**	**11.53**	5.9624	**19.91**	**15.56**	2.8392	**32.62**	**29.21**	13.5131
GFHF	kNN	26.97	15.00	0.4077	28.12	23.02	0.2020	63.17	53.87	1.3322
	$\mathcal{N}$-Fitted	31.42	17.28	6.8102	28.81	26.27	5.0929	63.41	56.37	16.9751
	SIS	31.72	17.16	35.5183	25.24	25.08	2.5673	64.76	57.93	21.3485
	$\mathcal{M}$-Fitted	**26.77**	**13.88**	6.0885	**20.97**	**15.64**	2.1755	**59.32**	**49.48**	11.1181
GFHF + CMN	kNN	17.93	11.87	0.3997	46.12	33.84	0.2113	30.12	27.28	1.3483
	$\mathcal{N}$-Fitted	19.76	13.37	13.7010	52.53	41.37	4.7314	30.18	27.11	17.8843
	SIS	19.05	13.00	35.5608	40.83	27.76	2.6938	29.96	27.52	28.2256
	$\mathcal{M}$-Fitted	**17.25**	**11.16**	6.2676	**38.23**	**25.38**	5.2851	**29.67**	**26.69**	12.2025

SSL takes.

Tables 1, 2 and Figure 4 clearly show that our proposed $\mathcal{M}$-fitted graphs lead to superior classification accuracy over the other graph construction approaches. In terms of the time cost, $\mathcal{M}$-fitted graphs consistently run faster than the competing $\mathcal{N}$-fitted and SIS graphs. Notice that $\mathcal{M}$-fitted graphs always include the kNN search step, so they are slower than kNN graphs.

6 Conclusion and Future Work

In this paper, we propose a sparse manifold fitting approach to embed the locally sparse manifold structure into neighborhood graph construction. In order to address the potential issues of ℓ_1 relaxed sparse coding, our approach directly employs ℓ_0 norm to induce sparsity, and designs a forward-backward fitting algorithm to yield an explicitly sparse solution in an efficient manner. Through extensive experiments, we demonstrate the performance of the novel $\mathcal{M}$-fitted graph generated by performing sparse manifold fitting for every input data point. $\mathcal{M}$-fitted graphs not only run faster than state-of-art graphs, but also result in superior classification accuracy when collaborating with well-known graph-based semi-supervised learning methods.

In the future work, we would like to study scaling up the $\mathcal{M}$-fitted graph proposed in this paper and enable it to work for massive datasets that are increasingly encountered in the current big data era. The prior work Anchor Graph [Liu *et al.*, 2010] offers a good example, but it has not incorporated too much manifold information which would become more evident in large-scale data collections. If we developed a scalable graph construction approach such that the underly-

E5630 quad-core CPU and 48GB RAM. All methods mentioned in this paper are implemented in MATLAB.

ing manifold structure can be captured to a deeper extent, we could improve the performance of some popular graph-based web-scale applications such as ranking [Xu *et al.*, 2011] and hashing [Liu *et al.*, 2011b].

Acknowledgement:

This work is supported in part by Xiamen University 985 Project. Wei Liu is supported by the Josef Raviv Memorial Postdoctoral Fellowship.

References

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

[Belkin *et al.*, 2006] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[Bohanec and Rajkovič, 1988] M. Bohanec and V. Rajkovič. Knowledge acquisition and explanation for multi-attribute decision. In *Proc. International Workshop of Expert Systems and Their Applications*, 1988.

[Breiman *et al.*, 1984] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

[Candés and Tao, 2005] E. Candés and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[Cheng *et al.*, 2009] H. Cheng, Z. Liu, and J. Yang. Sparsity induced similarity measure for label propagation. In *Proc. ICCV*, 2009.

[Elhamifar and Vidal, 2009] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Proc. CVPR*, 2009.

[Elhamifar and Vidal, 2010] E. Elhamifar and R. Vidal. Clustering disjoint subspaces via sparse representation. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[Elhamifar and Vidal, 2011] E. Elhamifar and R. Vidal. Sparse manifold clustering and embedding. In *NIPS 24*, 2011.

[Frank and Asuncion, 2010] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[Gao *et al.*, 2010] S. Gao, I. W. Tsang, and L. Chia. Kernel sparse representation for image classification and face recognition. In *Proc. ECCV*, 2010.

[Hastie *et al.*, 2009] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Second Edition, New York: Springer-Verlag, 2009.

[Jebara *et al.*, 2009] T. Jebara, J. Wang, and S.-F. Chang. Graph construction and b-matching for semi-supervised learning. In *Proc. ICML*, 2009.

[Jiang *et al.*, 2012] Y.-G. Jiang, Q. Dai, J. Wang, C.-W. Ngo, X. Xue, and S.-F. Chang. Fast semantic diffusion for large-scale context-based image and video annotation. *IEEE Transactions on Image Processing*, 21(6):3080–3091, 2012.

[Jing and Baluja, 2008] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1877–1890, 2008.

[Liu and Chang, 2009] W. Liu and S.-F. Chang. Robust multi-class transductive learning with graphs. In *Proc. CVPR*, 2009.

[Liu *et al.*, 2010] W. Liu, J. He, and S.-F. Chang. Large graph construction for scalable semi-supervised learning. In *Proc. ICML*, 2010.

[Liu *et al.*, 2011a] W. Liu, Y.-G. Jiang, J. Luo, and S.-F. Chang. Noise resistant graph ranking for improved web image search. In *Proc. CVPR*, 2011.

[Liu *et al.*, 2011b] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *Proc. ICML*, 2011.

[Nene *et al.*, 1996] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). Technical report, Columbia University, 1996.

[Ng *et al.*, 2001] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*, 2001.

[van Breukelen and Duin, 1998] M. van Breukelen and R. P. W. Duin. Neural network initialization by combined classifiers. In *Proc. International Conference on Pattern Recognition*, 1998.

[Wang and Zhang, 2008] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2008.

[Weston *et al.*, 2003] J. Weston, C. S. Leslie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. In *NIPS 16*, 2003.

[Xu *et al.*, 2011] B. Xu, J. Bu, C. Chen, D. Cai, X. He, W. Liu, and J. Luo. Efficient manifold ranking for image retrieval. In *Proc. SIGIR*, 2011.

[Zhang and Zha, 2004] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, 2004.

[Zhang, 2011] T. Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, 57(7):4689–4708, 2011.

[Zhou *et al.*, 2003a] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS 16*, 2003.

[Zhou *et al.*, 2003b] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS 16*, 2003.

[Zhu and Goldberg, 2009] X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan & Claypool, 2009.

[Zhu *et al.*, 2003] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. ICML*, 2003.