Optik 127 (2016) 376–382

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Semi-supervised behavioral learning and its application

Chun Zhang[a,b,*], Shafei Wang[c], Dongsheng Li[a,b], Junan Yang[a,b], Jiyang Zhang[d]

[a] Electronic Engineering Institute, Hefei, China
[b] Key Laboratory of Electronic Restriction, Hefei, China
[c] The Northern Institute of Electronic Equipment of China, Beijing, China
[d] Jiangnan Institute of Computing Technology, Wuxi, China

ARTICLE INFO

Article history:
Received 1 January 2015
Accepted 12 October 2015

Keywords:
Semi-supervised learning
Behavioral learning
Regularization
Manifold learning
Support vector machines

ABSTRACT

Semi-supervised learning has attracted significant attention in pattern recognition and machine learning. Among these methods, a very popular type is semi-supervised support vector machines. However, parameter selection in heat kernel function during the learning process is troublesome and harms the performance improvement of the hypothesis. To solve this problem, a novel local behavioral searching strategy is proposed for semi-supervised learning in this paper. In detail, based on human behavioral learning theory, the support vector machine is regularized with the un-normalized graph Laplacian. After building local distribution of feature space, local behavioral paradigm considers the form of the underlying probability distribution in the neighborhood of a point. Validation of the proposed method is performed with toy and real-life data sets. Results demonstrate that compared with traditional method, our method can more effectively and stably enhance the learning performance.

1. Introduction

Semi-supervised learning has attracted a significant interest in pattern recognition and machine learning. It exploits unlabeled data in addition to the limited labeled ones to improve the learning performance [1]. Many semi-supervised learning algorithms have been proposed during the past decade, among which a very popular type of algorithms is the semi-supervised support vector machines (S3VMs).

Examples of this type include the semi-supervised SVM [2], the transductive SVM (TSVM) [3], and the Laplacian SVM [4]. S3VM and the TSVM are built upon the cluster assumption and use the unlabeled data to regularize the decision boundary. Specifically, these methods prefer the decision boundary to pass through low-density regions [5]. The Laplacian SVM is a S3VM that exploits the data's manifold structure via the graph Laplacian. It encodes both the labeled and unlabeled data by a connected graph, where each instance is represented as a vertex and two vertices are connected by an edge if they have large similarity. The goal is to find class labels for the unlabeled data such that their inconsistencies with both the supervised data and the underlying graph structure are minimized.

However, while many efficient SVM methods have been developed for supervised learning, S3VMs still suffer from inefficiency issues. In particular, the optimization problem of Bennett and Demiriz's S3VM is formulated as a mixed-integer programming problem and so is computationally intractable in general. TSVM, on the other hand, iteratively solves standard supervised SVM problems. However, the number of iterations required may be large since the TSVM is based on a local combinatorial search that is guided by a label switching procedure. Unlike the TSVM, the Laplacian SVM focuses on regularization in reproducing Kernel Hilbert spaces and only needs to solve one small SVM with the labeled data. But Laplacian SVM utilizes heat kernel weights to form edge weights when constructing data adjacency graph. The performance of heat kernel weights highly depends on parameter selection and how to exactly fix parameter in different applications may be troublesome.

Currently, Defense Advanced Research Projects Agency (DARPA) is soliciting innovative research proposals in the area of machine learning for electronic warfare applications and sets up the Behavioral Learning for Adaptive Electronic Warfare (BLADE) program [6] in 2010. At the same time, more and more research fruits hold the viewpoint that human behavioral learning can effectively improve the performance of machine learning [7–9]. Inspired by these booming trends, we propose a novel approach called Local Behavioral-based Laplacian SVM (LB-LapSVM) to overcome the problem of parameter selection in Laplacian SVM.

The rest of this paper is organized as follows. Section 2 briefly describes the semi-supervised learning framework and its extension in Laplacian SVM. Section 3 presents the local behavioral searching method and its utility in LB-LapSVM. Section 4 performs extensive experiments on toy data sets and real communication

* Corresponding author. Tel.: +8613956049657; fax: +86055163415690.
E-mail address: zhangchuncwins@163.com (C. Zhang).

http://dx.doi.org/10.1016/j.ijleo.2015.10.089
0030-4026/

radio data sets. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

2. Related work

Inspired by the success of large margin principle, S3VMs are extensions of supervised SVMs to semi-supervised learning by simultaneously learning the optimal hyperplane and the labels for unlabeled instances. It was disclosed that S3VMs realize the low-density assumption by favoring the decision boundary going across low-density regions.

2.1. Semi-supervised learning framework

Formally, considering binary classification in semi-supervised learning, we are given a set of l labeled samples $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, and a set of u unlabeled samples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, where $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$. Let us now assume a general-purpose decision function f. The regularized functional to be minimized is defined as:

$$f^* = \underset{f \in \mathcal{H}_K}{arg\,min} \frac{1}{l}\sum_{i=1}^{l} V(\mathbf{x}_i, y_i, f) + \gamma_A \left\|f\right\|_K^2 + \gamma_I \left\|f\right\|_I^2 \quad (1)$$

where V represents a generic cost function of the committed errors on the labeled samples, $\mathcal{H}_K$ is a reproducing kernel Hilbert space (RKHS) induced by the kernel. γ_A controls the complexity of f in the associated Hilbert space $\mathcal{H}_K$, and γ_I controls its complexity in the intrinsic geometry of the marginal data distribution. For example, if the probability distribution is supported on a low-dimensional manifold, $\left\|f\right\|_I^2$ penalizes f along that manifold I. Note that this functional constitutes a general regularization framework that takes into account all the available knowledge.

2.2. Laplacian SVM

The previous semi-supervised learning framework allows us to develop many different algorithms just by playing around with the loss function, V, and the regularizes, $\left\|f\right\|^2$. In this paper, we focus on the Laplacian SVM formulation, which basically uses a SVM as the learner core and the graph Laplacian for manifold regularization.

2.2.1. Cost function of the errors

The Laplacian SVM uses the same hinge loss function as the traditional SVM:

$$V(\mathbf{x}_i, y_i, f) = \max\{0, 1 - y_i f(\mathbf{x}_i)\} \quad (2)$$

where f represents the decision function implemented by the selected classifier.

2.2.2. Decision function

We use as the decision function $f(\mathrm{x}_*) = \left\langle \mathrm{w}, \phi(\mathrm{x}_*)\right\rangle + b$, where $\phi(\cdot)$ is a nonlinear mapping to a higher (possibly infinite) dimensional Hilbert space $\mathcal{H}$, and $\mathbf{w}$ and b define a linear regression in that space. By means of the Representer Theorem [10], weights $\mathbf{w}$ can be expressed in the dual problem as the expansion over labeled and unlabeled samples $\mathbf{w} = \sum_{i=1}^{l+u}\alpha_i\phi(\mathbf{x}_i) = \mathbf{\Phi}\boldsymbol{\alpha}$, where $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_{l+u})]^{\mathrm{T}}$ and $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_{l+u}]$. Then, the decision function is given by:

$$f(\mathbf{x}_*) = \sum_{i=1}^{l+u}\alpha_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}_*) + b \quad (3)$$

and $\mathbf{K}$ is the kernel matrix formed by kernel functions, $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\right\rangle$. The key point here is that, without considering the mapping ϕ explicitly, a non-linear classifier can be constructed by selecting the proper kernel. Also, the regularization term can be fully expressed in terms of the kernel matrix and the expansion coefficients:

$$\left\|f\right\|_K^2 = \|\mathbf{w}\|^2 = (\mathbf{\Phi}\boldsymbol{\alpha})^{\mathrm{T}}(\mathbf{\Phi}\boldsymbol{\alpha}) = \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{K}\boldsymbol{\alpha} \quad (4)$$

2.2.3. Manifold regularization

The geometry of the data is modeled with a graph in which nodes represent both labeled and unlabeled samples connected by weights $\mathbf{W}_{ij}$. Regularizing the graph follows from the smoothness (or manifold) assumption and intuitively is equivalent to penalize the "rapid changes" of the classification function evaluated between close samples in the graph:

$$\left\|f\right\|_I^2 = \frac{1}{(l+u)^2}\sum_{i,j=1}^{l+u}\mathbf{W}_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 = \mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f} \quad (5)$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian, $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{W}$ given by $D_{ii} = \sum_{j=1}^{l+u}\mathbf{W}_{ij}$, and $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{l+u})]^T = \mathbf{K}\boldsymbol{\alpha}$, where we have deliberately dropped the bias term b.

The LapSVM algorithm is summarized in the Table 1.

From Table 1, we can clearly see that when LapSVM computes edge weights, heat kernel weights $\mathbf{W}_{ij} = e^{-\left\|x_i - x_j\right\|^2/4t}$ are utilized. The value of t varies in different applications. Unsuitable value of t can lead to a serious decline in performance. What's more, when computing edge weights, heat kernel weights only focus on sample $\mathbf{x}_i$ and $\mathbf{x}_j$. However, plenty additional information remained to improve affinity measuring. From the above analysis, we proposed a novel method called LB-LapSVM.

3. Local behavioral searching

In many real-life situations, human are exposed to a combination of labeled data and far more unlabeled data when they need to make a classification decision. Understanding how human combine information from labeled and unlabeled data to draw inferences about conceptual boundaries can have significant social impact. In the realistic setting where labeled and unlabeled data are available, semi-supervised learning offers very explicit computational hypotheses that can be empirically tested in the laboratory. To help understand description in this section, we start by providing a "translation" of relevant terms from semi-supervised learning to human behavioral learning:

That is, when stimulus arrives, human use their supervised experiences from teachers and passive experiences from nature to complete concept learning task. During the learning task, human take in to account of some mechanism in their mind. When concept category is obtained, the learning task is done.

Inspired by above analysis, we applied human behavioral learning strategy to LapSVM. In detail, when computing affinity between sample $\mathbf{x}_i$ and $\mathbf{x}_j$, LapSVM only focuses on sample $\mathbf{x}_i$ and $\mathbf{x}_j$. However, it's not the case in human behavior. Given a pair of samples, in human empirical cognition, local neighborhood of this pair samples plays an important role in affinity measuring. Based on local distribution of feature space, human behavioral paradigm considers the form of the underlying probability distribution in the neighborhood of a point.

Instead of selecting a single parameter t in heat kernel weights, we propose to calculate local behavioral parameters σ_i and σ_j for data points $\mathbf{x}_i$ and $\mathbf{x}_j$. The distance from $\mathbf{x}_i$ to $\mathbf{x}_j$ as "seen" by $\mathbf{x}_i$ is $d(\mathbf{x}_i, \mathbf{x}_j)/\sigma_i$ while the converse is $d(\mathbf{x}_j, \mathbf{x}_i)/\sigma_j$. Therefore, the square distance d^2 may be generalized as $d(\mathbf{x}_i, \mathbf{x}_j)d(\mathbf{x}_j, \mathbf{x}_i)/\sigma_i\sigma_j = d^2(\mathbf{x}_i,$