Neurocomputing 167 (2015) 112–121

Contents lists available at ScienceDirect

Neurocomputing

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

A general framework for co-training and its applications

Weifeng Liu [a], Yang Li [a], Dapeng Tao [b,*], Yanjiang Wang [a]

[a] College of Information and Control Engineering, China University of Petroleum(East China), Qingdao 266580, PR China
[b] School of Information Science and Engineering, Yunnan University, Kunming 650091, Yunnan, PR China

ARTICLE INFO

Article history:
Received 11 February 2015
Received in revised form
28 April 2015
Accepted 29 April 2015
Communicated by Jun Yu
Available online 14 May 2015

Keywords:
Semi-supervised learning
Co-training
Multi-view
Human action recognition
Social activity recognition

ABSTRACT

Co-training is one of the major semi-supervised learning paradigms in which two classifiers are alternately trained on two distinct views and they teach each other by adding the predictions of unlabeled data to the training set of the other view. Co-training can achieve promising performance, especially when there is only a small number of labeled data. Hence, co-training has received considerable attention, and many variant co-training algorithms have been developed. It is essential and informative to provide a systematic framework for a better understanding of the common properties and differences in these algorithms. In this paper, we propose a general framework for co-training according to the diverse learners constructed in co-training. Specifically, we provide three types of co-training implementations, including co-training on multiple views, co-training on multiple classifiers, and co-training on multiple manifolds. Finally, comprehensive experiments of different methods are conducted on the UCF-iPhone dataset for human action recognition and the USAA dataset for social activity recognition. The experimental results demonstrate the effectiveness of the proposed solutions.

1. Introduction

Currently, it is easy to obtain a massive amount of multimedia data with the fast growth of personal devices such as smart phones and digital cameras. However, most of these multimedia data are unlabeled, and it is difficult and time-consuming to conduct manual annotation. As a consequence, semi-supervised learning [1–3], which attempts to make use of costless and abundant unlabeled data in addition to labeled data to improve the performance, has attracted considerable attention. During the past decade, many semi-supervised learning approaches have been developed, such as generative-based methods, graph-based methods, semi-supervised support vector machines (S3VMs) and co-training [5–7].

Co-training is one of the most attractive paradigms of semi-supervised learning and is also an important part of multi-view learning [8–11], which was first proposed by Blum and Mitchell [5]. In recent years, a great amount of co-training variants under different names have been reported and have achieved great success in many applications, such as natural language processing [12–14], content-based image retrieval (CBIR) [15–19], image classification [20], computer-aided diagnosis [22] and others [13,21,23–24].

One main requirement for Blum and Mitchell's standard co-training is that the dataset can be described by two sufficient and redundant attribute subsets, namely, each view is sufficient to predict the class perfectly and the two views are independently given the class label. For example, web pages [5] can be described by either the test on the web page or the test on the hyperlinks pointing to the web page. Standard co-training works in an iterative manner on two distinct feature sets, namely, two classifiers are first trained using the initial labeled data on two different views and then each classifier is reinforced by the prediction results of unlabeled data in the other view, classifiers are iteratively reinforced until a fixed point is reached or some other stopping criterion is met. Specifically, two classifiers teach each other on two views to improve the classification performance.

In many practical situations, it is not intuitively obvious how to obtain two sufficient and redundant natural feature sets; hence, many co-training variants with other assumptions that guarantee its success have been proposed to relax the sufficient and redundant assumptions. Furthermore, many real-world datasets only have a single view instead of two views; therefore, some variants of co-training that do not require two views have been developed successively, and some empirical studies have shown that these co-training algorithms still work well. A key to the success of these co-training algorithms is to generate different learners by exploiting different techniques: one learner to help improve the accuracy of the other by providing it with unknown information. By taking advantage of the correlations between the learners, many co-training algorithms show their effectiveness. However, very little work has been performed to bring these methods into a unified framework. As a result, the common essence and differences of these algorithms are not completely clear. Therefore, it is essential and informative to provide a systematic framework

* Corresponding author. Tel.:+86 13560476207.
E-mail address: dapeng.tao@gmail.com (D. Tao).

http://dx.doi.org/10.1016/j.neucom.2015.04.087
0925-2312/

for a better understanding of the common properties and differences in these algorithms. In this paper, we present a simple and general framework in which diverse learners are constructed to learn each other. Specifically, we summarize these approaches in three groups including (1) learning with multiple views [5,25–31], (2) learning with multiple classifiers [32–34] and (3) learning with multiple manifolds [35]. We also conduct extensive experiments on the UCF-iPhone dataset for human action recognition and the USAA dataset for social activity recognition, respectively. The experimental results demonstrate the effectiveness of co-training algorithms.

The rest of this paper is assigned as follows. Section 2 presents some related work. Then, Section 3 introduces a unified framework for co-training and some theoretical analysis. Finally, Section 4 describes the experiments details and some discussions followed by a conclusion in Section 5.

2. Related work

A key reason for co-training algorithm success is to train multiple learners by exploiting different techniques and combining their predictions to learn each other and to decrease classification error. In this section, we review the previous work related to co-training and summarize these approaches in three groups including (1) learning with multiple views, (2) learning with multiple classifiers and (3) learning with multiple manifolds.

2.1. Co-training on multiple views

In a few special applications, the dataset has natural disjoint subsets of attributes, *i.e.*, web page classification [5]. In most real-world applications, the datasets have only one attribute set as opposed to two. Methods that have artificial and manual feature splitting are developed to take advantage of the interaction between multiple learners. Natural or artificial feature sets are called *view*.

Standard co-training was applied in domains with truly sufficient and independent feature splits. The procedure is simple, and it works as follows. Two classifiers with reasonable performance can be built using the original labeled data on each view separately. Then, two classifiers in a loop label all of the unlabeled examples, and each classifier takes turns in selecting the high-confident predicted examples and adds these into the training set of the other. Later, both classifiers will be refined using the newly added examples provided by the other view. The loop will repeat until a fixed point is reached or some other stopping criterion is met.

Dasgupta et al. [25] justified the sufficiency and independence assumptions and showed that the co-trained classifiers can make fewer generalization errors by maximizing their agreement over the unlabeled data. However, when the data has two views in real-world applications, it is rare that the two views are conditionally independent given the class label. So, several other assumptions on co-training were proposed to relax the two powerful assumptions. Abney [26] showed that the *conditional independence* can be relaxed to *weak dependence* for co-training to work well, and he presented a new co-training algorithm named the *greedy agreement algorithm*. Balcan et al. [27] suggested a weaker assumption called ε*-expansion*, and they theoretically showed that given an approximately strong PAC-learner on the two different views, the ε*-expansion* assumption on the underlying data distribution guarantees the success of co-training.

However, most real-world datasets only have a single attribute set as opposed to two. To exploit the advantages of co-training, effective methods that do not rely on the existence of two views are needed. A straightforward method is to split the attribute sets into two disjoint sets, where the aim is to maximize the disagreement between the two feature subsets and conduct standard co-training based on the manually generated views. Nigam et al. [28] showed that when attribute sets do not have natural feature sets but rather sufficient redundancy, co-training on a random division of the feature set may work well; however, many applications are not described by a large number of attributes. Du et al. [29] proposed four simple heuristic splitting methods to split a single view into two views. Unfortunately, their empirical results showed that view splitting is unreliable when the number of labeled examples is small. Chen et al. [30] proposed a novel feature decomposition algorithm named *pseudo multi-view co-training* (PMC), which automatically divides the features of a single-view data set into two mutually exclusive subsets for co-training to succeed. In addition, Zhou et al. [31] proposed a novel co-training style algorithm called *tri-training* in which three different classifiers are trained on bootstrap sampled labeled examples. The original labeled example set is bootstrap sampled to generate three different training sets, and each training set can be seen as a view.

In summary, for natural or artificial feature sets, each view has a unique feature space, co-training can generate learners with a disagreement on multiple views, and then one learner can use the disagreement and provide the other with unknown information to boost the performance of co-training.

2.2. Co-training on multiple classifiers

As described in Section 2.1, co-training on randomly partitioned views is not always effective, *e.g.*, the effective methods [28–30] that tailor the feature sets for standard two-view co-training is limited. Thus, some methods that use a single view without feature splitting are developed [32–34] by designing co-training on multiple classifiers.

Goldman et al. [32] proposed a co-training algorithm that does not rely on the existence of two views but instead requires different learning algorithms (*e.g.,* decision tree algorithm) to construct classifiers that can partition the input instance space into a set of equivalence classes. Additionally, they used 10-fold cross validation to identify the unlabeled examples to label. Later, they extended another single-view method and named it *Democratic co-training*, where it uses three or more learning algorithms [33] to build multiple classifiers. Wang et al. [34] presented a new PAC analysis on co-training style algorithms, and their theoretical study showed that if the two learners have large differences, the performance can be improved through the co-training process. If the two initial learners have small differences, the performance can be improved when the number of labeled examples is small. Moreover, they analyzed the reason why the performance of the co-training process could not be improved further after a number of rounds. This problem is often encountered in practical applications of co-training.

As a short summary, learning with a single view and two different classification algorithms (*e.g.,* decision tree algorithm, naïve Bayes) is used to generate two different learners. Different learners have different biases, which is an intuitive explanation on why co-training on multiple classifiers can succeed.

2.3. Co-training on multiple manifolds

Manifold regularization tries to explore the geometry of the intrinsic data probability distribution by penalizing the classification function along the implicit manifold. Belkin et al. [36] proposed a manifold regularization framework in reproducing kernel Hillbert space (RKHS). Sindhwani et al. [37] embedded manifold regularization into a semi-supervised kernel defi-