Neurocomputing 171 (2016) 901–909

Contents lists available at ScienceDirect

Neurocomputing

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

Local voting based multi-view embedding

Xinjian Gao [a,*], Tingting Mu [b], Meng Wang [a]

[a] School of Computer and Information, Hefei University of Technology, China
[b] Department of Electrical Engineering and Electronics, The University of Liverpool, UK

ARTICLE INFO

Article history:
Received 26 February 2015
Received in revised form
1 May 2015
Accepted 12 July 2015
Communicated by Yongdong Zhang
Available online 30 July 2015

Keywords:
Multi-view learning
Embeddings
k-NN search
Semi-supervised learning
Pairwise constraints

ABSTRACT

This paper studies subspace based multi-view learning, investigating how to mine useful information within various kinds of data or features (views) and achieve the optimal cooperation between views. Unlike most existing methods focused on learning an optimal weighting scheme to linearly combine different types of view information, we propose to first improve the original information provided by each view by designing a voting based scheme to model individual neighbor structures of the data. This leads to a set of refined local proximity matrices corresponding to different confidence levels. Then, different schemes can be applied to further combine this refined set of composite local neighborhood representations. Also, we provide the semi-supervised version of the proposed algorithms to incorporate partially labeled objects. The experimental results demonstrate effectiveness and robustness of the proposed algorithms.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

As the development of sensor and computer techniques, data that exhibits heterogeneous properties of the studied objects can be collected from various domains, feature collectors and extractors. These are often referred as multi-view representations of the objects and correspond to the multi-view learning task in machine learning, which has facilitated complex data analysis in areas such as video surveillance, multimedia, and image classification [1]. Each representation (view) has different physical meaning and particular statistical property. It has the potential to improve the performance of a given task by enabling effective collaboration between multiple views, so that a view can be enhanced by the complementary of the other views.

To deal with multi-view learning task, conventional machine learning approaches, such as support vector machines, discriminant analysis, and kernel machines, usually treat simply multi-view information as one view, which yields several disadvantages. For example, the complementary of views may not be thoroughly explored. Also, the physical meaning of each view may be eliminated and the simple concatenate can cause the curse of the dimensionality.

Recently, subspace-based multi-view learning techniques [2,3] have attracted increasing attentions, shown to be an effective strategy to combine multi-view information. It aims at learning a subspace under the assumption that there exist a subspace which can generate all the views. The dimensionality will be reduced and the complementary of multiple views will be explored when projecting different views from their original individual spaces to a common subspace. Such strategy has been applied to solve various data processing and machine learning problems in multiples areas, such as multimedia, image classification and machine learning [4–7].

A common approach for computing the subspace-based multi-view embeddings includes two stages of processing, starting from computing subspace embeddings for a single view, and then extending it by seeking appropriate combinations of the single-view embeddings.

There exists rich literature discussing how to compute the embedded feature representation. For example, a summary of different embedding methods can be found [8]. Specific embedding learning algorithms include principal component analysis (PCA) [9], linear discriminant analysis (LDA) [10], locally linear embedding (LLE) [11], ISOMAP [12], Laplacian eigenmap (LE) [13–15] and Hessian eigenmaps [16].

Based on these works, various algorithms for multi-view embedding have been developed to support the combination of multi-view information [17–21]. For example, the multiview spectral embedding (MSE) algorithm [19] applies LE to project different views to the same subspace, and utilizes the alternating optimization approach to compute the optimal weights for combining different views as well as the optimal embeddings for each view. When there is label information available to support learning embeddings to facilitate classification tasks, label information can be formulated as pairwise

* Corresponding author.
E-mail address: gao_xinjian@outlook.com (X. Gao).

http://dx.doi.org/10.1016/j.neucom.2015.07.043
0925-2312/© 2015 Elsevier B.V. All rights reserved.

constrains to enhance multi-view feature fusion [20]. When the label information is only available for a small amount of data objects, semi-supervised learning strategy can be developed, computing multi-view embedding from both labeled data and unlabeled data, for example, the Grassmannian regularized structured multiview embedding (GrassReg) algorithm [21] inspired by Fisher discriminative analysis (FDA) [10]. The algorithm calculates distances between different views based on the Grassmannian manifold [22,23], and takes such distances into account by assuming the views that are farther away from the other views is less confidential and vice versa.

Although the aforementioned algorithms are capable of combining multi-view information, these algorithms only consider the confidence level of the information provided by each view on its own, and combine by simple adding up. Such processing ignores the fact that even a view with the least confidence level may contain possible local structures that could benefit the performance of a specific task, while even a view with the highest confidence level may contain certain local structures that may harm the performance.

In this work, we take into account the aforementioned issues in multi-view embedding design. Unlike the conventional methods [19–21] that treat the original feature representations of different view as different input information resources for further combination, we propose to operate on a set of refined information resources, which correspond to a set of weight matrices representing a set of composite local neighborhood structures voted by different numbers of views. We suggest three schemes to compute embeddings based on such refined voting-based information. This results in a less noisy embedded subspace which is able to take into account the complementary information among views. Such design breaks the limitation of considering the same relationship between views for the whole set of objects as most existing algorithms do, while enables the opportunity of considering different relationships between views on individual local neighborhoods, thus offers more flexible and more accurate data structure modeling.

The rest of the paper is organized as follows: Section 2 presents the proposed algorithm. Section 3 explains the experimental setup and analyze the results. Section 4 concludes the work in the end.

2. Local voting based multi-view embedding

In this work, we study effective way of combining multi-view information to facilitate a classification task. Given a set of n objects each belonging to one of the c categories, the ith object is represented by a set of m feature vectors $\left\{\mathbf{x}_i^{(s)}\right\}_{s=1}^{m}$, where $\mathbf{x}_i^{(s)} = \left[x_{i1}^{(s)}, x_{i1}^{(s)}, x_{i1}^{(s)}, \ldots, x_{id_s}^{(s)}\right]$ and d_s denotes the dimensionality of each feature vector. Each d_s-dimensional feature vector corresponds to the feature representation of an object under the sth view. The $n \times d_s$ matrix $\mathbf{X}^{(s)} = [x_{ij}^{(s)}]$ is used to denote the feature matrix of all the objects under the sth view. The goal is to seek a low-dimensional embedded space, where the ith object is represented as a k-dimensional embedding vector $\mathbf{z}_i = [z_{i1}, z_{i2}, z_{i3}, \ldots, z_{ik}]$. This is equivalent to learning an $n \times k$ embedding matrix $\mathbf{Z} = [z_{ij}]$ for the n objects by combining the m different feature matrices $\left\{\mathbf{X}^{(s)}\right\}_{s=1}^{m}$ corresponding to m different views.

The single-view case ($m=1$) is equivalent to the standard embedding problem, and can be solved by computing the eigen-decomposition of a Laplacian matrix [8], for example the widely used classical approach LE [13]. One way for computing the Laplacian matrix $\mathbf{L}$, which is a square matrix with n rows corresponding to the n studied objects, is given as follows:

$$\mathbf{L} = \mathbf{I} - \frac{1}{\sqrt{\mathbf{D}}}\mathbf{W}\frac{1}{\sqrt{\mathbf{D}}}. \tag{1}$$

Here, $\mathbf{I}$ is an identity matrix of size n, the weight matrix $\mathbf{W} = [w_{ij}]$ is an $n \times n$ matrix computed from the input feature information, e.g., $\mathbf{X}^{(1)}$, characterizing the desired closeness between objects, and the $n \times n$ matrix $\mathbf{D}$ is a diagonal matrix with each diagonal element formed by the row sum of $\mathbf{W}$. For the multi-view case ($m > 1$), the strategy is to seek appropriate design of $\mathbf{W}$ based on the multi-view information $\left\{\mathbf{X}^{(s)}\right\}_{s=1}^{m}$.

2.1. Refine information by local voting

In this section, we first propose a local voting scheme to explore the closeness information between objects by considering the agreement and complementary between views.

For each view, we first compute an $n \times n$ proximity matrix $\mathbf{P}_s = \left[p_{ij}^{(s)}\right]$ from the corresponding feature matrix $\mathbf{X}_s$ based on a similarity measure, such as Euclidean distance and those as summarized in [8], where each element $p_{ij}^{(s)}$ represents the closeness between the ith and jth object. The local geometry of the objects is then explored by conducting the k-nearest neighbor (k-NN) search over each proximity matrix $\mathbf{P}_s$, leading to the following indicator:

$$\delta_{ij}^{(s)} = \begin{cases} 1 & \text{if } x_i^{(s)} \text{ and } x_j^{(s)} \text{ are undirected k} - \text{NNs} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

This results in the $n \times n$ indicating matrix $\mathbf{\Delta}^{(s)} = [\delta_{ij}^{(s)}]$. To highlight such local neighbor structure, we further weight $\mathbf{P}_s$ and obtain the following local proximity matrix:

$$\mathbf{P}_w^{(s)} = \mathbf{\Delta}^{(s)} \circ \mathbf{P}^{(s)}, \tag{3}$$

where $\circ$ denote Hadamard product.

Feature information generated under multiple views characterize the objects in multiple ways, leading to multiple local geometry structures as stored in $\left\{\mathbf{P}_w^{(s)}\right\}_{s=1}^{m}$. For many problems, there is no priorly known information on the superiority of views. Also there is a chance that a weak view may contain some information that could benefit the performance of a specific task, while strong view may contain some information that may harm the performance. It is difficult to pick up such outlier information within each individual view without introducing communication between views. Given different local neighborhood structures identified by different views, one possible way to evaluate their validity is to conduct comparison between these structures, for example, to examine whether a neighbor pair is identified and agreed by most of the views. Those neighbor pairs agreed by more views are of higher probability to be correct. Thus, agreements between views can be considered as a potentially reliable measurement to assess the quality of the information provided by different views. Guided by this, we further refine the local proximity information stored in $\left\{\mathbf{P}_w^{(s)}\right\}_{s=1}^{m}$ by separately examining varying numbers of views that agree with each other on the neighborhood structure between the objects, in order to achieve better collaboration between views.

More specifically, we define that if there exist exactly a indicators ($0 \le a \le m$) equal to one, these indicators are used to vote local neighborhood with confidence level a/m. The set $\mathbf{I}^a$ is used to record the indices of these a indicators, indicating the features that recognize this ijth object pair as a neighbor pair. Otherwise, for example, when there exist more or less than a features agreeing on the neighborhood, or when all the features have identified the ijth

Download English Version:

https://daneshyari.com/en/article/407464

Download Persian Version:

https://daneshyari.com/article/407464

Daneshyari.com