Neurocomputing 139 (2014) 130–137

Contents lists available at ScienceDirect

Neurocomputing

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

Semi-supervised classification with pairwise constraints

Chen Gong [a,b], Keren Fu [a], Qiang Wu [b], Enmei Tu [a], Jie Yang [a,*]

[a] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China
[b] School of Computing and Communications, University of Technology, Sydney, Australia

ARTICLE INFO

Article history:
Received 19 September 2013
Received in revised form
26 November 2013
Accepted 7 February 2014
Communicated by D. Tao
Available online 5 April 2014

Keywords:
Semi-supervised learning
Pairwise constraints
Smoothness regularizer

ABSTRACT

Graph-based semi-supervised learning has been intensively investigated for a long history. However, existing algorithms only utilize the similarity information between examples for graph construction, so their discriminative ability is rather limited. In order to overcome this limitation, this paper considers both similarity and dissimilarity constraints, and constructs a signed graph with positive and negative edge weights to improve the classification performance. Therefore, the proposed algorithm is termed as Constrained Semi-supervised Classifier (CSSC). A novel smoothness regularizer is proposed to make the "must-linked" examples obtain similar labels, and "cannot-linked" examples get totally different labels. Experiments on a variety of synthetic and real-world datasets demonstrate that CSSC achieves better performances than some state-of-the-art semi-supervised learning algorithms, such as Harmonic Functions, Linear Neighborhood Propagation, LapRLS, LapSVM, and Safe Semi-supervised Support Vector Machines.

1. Introduction

Semi-supervised learning (SSL) is widely adopted in many situations where the labeled examples are insufficient while the unlabeled examples are extremely abundant. Though these massive unlabeled examples do not have explicit labels, they provide the prior of underlying data distribution, which can support accurate classifications along with the labeled examples.

However, the unlabeled examples should be used properly with certain assumptions, otherwise they may hurt the performance instead. Two commonly adopted assumptions are cluster assumption and manifold assumption [1]. Cluster assumption assumes that the examples of different classes form several well-separated clusters, and the decision boundary falls into the low density area in the feature space. Representative algorithms include Transductive Support Vector Machines (TSVM, [2]), Multiple Kernel TSVM [3], concaVe Semi-supervised Support Vector Machine (VS3VM, [4]), Structural Regularized Support Vector Machines (SRSVM, [5]), and Safe Semi-supervised Support Vector Machines (S4VM, [6]), etc. Methods above are the variants of traditional supervised Support Vector Machines (SVM). The only differences are on the definition of loss function, since the hinge loss employed by traditional SVM cannot be directly applied to the semi-supervised settings.

Manifold assumption postulates that the geometry of data distribution is usually supported by an underlying manifold (*e.g.* Riemannian manifold). The manifold can be described by a graph, of which the examples are represented by vertices and their similarities are measured by weighted edges. Therefore, manifold assumption requires that the labels should vary smoothly on the graph. In other words, if two examples are connected by a strong edge, they tend to share similar labels. Under this assumption, many graph-based semi-supervised learning algorithms have been developed. Zhu et al. proposed Harmonic Functions (HF, [7]) and related it to random walks, electric networks, and spectral graph theory. Zhou et al. developed Local and Global Consistency (LGC, [8]), in which the smoothness of labels are defined by the normalized *graph Laplacian*. Moreover, Spectral graph partitioning [9] formulates SSL as a graph cut problem, which aims to find a partitioning that minimizes the defined objective function. Wang et al. proposed Linear Neighborhood Propagation (LNP, [10]) that assumes that each data point in the graph can be optimally reconstructed by its neighbors. By introducing the manifold regularizer, Belkin et al. proposed the Laplacian Support Vector Machines (LapSVM) and Laplacian Regularized Least Squares (LapRLS). The idea of manifold regularization was successfully adapted to multi-label classification by multiview vector-valued manifold regularization (MV^{3}MR, [11]) and manifold regularized multitask learning (MRMTL, [12]) algorithms. Other typical manifold assumption-based approaches include AnchorGraph [13], Graph Transduction via Alternative Minimization (GTAM, [14]), and Label Propagation through Sparse Neighborhood (LPSN, [15]), etc. In recent years, some hypergraph-based manifold learning algorithms were developed and adopted to solve the critical

[*] Corresponding author.
E-mail address: jieyang@sjtu.edu.cn (J. Yang).

http://dx.doi.org/10.1016/j.neucom.2014.02.053
0925-2312/© 2014 Elsevier B.V. All rights reserved.

problems in computer vision, such as image classification [16–18] and cartoon animation [19].

However, the graph established in the methods above only contains nonnegative edge weights. That is, only the similarities between examples are considered for classification, and the dissimilarity information is ignored. However, we believe that the dissimilarity information is important for improving the discriminative ability of semi-supervised classifiers. Therefore, this paper aims to design a novel semi-supervised classifier that incorporates both similarity and dissimilarity constraints between examples. In contrast to the traditional graph-based methods which require edge weights to be nonnegative, the weights in our algorithm are in the range $[-1,1]$. The positive weights representing “must-links” describe how similar the two connected examples are, and the negative weights standing for “cannot-links” evaluate the dissimilarity between the pairwise examples.

Actually, pairwise constraints including “must-links” and “cannot-links” have been widely adopted by various constrained clustering [20–22], dimensional reduction [23] and metric learning algorithms [24,25]. However, they are seldom employed to solve the semi-supervised classification problems. In this paper, pairwise constraints are adopted in order to improve the performance of traditional graph-based SSL algorithms, and the proposed classifier is named as Constrained Semi-supervised Classifier (CSSC). The most relevant work is [26], which also incorporates the dissimilarity into the framework of manifold regularization. However, the negative edges in this method should be manually generated among the unlabeled examples, which is different from CSSC that automatically constructs the graph of signed edges without any manual assistance.

The main contributions of this paper are summarized below:

1. A novel semi-supervised classification algorithm is proposed by incorporating both similarity and dissimilarity constraints.
2. The graph is built via similarity/dissimilarity propagation, in which the constraints imbalance is particularly considered.
3. A convex regularization framework is developed, so that the obtained solution is globally optimal.

The remainder of this paper is organized as follows: Section 2 constructs the signed graph with positive and negative edge weights. Section 3 derives the regularization framework of CSSC based on the established graph. We prove the convexity of the proposed model in Section 4, and present the empirical validations of CSSC and other experimental results in Section 5. Finally, a conclusion is drawn in Section 6.

2. Graph construction

For convenience, some important notations used in the rest of the paper are listed in Table 1. Given l labeled examples $\mathcal{L} = (\mathbf{x}_1,\mathbf{Y}_1),\ (\mathbf{x}_2,\mathbf{Y}_2),\dots,(\mathbf{x}_l,\mathbf{Y}_l)\} \in \mathbb{R}^d \times \mathbb{R}^C$ and u unlabeled examples $\mathcal{U} = \{(\mathbf{x}_{l+1},\mathbf{Y}_{l+1}),\ (\mathbf{x}_{l+2},\mathbf{Y}_{l+2}),\dots,(\mathbf{x}_n,\mathbf{Y}_n)\} \in \mathbb{R}^d \times \mathbb{R}^C \quad (n=l+u)$ drawn from the same distribution, the task of SSL is to propagate the labels $\{\mathbf{Y}_i\}_{i=1}^{l} \in \mathbb{R}^{1\times C}$ in $\mathcal{L}$, to the unknown labels $\{\mathbf{Y}_i\}_{i=l+1}^{l+u} \in \mathbb{R}^{1\times C}$ in $\mathcal{U}$. Here C is the total number of classes. Then the c'-th $(1 \le c' \le C)$ element of label vector $\{\mathbf{Y}_i\}_{i=1}^{n}$ is defined as $(\mathbf{Y}_i)_{c'} = 1$ if $\mathbf{x}_i$ belongs to the c'-th class, and $(\mathbf{Y}_i)_{c'} = 0$ otherwise. Consequently, a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ can be built where $\mathcal{V}$ is the vertex set composed of all the elements in $\mathcal{L} \bigcup \mathcal{U}$, and $\mathcal{E}$ is the edge set describing the similarity/dissimilarity between pairs of examples.

Table 1
Important notations used in this paper.

Notation	Description	Notation	Description
$\mathbf{x}_i$	The ith example	$\widehat{\mathbf{W}}$	The adjacency matrix of $\mathcal{G}$
$\mathbf{Y}_i$	The label vector of $\mathbf{x}_i$	$\mathbf{W}$	The matrix recording the values of $\lvert(\widehat{\mathbf{W}})_{ij}\rvert$
K	The number of neighborhoods	$\mathbf{S}$	Indicator matrix
$\tilde{\mathcal{G}}$	Unsigned graph	$\mathbf{I}$	Identity matrix
$\mathcal{G}$	Signed graph	$\mathbf{F}$	The obtained label matrix
$\mathbf{M}$	The adjacency matrix of $\tilde{\mathcal{G}}$	$\mathbf{H}$	Hessian matrix
$\overline{\mathbf{M}}$	Normalized $\mathbf{M}$	$\tilde{\mathbf{L}}$	Generalized graph Laplacian

Traditionally, there are two ways to compute the nonnegative edge weight between two examples. One is the 0–1 weight, which simply takes the binary value from $\{0,1\}$ to indicate whether an edge exists between the two vertices or not. The other is to use the RBF kernel, which produces a real value within [0, 1], to represent the similarity of examples. However, these two methods only generate nonnegative weights, so they are not suitable to represent both “must-link” and “cannot-link” constraints. Below we introduce a two-step approach called “balanced constraints propagation” to explicitly construct a graph with edge weights in the range of $[-1,1]$.

In the first step, we establish a traditional unsigned graph $\tilde{\mathcal{G}}$ with nonnegative edge weights. K nearest neighborhood (K-NN) graph is adopted because sparse graph usually leads to better performance [27]. The edge weights m_{ij} $(1 \le i,j \le n)$ of $\tilde{G}$ are computed by using the RBF kernel $m_{ij} = \exp(\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$ (σ is the kernel width), and thus we have the adjacency matrix $\mathbf{M}$ of $\tilde{\mathcal{G}}$ with $(\mathbf{M})_{ij} = m_{ij}$. Moreover, we define a diagonal matrix $\tilde{\mathbf{D}}$ in which the i-th diagonal element $\tilde{d}_{ii}$ is calculated as $\tilde{d}_{ii} = \sum_{j=1}^{n} m_{ij}$. Therefore, $\mathbf{M}$ can be further normalized by $\overline{\mathbf{M}} = \tilde{\mathbf{D}}^{-1/2}\mathbf{M}\tilde{\mathbf{D}}^{-1/2}$, so that the elements $\overline{m}_{ij}$ of $\overline{\mathbf{M}}$ satisfy $\sum_{j=1}^{n} \overline{m}_{ij} = 1$ for $1 \le i \le n$ [8].

In the second step, we aim to build a signed graph $\mathcal{G}$ that incorporates both positive and negative constraints based on $\overline{\mathbf{M}}$. It is obvious that $l(l-1)/2$ definitely correct constraints are already available based on the l labeled examples, and they are recorded by the similarity set $\mathcal{S}$ and dissimilarity set $\mathcal{D}$:

$$\mathcal{S} = \{(\mathbf{x}_i,\mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ come from the same class}\}$$

$$\mathcal{D} = \{(\mathbf{x}_i,\mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ come from different classes}\}.$$

The aim of our proposal is to propagate the limited available elements in $\mathcal{S}$ and $\mathcal{D}$, to the remaining pairs of examples. This process is called “balanced constraints propagation”.

To facilitate the mathematical manipulations, we use the matrix $\widehat{\mathbf{W}}^{(0)} \in \mathbb{R}^{n\times n}$ to encode the pairwise constraints in $\mathcal{S}$ and $\mathcal{D}$, namely

$$(\widehat{\mathbf{W}}^{(0)})_{ij} = \widehat{\mathbf{W}}_{ij}^{(0)} = \begin{cases} 1 & (\mathbf{x}_i,\mathbf{x}_j) \in \mathcal{S} \text{ or } i=j \\ -\gamma & (\mathbf{x}_i,\mathbf{x}_j) \in \mathcal{D} \\ 0 & (\mathbf{x}_i,\mathbf{x}_j) \text{ is not specified} \end{cases} . \tag{1}$$

In (1), $\gamma = a/b$ where $a = |\mathcal{S}|+n$, $b = |\mathcal{D}|$ and $|\cdot|$ represents the size of a set. Note that we set $\widehat{\omega}_{ij}^{(0)} = -\gamma$ rather than -1 to avoid the constraints imbalance. In fact, if $|\mathcal{D}|$ is very small, the element “1” in $\widehat{\mathbf{W}}^{(0)}$ will be much more than the element “0” because all the diagonal elements are 1s, thus the “must-links” may dominate the propagation process, which significantly weakens the propagation “strength” of the “cannot-links”. Alternatively, more negative constraints can be added to $\widehat{\mathbf{W}}^{(0)}$ based on the prior knowledge, so the negative constraints can significantly outnumber the positive constraints sometimes. Therefore, γ assigns larger weight