Neurocomputing 113 (2013) 195–203

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Learning mid-perpendicular hyperplane similarity from cannot-link constraints

Shan Gao, Chen Zu, Daoqiang Zhang*

Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

ARTICLE INFO

Article history:
Received 14 September 2010
Received in revised form
4 December 2012
Accepted 12 January 2013
Communicated by S. Choi
Available online 28 February 2013

Keywords:
Mid-perpendicular hyperplane similarity (MPHS)
Pairwise constraints
Semi-supervised
Kernel *k*-means

ABSTRACT

Pairwise constraints known as must-link and cannot-link constraints have been frequently used in semi-supervised clustering. In this paper, we propose a novel usage of cannot-link constraints and develop a method called *Mid-Perpendicular Hyperplane Similarity* (MPHS) for semi-supervised clustering. Since a cannot-link constraint means that the two objects linked by it are not in the same class, there is a mid-perpendicular hyperplane to distinguish them. For each cannot-link constraint, we first compute the corresponding mid-perpendicular hyperplane and then use distances of objects to this hyperplane to learn a new data representation and similarity matrix. Finally, we combine all the similarity matrices from all cannot-link constraints into single similarity matrix and perform kernel *k*-means on it to obtain the partition. We implement MPHS for two cases, i.e., a simple one performed in original input space when the data set is nearly linear-separable, and an advanced one in kernel-induced feature space when the data set is complex and nonlinear-separable. Experimental results on several UCI data sets and some image data sets show the effectiveness of our method.

1. Introduction

Semi-supervised learning which learns from both labeled and unlabeled data has attracted considerable interests in recent years. According to specific tasks, semi-supervised learning can be roughly categorized into semi-supervised classification, semi-supervised regression and semi-supervised clustering [21]. In this paper, we focus on semi-supervised clustering with supervision information in the form of pairwise constraints which specify whether two objects belong to the same class or not, known as the must-link constraints and the cannot-link constraints respectively. Especially, we are more interested in cannot-link constraints and want to investigate its values in improving clustering performances.

Considerable research has been proposed to use pairwise constraints for aiding clustering. Roughly speaking, there are three main ways to use pairwise constraints in clustering: (1) constraints-based [15,7], (2) distance-based [16,8,6] and (3) the hybrid methods [2], etc. In the first category, pairwise constraints are used to guide the clustering process. For example, Wagstaff et al. [15] proposed constrained *K*-means to make sure that there is no constraint-violation in each iteration. Kulis et al. [7] get a unified form of kernel *K*-means and formulated it as kernel matrix with pairwise constraints.

On the other hand, distance-based semi-supervised clustering usually directly or indirectly (e.g., through dimensionality reduction [14,19,18,5] or feature selection [20,13,17]) learns a distance metric. A lot of recent methods for semi-supervised clustering belong to this category. For example, Tang et al. [14] proposed a constraint-guided feature projection to represent the original data in a low dimensional space. Zhang et al. [19] developed a dimensionality reduction method based on the idea that objects with cannot-link constraints should come more far away while objects with must-link constraints should be closer after the transformation. Xing et al. [16] used pairwise constraints to learn a Mahalanobis distance metric. Oyama et al. [12] used only cannot-link constraints to learn a metric matrix, and applied it to name disambiguation problem. Hoi et al. [6] use graph Laplacian to develop a nonparametric kernel method, which is formulated into the SDP problem. Li et al. [8] consider the problem of pairwise constraints propagation, and formulated it into the SDP problem to learn a kernel matrix. Furthermore, to overcome the computational problem, they also proposed to adapt the spectral embedding to make it consistent with the pairwise constraints [9,10]. More recently, Lu et al. [11] and Baghshah et al. [1] proposed to use local geometry or topological structure of data in pairwise constraints based metric learning. Finally, besides the constraint-based and distance-based methods, there also exit some hybrid methods which combine the constraints-based and metric learning together [2].

In this paper, we consider a novel usage of cannot-link constraints, which is inspired from the maximum margin hyperplane of SVM [4]. As we know, when a data set is linearly separable, SVM searches the hyperplane that has the maximum margin to reduce the structure risks. However, SVM needs class

* Corresponding author.
E-mail address: dqzhang@nuaa.edu.cn (D. Zhang).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.01.002

labels to train a classifier for getting such hyperplane. Instead, we are concerned on how to get a similar hyperplane from pairwise constraints.

Intuitively, pairwise cannot-link constraints provide a natural way for getting such a hyperplane. Given two objects with a cannot-link constraint, we can compute its mid-perpendicular hyperplane which is perpendicular to the line across the two data points. Obviously, for a data set with only two objects consisting a cannot-link constraint, the maximum margin hyperplane of SVM is also the mid-perpendicular hyperplane from the cannot-link constraint. Since usually there are more than one cannot-link constraints, we can obtain multiple mid-perpendicular hyperplanes. Now the problem is how can we best use those mid-perpendicular hyperplanes? To address that problem, in this paper, we propose a novel method called *Mid-Perpendicular Hyperplane Similarity* (*MPHS*), which first represents data with distances to each mid-perpendicular hyperplane (gotten from corresponding cannot-link constraint) and then learns a aggregated similarity from those different representations. The main advantages of the proposed MPHS method are listed as below.

(1) It provides a novel usage of pairwise cannot-link constraints, i.e., representing data with distances to mid-perpendicular hyperplanes gotten from cannot-link constraints. To the best of our knowledge, this kind of study was not investigated previously.

(2) It can be used for semi-supervised clustering alone. We have developed two variants of MPHS, i.e., MPHS-linear (for simple and well-structured data) which is performed on original data space and MPHS-Gauss (for complex data) which is performed on Gaussian-kernel induced feature space.

(3) It can also be used for semi-supervised clustering together with other similarity-based methods. Specifically, we have developed a variant of MPHS, i.e., MPHS-PCP which first learns a data-dependent kernel similarity (PCP-kernel [8]) and then perform MPHS in PCP-kernel induced feature space.

(4) The experimental results on a series of data sets show that our method (MPHS-PCP) achieves better performances than existing semi-supervised clustering methods. Moreover, MPHS has a lower computational complexity compared with most semi-supervised clustering methods.

The rest of this paper is organized as follows: in Section 2, we introduce our MPHS methods in detail including MPHS-linear, MPHS-Gauss and MPHS-PCP. Section 3 discuss the experimental results on several real data sets. Finally, in Section 4, we conclude this paper.

2. The MPHS method

In this section, we first describe the main idea of the proposed MPHS method, and then derive the three variants, i.e., MPHS-linear, MPHS-Gauss and MPHS-PCP. Among them, the former one is for linear condition, while the latter two are for non-linear condition. The difference between MPHS-Gauss and MPHS-PCP lies in that the former uses the fixed Gaussian kernel, while the latter learns a data-dependent kernel using PCP [11].

2.1. Main idea

Given a data set of n objects $\mathcal{X}=\{x_1,x_2,\ldots,x_n\}$, a must-link constraint set $\mathcal{M}=\{(x_i,x_j)\}$, a cannot-link constraint set $\mathcal{C}=\{(x_i,x_j)\}$, MPHS contains three main steps. First, it learns a new data representation using the mid-perpendicular hyperplane corresponding to each cannot-link constraint, which can also be regarded as dimensionality reduction. Second, it learns individual similarity matrix according to the new data representation corresponding to each cannot-link constraint. Finally, the individual similarity matrices are aggregated into a similarity matrix and then perform kernel k-means [4] on it.

MPHS is formulated for two conditions. When the data set is simple and well-structured (linearly separable), MPHS can be applied directly on it. Otherwise, we use a (fixed or data-dependent) kernel to transform data into high-dimensional feature space, where MPHS is then performed. For simplicity, here we illustrate our idea for linear condition only. We first give the definition of mid-perpendicular hyperplane as below.

Definition. Let x_i and x_j be two points in some space, $w=x_i-x_j$, $b=(x_i+x_j)/2$. The mid-perpendicular hyperplane determined by x_i and x_j is a hyperplane that has w being its normal vector and the point b on it.

Here we consider each mid-perpendicular hyperplane one by one instead of directly combining them. Note that a cannot-link constraint implies that the two objects are not in the same class, and the mid-perpendicular hyperplane may distinguish between them. Furthermore, we assume that this mid-perpendicular hyperplane might also be a good hyperplane to distinguish the two classes which the two objects respectively belong to. Given a mid-perpendicular hyperplane, for any two data objects we calculate the dissimilarity between them if they are at different sides of the hyperplane and calculate the similarity if they are at the same side.

To better illustrate our idea, let us consider the classification task on a toy data set shown in Fig. 1. Suppose that the data set is linear-separable. The line linking A(0.1,0.3) and B(0.7,0.9) is a cannot-link constraint. L ($y=-x+1$) is the mid-perpendicular hyperplane of A and B, such that all points are divided into two parts. Then we make a projection that each point is projected on the direction of vector AB, which can be obtained by calculating the distances of all points to L (allow the negative distance). For example, the distance of A to L is -0.4242, the distance of B is 0.4242, and that of C is -0.2121. Thus the dimensionality is reduced to be one. We say the left-hand points are not similar with the right-hand points, so we calculate the dissimilarity between them, for example, the dissimilarity between B and C can be $-|-0.2121-0.4242|=-0.6363$. On the other hand, we calculate the similarity between points within the same parts of the hyperplane, for example, the similarity between A and C can be $+|-0.2121-(-0.4242)|=0.2121$. Thus, we can get a similarity matrix from this cannot-link constraint. Since there are more

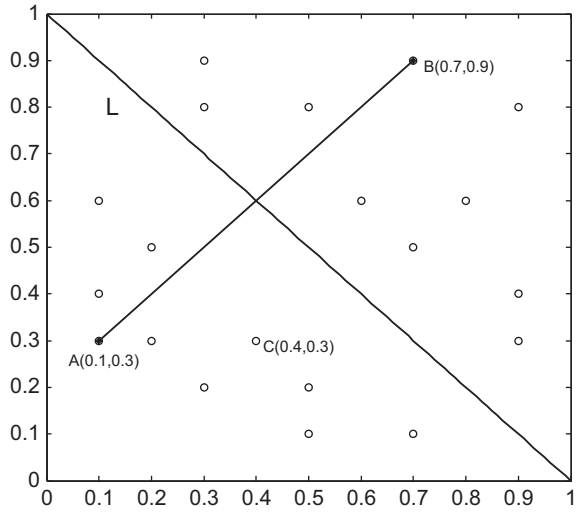

Fig. 1. A toy problem for illustration of how to use the mid-perpendicular hyperplane of a cannot-link constraint.