Boosting for Multiclass Semi-Supervised Learning
Tanha, J.; van Someren, M.W.; Afsarmanesh, H.

Published in:
Proceedings ICT.OPEN 2012

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
1 Introduction

Supervised learning methods are effective when there are sufficient labeled instances. In many applications, such as object detection, document and web-page categorization, labeled instances however are difficult, expensive, or time consuming to obtain because they require empirical research or experienced human annotators. Semi-supervised learning algorithms use not only the labeled data but also the unlabeled data to build a classifier. The goal of semi-supervised learning is to use unlabeled instances and combine the information in the unlabeled examples with the explicit classification information of labeled examples for improving the classification performance. Most of the semi-supervised learning algorithms were designed for binary classification problems. However, many practical domains, for example recognition of speech, objects, and characters, involve more than two classes. A Multiclass classification problem can be decomposed into a number of independent binary classification problems by utilizing methods like one-versus-all. However, these schemes have their problems. One-versus-all results in imbalanced distributions. Since each classifier is trained independently, the weights of their outputs may be on different scales, so that combining them is non-trivial. There is thus a need for direct multiclass algorithm for semi-supervised learning.

In this paper we propose a new algorithm for Multiclass semi-supervised learning that follows the boosting approach and is a direct generalization of the binary SemiBoost algorithm [3], which uses both the similarity between the points and the classifier predictions to sample and assign “pseudo-labels” to the unlabeled examples, to the multiclass setting, named as Multiclass SemiBoost. The key advantage of Multiclass SemiBoost is to exploit both the manifold and the cluster assumption to train the classifiers using boosting. We derive the algorithm from an objective function that combines empirical loss on the labeled data and inconsistency of the labels over all data. Multiclass SemiBoost uses an exponential loss function to minimize the inconsistency between labeled and unlabeled data as well as between unlabeled data. Based on this loss function we derive a criterion to sample and assign “pseudo-labels” and estimate confidence for them.

2 Multiclass Loss Function

Two ideas of the proposed algorithm are to maximize both the consistency between data and the margin. For that we design an optimization problem as in (1). In (1) the first term, $F_{lu}$, expresses that the labeled and unlabeled examples with high similarity must share the same label, and the second, $F_{uu}$ expresses that unlabeled examples with high similarity must have the same label. In order to formulate $F_{lu}$ and $F_{uu}$ as an optimization problem, we design an exponential multiclass loss function and introduce an objective function $F(Y, S, H)$ that consists of two terms, $F_{lu}$ and $F_{uu}$, as follows:

$$F(Y, S, H) = C_1 F_{lu}(Y, S^{lu}, H) + C_2 F_{uu}(S^{uu}, H)$$  \hspace{1cm} (1)

where $C_1$ and $C_2$ are the weights for the contribution of the labeled and unlabeled data respectively, $S$ is the pairwise similarity, $H$ is a multiclass ensemble classifier, and $Y$ is a vector of labels where $y_i \in Y$ is a $K$-dimensional vector with all entries equal to $-\frac{1}{K}$ except a 1 in position $k$, which corresponds to the actual class label.

Here, we formulate the objective function for $F_{lu}(Y, S^{lu}, H)$, which minimizes the inconsistency between labeled and unlabeled data, as:

$$F_{lu}(Y, S^{lu}, H) = \sum_{i=1}^{n_l} \sum_{j=1}^{n_u} S^{lu}(x_i, x_j) \exp\left(-\frac{1}{K} Y_i H(x_j)\right)$$  \hspace{1cm} (2)

where $n_l, n_u$, and $K$ are the number of labeled, unlabeled data, and classes respectively.

The second term of (1) measures the inconsistency between the unlabeled examples using the similarity matrix and “pseudo-labels” for the unlabeled examples as follows:

$$F_{uu}(S^{uu}, H) = \sum_{i,j \in n_u} S^{uu}(x_i, x_j) \exp\left(\frac{1}{K} (H(x_i) - H(x_j))\right)$$  \hspace{1cm} (3)
As a result, function (1) measures the overall consistency between the data. The goal of the learning algorithm here is to maximize the consistency as represented in (1).

3 Multiclass SemiBoost Algorithm

We assume that a base classifier \( C \) as black box and use the unlabeled data to improve the performance of \( C \). Unlike ASSEMBLE [1] and MarginBoost [2] which use only the classifier predictions, our algorithm exploits both cluster and manifold assumption in training classifiers through the boosting procedure using two regularization terms in loss function. To achieve this goal, the pairwise similarities between data and classifier predictions are used to sample and assign “pseudo-labels” to the unlabeled data at each iteration of the boosting procedure. Then, the high-confidence newly-labeled data along with labeled data are used to construct a new component classifier, which minimizes the loss function. The final classification model is formed as the combination of the generated classifiers with their weights. The outline of the Multiclass SemiBoost algorithm is given in 1.

Algorithm 1 An outline of the Multiclass SemiBoost algorithm

- Initialize: L,U, S,H(x)
- L: Labeled data; U: Unlabeled data
- S: Similarity Matrix; H(x): Ensemble of Classifiers
- At each iteration i:
  - Assign “pseudo-labels” to the unlabeled examples based on the pairwise similarity and classifier prediction
  - Sample the high-confidence examples for a component classifier
  - Build a new component classifier based on both newly-labeled and original labeled examples
  - Update ensemble H
- Generate final hypothesis

4 Results

The results of experiments on the UCI datasets are reported in this section. The columns DT, AdaB, ASEML, SMA, and MCSB give the classification performance of the supervised decision tree classifier (J48), supervised AdaBoost with J48 as base learner, semi-supervised ASSEMBLE, Semi-MultiAdaBoost [4] which is the multiclass version of ASSEMBLE, and Multiclass SemiBoost respectively.

As can be seen in Table 1, there is a clear pattern in the results. AdaBoost is equal to its base learner or better for Decision Tree learning. The semi-supervised algorithms using the unlabeled data improve these results with only a few exceptions. In some cases the difference is quite substantial.

As shown in Table 1, Multiclass SemiBoost significantly improves the classification accuracy for nearly all datasets. It outperforms on almost all datasets and performs better than ASSEMBLE and SemiMultiAdaBoost [4] on 11 out of 11 datasets, and significantly better on 4 of the datasets (e.g. glass, iris, vote, and vowel). ASSEMBLE and SemiMultiAdaBoost improve the performance of 9 out of 11 datasets. There is dataset where the supervised learner or the AdaBoost meta classifier outperforms Multiclass SemiBoost algorithm. In this case the supervised algorithm outperforms all the semi-supervised algorithms, for example the meta classifier AdaBoost outperforms on Wave dataset. These kinds of results emphasize that the unlabeled examples do not guarantee that they always are useful and improve the performance.

References