Neurocomputing 315 (2018) 107–114

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Structured sparse graphs using manifold constraints for visual data analysis

F. Dornaika[a,b,*], L. Weng[a], Z. Jin[c]

[a] UPV/EHU, University of the Basque Country, San Sebastian, Spain
[b] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
[c] Nanjing University of Science and Technology, Nanjing, China

ARTICLE INFO

Article history:
Received 9 February 2018
Revised 9 June 2018
Accepted 26 June 2018
Available online 5 July 2018

Communicated by Su-Jing Wang

Keywords:
Graph construction
Sparse representation
Manifold constraints
Graph-based semi-supervised learning
Label propagation
Image classification

ABSTRACT

Data-driven graphs constitute the cornerstone of many machine learning approaches. Recently, it was shown that sparse graphs (sparse representation based graphs) provide a powerful approach to graph-based semi-supervised classification. In this paper, we introduce a new structured sparse graph that is derived by integrating manifold-type constraints on the sparse coefficients without any a priori graph or similarity matrix. Furthermore, we introduce a direct and efficient solution to the proposed optimization problem. Unlike recent sparse graph construction methods that are based on the use of hand-crafted constraints or a predefined reference similarity matrix, our constraints are directly defined on the graph weights themselves, and can provide additional information to both local and global structures of the sparse graph. Experiments conducted on several image databases show that the proposed graph can give better results than many state-of-the-art sparse graphs when applied to the problem of graph-based label propagation.

1. Introduction

Graphs are entities that can encode pairwise similarity among data samples [1–6]. Graph-based learning tasks operate on a data driven graph [7–16]. In the graphs, the vertices correspond to data samples and the weighted edges between vertices quantify the similarity between two vertices. The most known method to build a graph, or equivalently to compute its affinity matrix, is to build k-nearest neighbor graphs [17] or ε-neighborhoods graphs. The obtained edges are then estimated using a pairwise similarity function that encodes the similarity between a vertex and its nearest neighboring vertexes (data samples). These approaches suffer from two major drawbacks. First, setting the parameters of these approaches is very challenging. The parameters of such approaches can impact the final performance of the task [18]. Second, even with their best parameter configuration, these approaches are not guaranteed to provide the best results when compared to other competing graph construction methods. In [8], the authors exploit the criterion used by Locally Linear Embedding (LLE) in order to estimate the edge weights of the graphs using a linear coding scheme. In [9], the authors rely on the LLE criterion for semi-supervised dimensionality reduction and estimate a neighborhood preserving graph. On the other hand, sparse coding becomes a widely adopted tool which supposes that any signal can be composed by some basic signals. In [11], the authors build a sparse graph for which the edges weights are set to the coefficients of the sparse coding. They apply the obtained graph to the problem of human face recognition via the computation of the linear projection associated with the Locality Preserving Projection method. In [19], the authors also use the sparse codes in order to build the graph. The graph is used for semi-supervised classification [20] and multi-label classification [21]. A lot of graph based semi-supervised learning (SSL) techniques have been introduced [22–24]. These techniques can be classified into two categories: (i) those that exploit the graph structure and weights to propagate labels from labeled data samples to unlabeled ones and (ii) those that provides a data projection that incorporates the smoothness constraints encoded by the graph. Nowadays, graph based SSL have led to many advances in many real world applications such as emotion recognition in videos, face recognition, audio recognition, text classification, webpage classification, protein structure prediction, and image classification.

Despite its increasing popularity, little work performed comprehensive and unbiased empirical studies that show the impact that graph construction methods have in both, stability and classification performance of the graph-based learning tasks.

* Corresponding author at: UPV/EHU, University of the Basque Country, San Sebastian, Spain.
E-mail address: fadi.dornaika@ehu.es (F. Dornaika).

https://doi.org/10.1016/j.neucom.2018.06.067
0925-2312/

In [25], the authors propose a graph construction method using b-matching. The objective is to enforce that all the vertexes of the graph will have the same degree. The degree of a vertex refers to the number of the edges connected to the vertex. In the past decade, researchers proposed many different coding schemes and code book generation algorithms. In [26], the authors proposed to build hard graphs by exploiting a criterion similar to the one used by Roweis and Saul [27]. In this work, the edge weighting and neighbor selection are conducted in a single phase; the affinity matrix of the graph is symmetric and only positive or zero weights are allowed. The weights are estimated by constraining the degree, or the weighted degree of each vertex, to be equal to or greater than one. The authors develop a quadratic program that estimates the non-negative weights. The non-tractability of the solution is avoided by adopting an incremental approach in which only one subset of edges is estimated at a time.

In [28], the authors propose semantic ℓ_1 and KNN graphs in order to infer the label of superpixels (image regions) in a collection of labeled images. The objective is to provide a parsing for the images. The sought graph similarity matrix is related to all superpixels. For semantic ℓ_1 graph, instead of performing the ℓ_1 coding with all superpixels, it is performed on individual dictionary that collects the superpixels in all images having a given label. The chosen coding dictionary is the one that provides the smallest ℓ_1 norm. For semantic KNN graphs, the coding dictionary is chosen such that it minimizes the pairwise similarity between the neighbors of a given class and the neighbors of its complementary class. In [29], the author proposes two types of semantic hypergraphs, which explore both intra-image and inter-image high-order semantic relevance. Besides, they provide a scheme for fusion three graphs over the superpixels: the two semantics hypergraph and the semantic KNN graph.

In [30], we have proposed a graph construction method that is based on data self-representativeness. The method uses Local Hybrid Coding on adaptively obtained local and non-local bases. Both locality and sparsity are simultaneously taken into account when generating the graph affinity matrix. The resulting graph can be very informative. However, the method has a high computational cost.

In [14], we proposed a dynamic graph construction method for inductive semi-supervised learning. The proposed method, after receiving new samples, updates the affinity graph dynamically without the need to construct the graph from scratch. In the first phase, the new samples are appended to the graph. Then, data samples, in the already available graph, who are similar to the new added samples undergo possible changes of their edges and edge weights.

1.1. Motivation and contribution

The fact that images lie on a manifold has been successfully exploited in many inference and learning tasks. Therefore, many methods relied on the assumption that data are on or close to a manifold [31–36]. They attempted to get a linear or non-linear projection from the original space in which data live to a low dimensional subspace. Therefore, when one has to categorize a well-defined object category (e.g., digits, faces, animals, etc.), any classification paradigm taking advantage of the recovered low dimensional subspace can get improved performance. For instance, locality preserving is basically used in Laplacian eigenmaps [31] and its linearized version (locality preserving projection) [32].

This paper introduces a new insight to the graph construction problem. In particular, the paper exploits a relevant intuition that was neglected in previous graph construction methods. For data projection, manifold constraints were already used. However, for the problem of graph construction, the main crux is to efficiently embed these manifold constraints on the graph structure since these constraints are depending on the graph itself. This paper proposes a structured sparse graph construction method in which the graph is obtained by enforcing these manifold constraints. The proposed method is called structured sparse graph (SSG). Unlike constrained sparse graphs that are based on the use of hand-crafted constraints or a predefined reference similarity matrix, our proposed SSG method is generic and implicitly exploits smoothness constraints on the graph weights. Therefore, it provides additional information to the global and local structures of the sparse graph.

To the best of our knowledge, our previous work [37] is the first one that investigates the use of such manifold constraints on the rows or columns of the unknown affinity matrix associated with the graph. In this paper, we propose two main extensions to the proposed framework in [37]. Firstly, our current work introduces a criterion that provide sparse graphs; whereas the criterion of [37] provides non-sparse graphs. Secondly, the current work proposes a direct minimization of the criterion in which the graph affinity matrix is directly estimated whereas in [37] the graph is recursively estimated. The resulting graph integrates three powerful criteria needed for getting an informative graph: (i) data self-representativeness, (ii) graph sparsity, and (iii) manifold constraints on the graph coefficients.

In [38] and [39], a novel scheme for dictionary learning was introduced. The authors estimate the dictionary from data by imposing the smoothness of sparse codes of the data. This is easily encoded since the data graph is known in advance. Unlike [38] and [39], where the graph is known a priori, our proposed approach aims to estimate the graph from data alone using manifold constraints on the edge weights. The main difference between our introduced method and the constrained graphs described in [40] and [41] is the fact that ours does not require the availability of a predefined affinity matrix. Thus, our work avoids the dependency on an a priori affinity matrix.

The paper is organized as follows. Section 2 briefly reviews some related works in sparse graph construction. Section 3 presents the proposed graph construction method. Section 4 gives some experimental results obtained with five real image datasets, showing the efficacy and efficiency of the proposed method. Section 5 presents some conclusions. Matrices are denoted by capital bold letters and vectors are denoted by small bold letters.

2. Review of sparse graphs

2.1. Sparse graphs

Qiao et al. [11] and Yan and Wang [19] proposed sparsity representation based graph construction methods in which every sample is represented as a sparse linear combination of the rest of input samples and the coefficients are considered as weights. These are estimated using ℓ_1 minimization given by:

$$\min \|\mathbf{s}_i\|_1, s.t. \quad \mathbf{x}_i = \mathbf{X}\,\mathbf{s}_i, \tag{1}$$

where $\mathbf{s}_i = [s_{i1}, \ldots, s_{i,i-1}, 0, s_{i,i+1}, \ldots, s_{in}]^T$ is a vector $\in \mathbb{R}^n$ whose ith component is equal to zero (meaning that the sample $\mathbf{x}_i$ is extracted from $\mathbf{X} \in \mathbb{R}^{D\times n}$).

After the weight vector $\mathbf{s}_i$ for each $\mathbf{x}_i$, $i = 1, 2, \ldots, n$ is obtained, the graph matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ can be defined by:

$$\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n]^T, \tag{2}$$

where $\mathbf{s}_i$ corresponds to the solution of Eq. (1).

Robust sparse graphs can be obtained if the sparse codes are estimated using a robust ℓ_1 minimization problem:

$$\min \|\mathbf{s}_i\|_1 + \|\mathbf{e}\|_1, s.t. \quad \mathbf{x}_i = \mathbf{X}\,\mathbf{s}_i + \mathbf{e}. \tag{3}$$