Neural Networks 95 (2017) 91–101

Contents lists available at ScienceDirect

Neural Networks

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

Graph construction using adaptive Local Hybrid Coding scheme

Fadi Dornaika [a,b,*], Mahdi Tavassoli Kejani [c], Alireza Bosaghzadeh [d]

[a] University of the Basque Country, UPV/EHU, Manuel Lardizabal 1, 20018 San Sebastian, Spain
[b] IKERBASQUE, Basque Foundation for Science, Maria Diza de Haro, 3, 48013 Bilbao, Spain
[c] University of Isfahan, Iran
[d] Shahid Rajaee Teacher Training University, Tehran, Iran

ARTICLE INFO

Article history:
Received 20 September 2016
Received in revised form 4 August 2017
Accepted 8 August 2017
Available online 24 August 2017

Keywords:
Graph construction
Sparse coding
Local Hybrid Code
Label propagation
Classification

ABSTRACT

It is well known that dense coding with local bases (via Least Square coding schemes) can lead to large quantization errors or poor performances of machine learning tasks. On the other hand, sparse coding focuses on accurate representation without taking into account data locality due to its tendency to ignore the intrinsic structure hidden among the data. Local Hybrid Coding (LHC) (Xiang et al., 2014) was recently proposed as an alternative to the sparse coding scheme that is used in Sparse Representation Classifier (SRC). The LHC blends sparsity and bases-locality criteria in a unified optimization problem. It can retain the strengths of both sparsity and locality. Thus, the hybrid codes would have some advantages over both dense and sparse codes. This paper introduces a data-driven graph construction method that exploits and extends the LHC scheme. In particular, we propose a new coding scheme coined Adaptive Local Hybrid Coding (ALHC). The main contributions are as follows. First, the proposed coding scheme adaptively selects the local and non-local bases of LHC using data similarities provided by Locality-constrained Linear code. Second, the proposed ALHC exploits local similarities in its solution. Third, we use the proposed coding scheme for graph construction. For the task of graph-based label propagation, we demonstrate high classification performance of the proposed graph method on four benchmark face datasets: Extended Yale, PF01, PIE, and FERET.

1. Introduction

Semi-supervised learning is one of the most important fields in machine learning. Very often, it is used for cases for which there are far more unlabeled data than labeled. It can be attractive whenever acquiring data samples is cheap, but getting the labels costs a lot of time, effort, or money. This is the case in many application areas of machine learning, for example: (i) static image classification, (ii) person emotion recognition in videos (Araujo & Kamel, 2014), (iii) speech recognition (Liu & Kirchhoff, 2014), (iv) webpage classification (Bai, Zhang, Li, & Li, 2012), and (v) protein sequence classification (Shi & Zhang, 2010). During the last decade, the most active area of research in semi-supervised learning has been in graph-based methods (e.g., Dornaika & Bosaghzadeh, 2015; Gong, Tao, Fu, & Yang, 2015; Gong, Tao, Liu, Liu, & Yang, 2016; Gong, Tao, Maybank, Liu, Kang, & Yang, 2016; Yuan, Mou, & Lu, 2015). Indeed, graph-based algorithms are widely used nowadays in a variety of machine learning tasks such as: (i) semi-supervised learning for label propagation and regression (Gong et al., 2016), (ii) feature selection, (iii) graph-based embedding (Gong, Tao, Yang, & Fu, 2014), (iv) spectral clustering (Wang, Qian, & Davidson, 2014), and (v) uncertainty sampling (Yang, Ma, Nie, Chang, & Hauptmann, 2015).

Despite the increasing popularity of the graph-based methods, the literature lacks comprehensive and unbiased empirical studies that show the influence that graph construction methods have in both, classification performance and stability of the graph-based learning algorithms. In some real world domains (e.g., the web, social networks, citation networks, etc.), the data is relational in nature and there is already an implicit underlying graph. However, for a majority of learning tasks, the data instances are assumed to be independent and identically distributed (i.i.d.). In such cases, there is no explicit graph structure to start with. For instance, unlabeled images of objects do not have an initial relational model. It is challenging to choose the pairwise relation for automatic graph construction methods. This difficulty is obvious when dealing with instances used by Fine Grained Recognition (e.g., face recognition, plant categorization). The common practice is to create a graph in the first step from independent data instances, and in the second step apply one of the graph-based learning tasks on the constructed graph. Recently, researchers developed many coding

* Corresponding author at: University of the Basque Country, UPV/EHU, Manuel Lardizabal 1, 20018 San Sebastian, Spain.
E-mail addresses: fadi.dornaika@ehu.es (F. Dornaika), mtk.tavassoli@gmail.com (M.T. Kejani), A.bosaghzadeh@srttu.edu (A. Bosaghzadeh).

http://dx.doi.org/10.1016/j.neunet.2017.08.002
0893-6080/

schemes for data representation. These coding schemes can be used for collaborative representation based classification (e.g. Xu, Zhang, Yang, & Yang, 2011; Zhang, Yang, Feng, Ma, & Zhang, 2012), data graph building (e.g., Cheng, Yang, Yan, Fu, & Huang, 2010; Dornaika, Bosaghzadeh, & Raducanu, 2013; He, Zheng, Hu, & Kong, 2011; Nie, Wang, Jordan, & Huang, 2016), and dictionary learning (e.g., Wang, Yang, Yu, Lv, Huang, & Gong, 2010).

Recently, the importance of graph construction for resulting success in post learning algorithms has begun to be recognized (Cheng et al., 2010; Jebara, Wang, & Chang, 2009; Maier, von Luxburg, & Hein, 2008). Over the past decade, several graph construction methods have been proposed. In Zhang, Qiao, and Chen (2010), the authors proposed a unified method which calculates the graph and the embedding space in an iterative process. The main goal is to minimize the augmented Locality Preserving Projections criterion (He & Niyogi, 2003) over both the linear transform and the weight (affinity) matrix of the graph. The Linear Neighborhood Propagation (LNP) algorithm (Wang & Zhang, 2008) is a two step process whose first step involves a graph construction and second step involves label propagation through the graph edges. The method proceeds as follows. Firstly, the adjacency graph is estimated using the traditional KNN graph. Secondly, the weights of the graph are estimated using a similar criterion to Roweis and Saul (2000) (the weights of a node sum to one) with the additional constraint of non-negative weights.

In this paper, we propose a novel graph construction method that is based on the recently proposed Local Hybrid Coding (Xiang, Wang, & Long, 2014). This coding scheme retains the advantages of both the bases-locality and sparsity criteria in a unified optimization problem. Thus, LHC tends to retain the advantages of two types of coding. The first type (i.e., dense coding with ℓ_2 regularization) captures the intrinsic geometric structure of data manifold, introducing the discriminative power to the codes for better classification accuracy (Waqas, Yi, & Zhang, 2013). The second type (i.e., ℓ_1-sparsity) guarantees that the generated codes can represent input features accurately using only a small number of activated coefficients (Sprechmann, Bronstein, & Sapiro, 2015; Wright, Ma, Mairal, Sapiro, Huang, & Yan, 2010; Wright, Yang, Ganesh, Sastry, & Ma, 2009).

The main differences between our approach and the LHC scheme of Xiang et al. (2014) are as follows. First, in our work, we construct a data-driven graph using data self-representativeness whereas in Xiang et al. (2014), the authors propose a variant of the Sparse Representation Classifier that uses the hybrid coding instead of the sparse coding. Thus, in our work, the dictionary used is given by the data themselves, whereas in Xiang et al. (2014), the dictionary is pre-trained. Second, in our work, the local and sparse bases are determined according to similarity coefficients that are given by a Locality-constrained Linear Coding (LLC) scheme whereas in Xiang et al. (2014), the selection of local and non-local bases is solely based on Euclidean distances in original space. Third, our proposed adaptive LHC uses biased weights for the coefficients of the local bases.

The remainder of the paper will be organized as follows: In Section 2, some related graph construction works will be introduced. Besides, this section reviews the LHC scheme (Xiang et al., 2014). In the third section, the proposed method is presented. In the fourth section, we present some experiments that are conducted on four real image datasets to prove the efficacy of the proposed method. In the last section, some conclusions will be drawn. In the sequel, capital bold letters denote matrices and small bold letters denote vectors. For convenience, Table 1 summarizes the main notations used in this paper.

2. Related work

This section describes some existing methods for graph construction. Then, it will present a review of the recent Local Hybrid Coding scheme.

Table 1
Description of the main notations used in this paper.

Notation	Description
n	Number of images or samples
$\mathbf{W}$	Graph similarity matrix $\in \mathbb{R}^{n\times n}$
$\mathbf{x}, \mathbf{b}$	Data sample or feature vector $\in \mathbb{R}^d$
d	Dimension of the feature vectors $\mathbf{x}$ and $\mathbf{b}$
$\mathbf{X}, \mathbf{B}$	Matrices of samples
$\mathbf{X}^{(l)}, \mathbf{B}^{(l)}$	Matrices of local bases
$\mathbf{X}^{(s)}, \mathbf{B}^{(s)}$	Matrices of non-local bases
$\mathbf{c}$	Vector of coding coefficients
$\mathbf{c}^{(l)}$	Local code
$\mathbf{c}^{(s)}$	Non-local (sparse) code
k	Neighborhood size for KNN adjacency graphs
ϵ	Neighborhood radius
k_l	Size of local bases
k_s	Size of non-local bases
$\sigma, \gamma, \lambda, \alpha, \beta$	Regularization parameters
$\mathbf{P}, \mathbf{D}$	Diagonal matrices
$\mathbf{e}$	Residual error vector
q	Number of labeled images per class

2.1. Graph construction methods

2.1.1. Traditional graphs

k-nearest neighbor graphs and ε-neighborhoods graphs are two traditional graph construction methods. Let the original dataset be denoted by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$.

In KNN graphs, samples $\mathbf{x}_i$, $\mathbf{x}_j$ are connected by an edge if $\mathbf{x}_i$ is in $\mathbf{x}_j$'s k-nearest neighborhood or vice versa. k is a parameter that controls the density of the graph.

In ε-neighborhoods graphs, samples $\mathbf{x}_i$, $\mathbf{x}_j$ are connected by an edge if the distance $d(\mathbf{x}_i, \mathbf{x}_j) \leq \varepsilon$. The parameter ε controls neighborhood radius.

After the edges are decided, those should be weighted using pairwise similarities. Thus, every entry in the graph affinity matrix is given by Eq. (1). For instance, $sim(\mathbf{x}_i, \mathbf{x}_j)$ can be set to 1 or $sim(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\mu}}$, where μ is a parameter.

$$W_{ij} = \begin{cases} sim(\mathbf{x}_i, \mathbf{x}_j), & \mathbf{x}_i \in \delta_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \delta_k(\mathbf{x}_i), \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In Jebara et al., (2009), the authors propose a graph construction method via b-Matching. The goal is to produce a binary adjacency matrix with the constraint that the resulting graph is undirected (symmetric weight matrix) and the constraint that each node will have the same degree[1] given by the parameter b. The solution is obtained by loopy belief propagation. It is shown that the label propagation algorithm that uses the resulting adjacency graph has slightly better performance than that based on the KNN graph. However, the b-matching graph construction needs tuning the parameter b. Furthermore, since the output of b-matching is a binary weight matrix, an additional stage is needed for edge re-weighting.

2.1.2. Data self-representation graphs

Locally Linear Embedding (LLE) focuses on preserving the local structure of data (Roweis & Saul, 2000). LLE formulates the manifold learning problem as a neighborhood-preserving embedding, which learns the global structure by exploiting the local linear reconstructions. It estimates the reconstruction coefficients by minimizing the reconstruction error of the set of all local neighborhoods in the dataset. It turned out that the linear coding used by LLE can be used for computing the graph weight matrix.

[1] The degree of a node is equal to the sum of weights of all edges linked to that node.