Neurocomputing 267 (2017) 210–219

Contents lists available at ScienceDirect

Neurocomputing

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

Kernel-driven similarity learning

Zhao Kang[a,b,*], Chong Peng[b], Qiang Cheng[b]

[a] School of Computer Science & Engineering University of Electronic Science and Technology of China Chengdu, Sichuan, 611731, China
[b] Department of Computer Science, Southern Illinois University Carbondale, IL, 62901, USA

ARTICLE INFO

Article history:
Received 11 August 2016
Revised 11 March 2017
Accepted 2 June 2017
Available online 9 June 2017

Communicated by Dr. Haiqin Yang

Keywords:
Similarity measure
Nonlinear relation
Sparse representation
Kernel method
Multiple kernel learning
Clustering
Recommender systems

ABSTRACT

Similarity measure is fundamental to many machine learning and data mining algorithms. Predefined similarity metrics are often data-dependent and sensitive to noise. Recently, data-driven approach which learns similarity information from data has drawn significant attention. The idea is to represent a data point by a linear combination of all (other) data points. However, it is often the case that more complex relationships beyond linear dependencies exist in the data. Based on the well known fact that kernel trick can capture the nonlinear structure information, we extend this idea to kernel spaces. Nevertheless, such an extension brings up another issue: its algorithm performance is largely determined by the choice of kernel, which is often unknown in advance. Therefore, we further propose a multiple kernel-based learning method. By doing so, our model can learn both linear and nonlinear similarity information, and automatically choose the most suitable kernel. As a result, our model is capable of learning complete similarity information hidden in data set. Comprehensive experimental evaluations of our algorithms on clustering and recommender systems demonstrate its superior performance compared to other state-of-the-art methods. This performance also shows the great potential of our proposed algorithm for other possible applications.

1. Background and motivation

Similarity measurement is an indispensable preprocessing step for a number of data analysis tasks, such as clustering, nearest neighbor classification, and graph-based semi-supervised learning. In many algorithms, the initial data are not needed any more once we obtain similarity information between data points. Therefore, similarity measure is crucial to the performance of many techniques.

One well known fact is that the choice of a particular similarity metric may improve an algorithm's performance on a specific dataset [1]. For example, there are four open issues in the widely used Laplacian matrix construction of a graph [2]: (1) selecting the appropriate number of neighbors, (2) choosing the appropriate similarity metric to measure the affinities among sample points, (3) making algorithms robust to noise and outliers, (4) determining the appropriate scale of data. In practical applications, one often adopted strategy is to try different kinds of similarity measure, such as Cosine, Gaussian function, Jaccard metric, and different neighborhood size and parameters [3]. However, this approach is time consuming and impractical for large scale data. Kar and Jain [4] proposes a framework to measure the goodness of similarity metric in classification tasks. Nevertheless, this approach is hard to adapt to different settings.

Even if one tries different similarity metrics, those predefined similarity metrics may still learn incomplete and inaccurate relationships. In recent years, the dimension of data has become increasingly high. Significant work has focused on discovering potential low-dimensional structures of the high-dimensional data. Some of the state-of-the-art methods are locally linear embedding (LLE) [5,6], isomeric feature mapping (ISOMAP) [7], and locality preserving projection (LPP) [8]. Most of these algorithms need to construct an adjacency graph of neighborhood. Traditional similarity measures often fail to consider the local environment of data points. In Fig. 1, for instance, the points near the intersection are pretty close if they are measured by the Euclidean distance; however, they can be from different clusters, which are represented in different colors. In this case, it is unlikely that any standard similarity function will be adequate to capture the local manifold structure precisely.

Notations. In this paper, matrices and vectors are represented by upper case letters and boldfaced lower-case letters, respectively. The ith column of X is denoted by X_i. The ℓ_1-norm of matrix A is defined as the absolute summation of its entries, i.e., $\|A\|_1 = \sum_i \sum_j |a_{ij}|$. The ℓ_2-norm of a vector $\mathbf{x}$ is defined as $\|\mathbf{x}\|^2 = \mathbf{x}^T \cdot \mathbf{x}$. I denotes the identity matrix. $\mathrm{Tr}(\cdot)$ is the trace operator.

* Corresponding author.
E-mail address: Zhao.Kang@siu.edu (Z. Kang).

http://dx.doi.org/10.1016/j.neucom.2017.06.005
0925-2312/

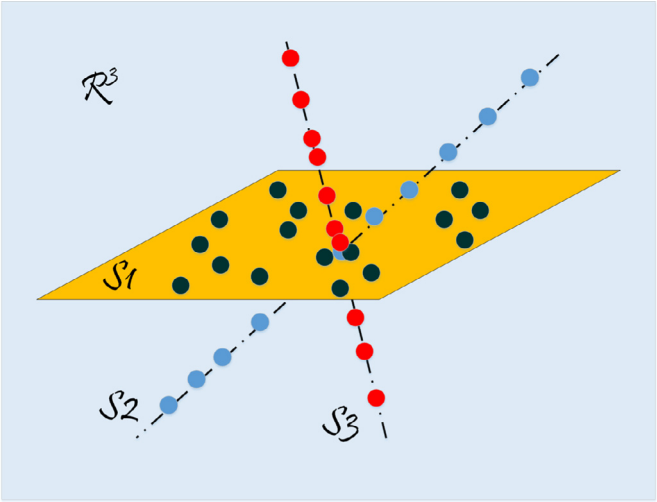

Fig. 1. Three clusters in $\mathcal{R}^3$ denoted in three different colors. Although some points look close, they are from different clusters.

1.1. Introduction to sparse representation

Recently, the pairwise similarity has been learned from the data using a sparse representation, which is regarded as a data-driven approach. According to LLE [5], locally linear reconstruction of a data point by its neighboring points can capture the local manifold structure. This reconstruction can be written as:

$$X_i \approx \sum_{j \in N(i)} X_j a_{ji}, \tag{1}$$

where $N(i)$ represents the neighborhood and a_{ji} is the weight for data point X_j. More similar sample points should receive bigger weights and the weights should be smaller for less similar points. Thus the resulting coefficient matrix A is also called a similarity matrix. In the literature, (1) is also called self-expressive property of the data [54]. Hence the objective function can be formulated as:

$$\min_A \sum_{i=1}^{n} \|X_i - \sum_{j \in N(i)} X_j a_{ji}\|^2. \tag{2}$$

There are usually a number of issues affecting the learning performance with neighborhood-based approaches, such as how to choose a proper size of the neighborhood and what distance to use to measure the closeness. To avoid these problematic issues related to determining neighbors, we relax the requirement that a_{ji} be zero outside neighborhood; in the meantime, as a compensation, we seek a sparse solution of A. It is natural to introduce a regularizer $\|A\|_0$, which counts the number of non-zero elements in A. Recent development in [9,10] has found that the sparse solution could be approximately obtained by solving the ℓ_1 minimization problem, which enjoys the advantage of being continuous and possessing smoothness. By using the l_1 heuristic, then our objective function becomes

$$\min_A \|X - XA\|_F^2 + \lambda \|A\|_1, \quad s.t. \quad A \geq 0, \quad diag(A) = 0. \tag{3}$$

Here, we restrict the reconstruction weights A to be nonnegative for ease of interpretation, and the second constraint is used to avoid the numerically trivial solution ($A = I$). Coefficient λ is to balance the contribution of the sparsity. It is worthy of noting that sparse representation can often lead to resilience noise and outliers [11]. In addition, in Eq. (3), there is no scale consistence restriction for the data points. Therefore, sparse representation helps address both scale inconsistence and outlier issue [2]. On the other hand, sparsity does not encourage locality. Yu et al. [12] has pointed out that locality is more essential than sparsity in some situations.

1.2. Contributions

Although sparsity has shown good performance in various applications, such as subspace recovery [13], denoising [14] and classification [11], the similarity information in the sparse model is learned in the original feature space and Eq. (3) assumes linear relations among data points. Thus, it cannot effectively capture nonlinear relations hidden in the data. For many high-dimensional data in real world, it is often necessary and favorable to model the nonlinearity of data [15]. For instance, face images are assumed from a nonlinear submanifold [16]. Recall that nonlinear data may exhibit linearity when mapped into a higher dimensional feature space via the kernel trick [17]. By doing so, we can use (3) model to capture the linear relations in the transformed space, and thus the nonlinear relations in the original data space.

In this paper, we first extend model (3) to kernel spaces so as to learn underlying nonlinear relations of a given data set. By doing so, we then need to address a relevant problem. It is well known that the type of kernels plays an important role in the performance of kernel methods. How can we find the most appropriate kernel for a given learning task on a specific data? Exhaustive search of a predefined pool of kernels is time-consuming if the pool size is large [18]. To handle this issue, we further propose a multiple kernel learning algorithm. Specifically, it learns simultaneously data similarity and an appropriate linear combination of multiple input kernels. An iterative optimization procedure is developed to mutually reinforce similarity learning and consensus kernel construction. Furthermore, there is an important benefit by leveraging a consensus kernel from multiple kernels: it is of great potential to integrate complementary information from heterogeneous sources or at different scales, which in turn improves the performance of a single kernel based method [19]. After constructing our models, we develop their optimization algorithms. In particular, the optimal weights for kernels have closed-form solutions. We then apply the proposed methods to clustering problem and recommender systems to demonstrate their high potential for real applications.

The rest of the paper is organized as follows. We first introduce the related works in Section 2. The single kernel-based learning method (SKLM) is described in Section 3. Section 4 provides the multiple kernel-based learning method (MKLM). Experiments on clustering and Top-N recommendation are presented in Sections 5 and 6, respectively. Finally, we provide some concluding remarks in Section 7.

2. Related work

In the literature, there are a number of works that combine sparse models with kernel methods [20–25]. Qi and Hughes [20] exploits kernel-based sparse representation in the context of compressive sensing. This idea has also been proposed for supervised learning tasks such as image classification [24–26], face recognition [25], object recognition [23], and kernel matrix approximation [21]. Unfortunately, an a priori dictionary must be known. The authors in [23–25] involve a dictionary learning step. Thiagarajan et al. [22] develops a framework for multiclass object classification, where their kernel weights are tuned according to graph-embedding principles and the associated optimization problem is non-convex and computationally expensive.

Unlike previous work, we need no predefined dictionary, and the kernel weights have closed-form solutions. Moreover, we are guaranteed to obtain the optimal solution due to the convexity of our objective function. With our multiple kernel learning algorithm, the choice of the most appropriate kernel would be