Beyond Banditron: A Conservative and Efficient Reduction for Online Multiclass Prediction with Bandit Setting Model

Guangyun Chen, Gang Chen, Jianwen Zhang, Shuo Chen, Changshui Zhang
State Key Laboratory of Intelligent Technologies and Systems
Tsinghua National Laboratory for Information Science and Technology(TNList)
Department of Automation, Tsinghua University, Beijing, P.R.China
{cgy08, g-c05, jw-zhang06, chenshuo07}@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn

***Abstract*—In this paper, we consider a recently proposed supervised learning problem, called online multiclass prediction with bandit setting model. Aiming at learning from partial feedback of online classification results, *i.e.* "true" when the predicting label is right or "false" when the predicting label is wrong, this new kind of problems arouses much of researchers' interest due to its close relations to real world internet applications and human cognitive procedure. While some algorithms have been brought forward, we propose a novel algorithm to deal with such problems. First, we reduce the multiclass prediction problem to binary based on *Conservative one-versus-all others Reduction* scheme; Then *Online Passive-Aggressive Algorithm* is embedded as binary learning algorithm to solve the reduced problem. Also we derive a pleasing cumulative mistake bound for our algorithm and a time complexity bound linear to the sample size. Further experimental evaluation on several real world multiclass datasets including *RCV1*, *MNIST*, *20 Newsgroups* and *USPS* shows that our method outperforms the existing algorithms with a great improvement.**

***Keywords*-online multiclass prediction; bandit setting model; one versus all reduction; passive-aggressive algorithm;**

I. Introduction

A new supervised learning problem, the multiclass prediction with bandit setting model, is first proposed by [1]. Unlike the conventional supervised learning paradigm, it focuses on applications in which only partial feedback, instead of full label information, is received by the learner itself. "Partial feedback" as mentioned, means that the learner receives only "right" or "wrong" feedback about its prediction results from some independent oracles, such as people.

Naturally, this kind of paradigm is online. And for most real world internet usages, full label information is hardly revealed. On one hand, as we all know, the internet surfers are too "lazy" to enter true labels even if they have already known them; On the other hand, even the surfer does not know what exactly he or she wants. For example, people put some requiring features to the search engine, but they cannot figure out the definite item type they request. However, a simple click of the mouse responses an approximate inclination of the user, representing kind of "partial feedback".

The rising usages of internet raise new challenges to the traditional learning fields and impose a powerful encouragement of this learning model. Such an online recommender system is mentioned by [1], [2]. It is said that when user makes a query (e.g. requiring song types, commodity features, etc.) to the recommender system, the system gives a suggestion under its former knowledge about the user (e.g. searching records, purchasing records, browsing histories, etc.); Then the user responses to the suggestion(s) by either clicking or not clicking it(them). Nevertheless the system does not learn any about what would happen if it provides other suggestions as substitutions.

Besides, we find that this kind of learning paradigm has something to do with the human cognitive procedure. Consider the puzzle game as an example. If you join in a puzzle game, your partner asks you the *Riddle of the Sphinx*: "Which creature in the morning goes on four legs, at midday on two, and in the evening upon three, and the more legs it has, the weaker it be?" Then he gives you ten choices of animals, including "man", "tiger", "monkey", "bird" etc. You may answer: Tiger. Then your partner replies a "no" as feedback. You may take use of this information and your former thinking procedure to make a new choice. Then you get a reply, and you will think again if it is "no"... At the end, when you reach the answer "man", you will get a "yes" feedback. From then on, if you encounter similar puzzles, you would probably reach the correct answer quickly.

Though the real world applications are more complicated, in this paper, we only focus on the multiclass prediction with bandit setting model as in [1]. Essentially, we formalize the model setting as follows: the learner gets an input feature vector $\mathbf{x_t}$ at each round t; then based on information obtained from the former round, the learner makes a prediction and assigns a label $\hat{y}_t$ to this input; finally, according to its prediction and the true label of the input $\mathbf{x_t}$, the learner receives a limited feedback that whether its prediction is correct or not. In contrast, conventional online supervised learning problem would disclose the true label y_t to the learner at each consecutive round. So with partial feedback, this kind of problems are harder than conventional supervised learning problems.

In order to take advantage of the partial feedback information as completely as possible to build a "good" learner

for future prediction, [1] proposed the *Banditron* Algorithm. Based on multiclass perceptron algorithm, the *Banditron* uses an *exploitation-explore* scheme to handle the difficulties of utilizing the negative feedback. In some rounds where the algorithm explores, it makes a prediction uniformly with probability γ from the full label set instead of the most probable one the learner believes.

Another approach is *Offset Tree reduction* algorithm[2], a recent research work for learning with partial labels, which deals with a more general problem than what we consider. However, since "When solving a given problem, try to avoid solving a more general problem as an intermediate step"[3], our method focuses on the specific problem instead of the general one. When the partial label learning problem requires an interactive setting, the reward for one prediction choice is restricted in 1 and 0 and it's always possible to choose the best action, the partial label learning problem degenerates to the online multiclass prediction with bandit setting problem and the corresponding algorithm is called *Realizable Offset Tree Reduction* algorithm. And *Costing* algorithm[4] is applied to the updating rule according to the partial feedback.

Some other works is related to "multi-armed bandit" problem[5], [6]. Though both dealing with side information, the online multiclass prediction with bandit setting model focuses more on the classification problem and finding efficient algorithms, while those consider more on the abstract hypothesis spaces[1].

This paper provides a new perspective for online multiclass prediction with bandit setting model. Our algorithm, called *Conservative OVA(one-versus-all) Reduction with Online Passive-Aggressive Algorithm*, enjoys a pleasing theoretical result of cumulative error rate as well as updating time cost linear to the sample size. First, we reduce the multiclass problem to binary according to *Margin Based Reduction from Multiclass to Binary*[7]; then a conservative scheme is applied for dealing with the bandit setting ; finally *Online Passive-Aggressive* algorithm[8] is embedded to handle binary problems. More details about our algorithm will be discussed later. Further experiments on several multiclass prediction dataset verify our theoretical bound and demonstrate that our algorithm performs far better than the *Banditron* Algorithm and the *Realizable Offset Tree Reduction* Algorithm.

II. Basic Definitions

In this section, we will define the problem of online multiclass prediction with bandit setting model in details, as well as the concept of *Online Binary Linear Predictor*, which will be used in the following sections.

A. Online Multiclass Prediction with Bandit Setting Model

In Online multiclass prediction with bandit setting problem, the learner observes instances in a consecutive manner. At each round t, the learner receives an instance feature vector $\mathbf{x_t} \in \mathbb{R}^d$, then it predicts a label $\hat{y_t}$ from a predefined label set, which includes k labels, denoted by $[k] = \{1, \ldots, k\}$. Once a predicted label is given, in traditional multiclass problem, the information that which label $y_t \in [k]$ the instance $\mathbf{x_t}$ actually corresponds to would be revealed to the learner; however, in bandit setting, the learner would only receive partial feedback $\mathbf{1}[\hat{y_t} = y_t]$, where $\mathbf{1}[\pi]$ is equal to 1 if the statement π is true and 0 otherwise. Under this circumstance, the learner knows the actual label of instance $\mathbf{x_t}$ only if it gets a positive feedback; if it receives a negative feedback, the learner would only know that the predicted label $\hat{y_t}$ is wrong, but it is not disclosed that what the true label y_t is.

As conventional online learning, the ultimate goal of the task is to minimize the cumulative number of prediction error E. Until round T, E can be shown as

$$E = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t \neq y_t] \tag{1}$$

To minimize E, we have to take every efforts of what we can to make a better classifier for later round prediction at each round.

Also we denote learning hypothesis of this problem at each round t to be h_t, where $h_t : \mathbb{R}^d \to [k]$ belongs to a class of hypotheses $\mathcal{H}$.

B. Online Binary Linear Predictor

For instance $(\mathbf{x_t}, y_t)$, where $\mathbf{x_t} \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$, an online binary linear predictor maintains a weight vector $\mathbf{w_t} \in \mathbb{R}^d$ and makes a prediction based on the sign of $(\mathbf{w_t} \cdot \mathbf{x_t})$. Then $\mathbf{w_t}$ is updated after the label feedback is received at each round. Not only the sign of $(\mathbf{w_t} \cdot \mathbf{x_t})$ indicates the predicted label, but also the magnitude of it represents some kind of confidence. We denote the term $y_t(\mathbf{w_t} \cdot \mathbf{x_t})$ as the (signed) margin acquired on round t as [8]. So aiming at realizing a large margin which convinces a higher confidence, some online binary linear predictor would attain a margin of at least 1. Therefore, this kind of algorithm incurs a loss whenever a margin is less than 1 occurs. A usual loss used is called *hinge-loss* function defined as following,

$$l_t\left(\mathbf{w_t}; (\mathbf{x_t}, y_t)\right) = \begin{cases} 0 & y_t(\mathbf{w} \cdot \mathbf{x}) \geq 1 \\ 1 - y_t(\mathbf{w_t} \cdot \mathbf{x_t}) & \text{otherwise} \end{cases} \tag{2}$$

This loss is equal to 0 when the (signed) margin is larger than 1; otherwise it is equal to the distance(difference) between 1 and the margin value. In *Online Binary Linear Prediction* problem, the small *cumulative loss* $\sum_{t=1}^{T} l_t$ or small *cumulative squared loss* $\sum_{t=1}^{T} l_t^2$ is desired. The cumulative loss or cumulative squared loss is an upper bound of the cumulative number of prediction mistakes E, because

whenever a mistake is made then $l_t \geq 1$ and $l_t^2 \geq 1$, as well as the obvious fact that $l_t \geq 0$, which is noted in Eq. (1).

III. The Conservative OVA Reduction with Online Passive-Aggressive Algorithm

A. Overview

In this section, we will introduce our *Conservative OVA(one-versus-all others) Reduction with Online Passive-Aggressive Algorithm* in details, which is novel and concentrated on the idea of "conservative".

In contrast to the *Banditron Algorithm*[1], our method doesn't take much efforts to deal with the balance between exploitation and exploration, since we abandon the Kesler's construction [9] for multiclass setting. Instead, we bring in the method of *Margin Based Reduction from Multiclass to Binary*[7] as an oracle framework for multiclass prediction problem with bandit setting. The simplicity of this method as well brings a lot of benefits for fully theoretical analysis.

What we mean by calling our algorithm "conservative" is that we choose to take advantage of maximum information each sample discloses fully, and not to explore greedily to obtain more information. More about "conservative" would be clearly explained in subsection III-C.

Furthermore, we embed *Online Passive-Aggressive Algorithms*[8] for online binary learners as base learning algorithm to accomplish our method for online multiclass prediction with bandit setting problem. Noting that other online binary learning algorithms can be suitable for our oracle *Conservative OVA Reduction* scheme, we adopt this choice to make a comparable part against *Banditron Algorithm*, as well as further theoretical analysis.

B. Margin based Reduction from Multiclass to Binary

Reducing multiclass prediction problem to binary attracts much attention during the years, which includes two of the most famous methods, *one-versus-all others* and *all-pairs* approaches[10]. Usually these kinds of reduction for multiclass prediction enjoy acceptable performance as good as some direct algorithms.

In this paper, we adopt the *Margin based Output Coding*[7] as the basic scheme for our problem. First, a *coding matrix*

$$\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$$

is given, where k is the number of classes and l is the number of binary classifiers. Any binary classifier learning algorithm is provided with labeled examples in the form of $(\mathbf{x_i}, M(y_i, s))$, where $s = 1, \ldots, l$. In definition, $M(y_i, s) = 1$ means that the sample $(\mathbf{x_i}, y_i)$ is a positive example for s_{th} binary classifier, while $M(y_i, s) = -1$ indicates that it is a negative example and $M(y_i, s) = 0$ means this sample is omitted by the s_{th} binary classifier.

For instance, for *all-pair* reduction, M is a $k \times \binom{k}{2}$ matrix with each column corresponding to a label pair (k_1, k_2). In this column, k_1 row is $+1$, k_2 row is -1 and others are equal to 0. Thus one classifier is maintained for examples from these two classes; for *one-versus-all others* approach, the matrix M is $k \times k$ with only $+1$ along diagonal, while -1 in other elements.

Then for each binary learning algorithm, we provide a single hypothesis $f_s(x)$ to deal with every samples $(x_i, M(y_i, s))$. Therefore, variants of binary learning algorithm are suitable for such reduction.

Then, how to combine every binary output to predict the final class? Several methods have been carefully studied. First, Let $\mathrm{M}(r)$ denote the r_{th} row of matrix M and $\mathrm{f}(x)$ represent the prediction vector of x:

$$\mathbf{f}(x) = (f_1(x), f_2(x), \ldots, f_l(x)).$$

The basic idea is to find out a label r whose corresponding $\mathrm{M}(r)$ is most similar to the output $\mathrm{f}(x)$.

The simplest method of combining binary classifiers is called *Hamming Decoding*[7]. It is to count the number of difference between the prediction $\mathrm{f}(x)$ and the r_{th} row $\mathrm{M}(r)$ of matrix M, which is so-called *hamming distance*:

$$d_H(\mathbf{M}(r), \mathbf{f}(x)) = \frac{1}{2} \sum_{s=1}^{l} (1 - \mathrm{sign}(M(r, s) f_s(x))) \quad (3)$$

where $\mathrm{sign}(z)$ equals $+1$ when $z > 0$, -1 when $z < 0$ and 0 when $z = 0$. For an input instance x, the predicted label $\hat{y} \in [k]$ is

$$\hat{y} = h(\mathbf{x}) = \arg\min_r d_H(\mathbf{M}(r), \mathbf{f}(x)) \quad (4)$$

However, the *Hamming Decoding* only exploits the "sign" information of the prediction f(x) and ignores the advantage of the prediction margin which can be shown as kind of "confidence". Another approach called *loss-based decoding*[7] is proposed, which aims to predict the label $\hat{y}$ by minimizing the total loss on an input instance $\mathbf{x}$,

$$d_L(\mathbf{M}(r), \mathbf{f}(x)) = \sum_{s=1}^{l} L(M(r, s) f_s(x)). \quad (5)$$

where $L(z) = (1 - z)_+ = \max\{0, 1 - z\}$ is a convex loss function. Other convex loss functions can be used, but here we only consider the above one.

Therefore, similar to *hamming decoding* the predicted label $\hat{y} \in [k]$ can be denoted as:

$$\hat{y} = h(\mathbf{x}) = \arg\min_r d_L(\mathbf{M}(r), \mathbf{f}(x)). \quad (6)$$

C. The Conservative OVA Reduction Scheme

In this subsection, we describe our *Conservative OVA Reduction Scheme* for multiclass prediction with bandit setting problem in details.

At first, we reduce the multiclass problem to binary by *one-versus-all others* reduction. As mentioned in III-B, now the *coding matrix* $\mathbf{M}$ is $k \times k$,

$$\mathbf{M} = \begin{pmatrix} +1 & -1 & \cdots & -1 \\ -1 & +1 & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \cdots & -1 & +1 \end{pmatrix} \tag{7}$$

Generally speaking, we maintain k binary classifiers to distinguish each class from all other classes. An example (x, y) is a positive example only for the y_{th} classifier, and a negative example for the other $k-1$ classifiers.

Then based on *one-versus-all others* reduction, we will talk about our *Conservative Scheme*. In bandit setting, if a learner receives a positive feedback of its prediction, then it immediately reveals what the actual label of this sample is. Thus an example with full label information is obtained and all the k binary classifier can be updated by the sample (x_t, y_t) at this step; Otherwise, if a negative feedback is accepted, the learner will only know that this example doesn't belong to a certain class and has no further information about which class it belongs to, denoted by $\hat{y}_t$. Thus only the binary classifier corresponding to y_t with x_t as a negative example is updated, and with other binary classifiers unchanged.

Finally, the learner outputs the predicted label based on its former knowledge by minimizing the total loss as Eq. (6), where $\mathbf{M}$ is shown as Eq. (7).

In this *Conservative* scheme, when an input instance $\mathbf{x_t}$ is misclassified, only the corresponding binary classifier is updated in order to put on more loss to assign such example as this misclassified class. So if a similar instance $\mathbf{x_{t'}}$ comes, then the learner is more likely to predict it as another class. Since other binary classifiers are not changed, the combined learner's preference of other labels will only root in the knowledge obtained from former steps. Thus we try not to make any prior knowledge to influence the learner, and let the learner exploit the knowledge by itself.

This updating scheme is *conservative*. We don't take any efforts to explore what the real label the example x_t is when the predicted result is incorrect. Instead, we try to make use of the partial information that which class it does not belong to as entirely as possible. Thanks to the *one-versus-all others* reduction, these partial feedbacks can be easily handled by only updating corresponding binary classifiers.

In addition, an online algorithm is called *conservative* if it updates its prediction rule only in rounds in which it makes a prediction error[11]. Nevertheless in our work, the definition of *conservative* is not quite the same. We call our algorithm *conservative* because that when the algorithm gets a negative feedback, it only updates the corresponding binary classifier, while leaving others unchanged.

D. Embedding Online Passive-Aggressive Algorithms

In subsection III-C, we introduced the *Conservative OVA Reduction Scheme* for multiclass prediction problem with bandit setting model. However, we do not provide any algorithm to handle the binary learning problem. In this subsection, we will focus on *Online Binary Passive-Aggressive* Algorithm[8] to accomplish our method.

Algorithm 1: The Conservative OVA Reduction with Online Passive-Aggressive Algorithm

Data: sequential data $(x_1, y_1), \ldots, (x_T, y_T)$
Aggressive Parameters: C
Initialization: $w_{s0} = \mathbf{0} \in \mathbb{R}^d$, $s = 1, \ldots, k$
for $t = 1, 2, \ldots, T$ **do**
  Receive $x_t \in \mathbb{R}^d$;
  Set $\hat{y}_t = \arg\min_r d_L(\mathbf{M}(r), \mathbf{f}(x))$;
  Predict $\hat{y}_t$ and reveal the feedback $\mathbf{1}(\hat{y}_t = y_t)$;
  if $\mathbf{1}(\hat{y}_t = y_t)$ **then**
    for $s = 1, \ldots, k$ **do**
      $w_{s(t+1)}$=PA$(C, w_{st}, (x_t, M(\hat{y}_t, s)))$
    end
  else
    $w_{\hat{y}_t(t+1)}$=PA$(C, w_{\hat{y}_t t}, (x_t, -1))$;
  end
end

Algorithm 2: Online Passive-Aggressive Algorithm(PA)

Input: $C, w_t, (x_t, y_t)$
Output: w_{t+1}
Suffer loss: $l_t = \max\{0, 1 - y_t(w_t x_t)\}$;
Update:
$w_{t+1} = w_t + \alpha_t y_t x_t$;
where
$\alpha_t = \frac{l_t}{||x_t||^2}$(Basic PA);
or
$\alpha_t = \min\left\{C, \frac{l_t}{||x_t||^2}\right\}$(PA-I);
or
$\alpha_t = \frac{l_t}{||x_t||^2 + \frac{1}{2C}}$(PA-II);

The binary classification learner maintains a weight vector $\mathbf{w_t} \in \mathbb{R}^d$, which is initialized to $\mathbf{w_0} = (0, \ldots, 0)$. When a sample $(x_t, \tilde{y}_t)$, where $\tilde{y}_t \in \{-1, +1\}$, is coming in round t, the basic *Passive-Aggressive* algorithm updates $\mathbf{w_t}$ according to

$$\begin{aligned} \mathbf{w}_{t+1} &= \arg\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2 \\ s.t. \quad & l(\mathbf{w}; (x_t, \tilde{y}_t)) = 0 \end{aligned} \tag{8}$$

which is called *PA* model. Whenever the *hinge-loss* l is 0, $\mathbf{w}_{t+1}$ remains as $\mathbf{w}_t$; And when the loss is positive,

Eq. (8) forces the updated weight vector $\mathbf{w_{t+1}}$ to satisfy $l(\mathbf{w}; (x_t, \tilde{y}_t)) = 0$.

Since problem (8) is a convex optimization problem, it can be easily solved. According to Karush-Khun-Tucker conditions[12], the closed form of solution for (8) is

$$\mathbf{w_{t+1}} = \mathbf{w_t} + \alpha_t \tilde{y}_t x_t, \text{ where } \alpha_t = \frac{l_t}{||x_t||^2} \tag{9}$$

It seems too brutal to force the weight vector $\mathbf{w_{t+1}}$ to satisfy the constraint imposed by the current example $(x_t, \tilde{y}_t)$ because of the common phenomenon of the noisy label information. Thus a nonnegative slack variable ξ is brought in to achieve a soft-margin classifier. When the objective function scales linearly with ξ, we call it *PA-I* model:

$$\begin{aligned} \mathbf{w}_{t+1} &= \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2 + C\xi \\ s.t. \quad & l(\mathbf{w}; (x_t, \tilde{y}_t)) \leq \xi \\ & \xi \geq 0 \end{aligned} \tag{10}$$

On the other hand, when the objective function scales quadratically with ξ, we call it *PA-II* model:

$$\begin{aligned} \mathbf{w}_{t+1} &= \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2 + C\xi^2 \\ s.t. \quad & l(\mathbf{w}; (x_t, \tilde{y}_t)) \leq \xi \\ & \xi \geq 0 \end{aligned} \tag{11}$$

where C is a parameter that adjusts how much the soft-margin ξ affects the objective function.

Eq. (10) and Eq. (11) are also convex optimization problems, the solutions to which also take the closed form $\mathbf{w_{t+1}} = \mathbf{w_t} + \alpha_t \tilde{y}_t x_t$, where

$$\alpha_t = \min\left\{C, \frac{l_t}{||x_t||^2}\right\} \text{ (PA-I)} \tag{12}$$

or

$$\alpha_t = \frac{l_t}{||x_t||^2 + \frac{1}{2C}} \text{ (PA-II)}$$

Embedding above online binary prediction algorithm into our *Conservative OVA Reduction* scheme, we can conclude our algorithm as *Algorithm* 1.

IV. Theoretical Analysis

In this section, we will carefully derive some relative bounds for our *Conservative OVA with PA algorithm* and verify the effectiveness of our algorithm.

First, as [7], we define the distance between two rows of *coding matrix* M, $\mathbf{u}, \mathbf{v} \in \{-1, 0, +1\}^l$, as

$$\begin{aligned} \Delta(u, v) &= \sum_{s=1}^{l} \begin{cases} 0 & \text{if } u_s = v_s \wedge u_s \neq 0 \wedge v_s \neq 0 \\ 1 & \text{if } u_s \neq v_s \wedge u_s \neq 0 \wedge v_s \neq 0 \\ \frac{1}{2} & \text{if } u_s = 0 \vee v_s = 0 \end{cases} \\ &= \sum_{s=1}^{l} \frac{1 - u_s v_s}{2} \\ &= \frac{l - \mathbf{u} \cdot \mathbf{v}}{2} \end{aligned}$$

Thus, the minimum distance ρ between all of the distinct rows can be denoted as:

$$\rho = \min\{\Delta(M(r_1), M(r_2)) : r_1 \neq r_2\} \tag{13}$$

Eq. (13) is important for our further analysis of mistake bounds. For instance, for *one-versus-all others* coding, on which will be mainly concentrated in this paper, $\rho = 2$; and for *all-pairs* coding, $\rho = (\binom{k}{2} - 1)/2 + 1$.

Second, by noting that the *hinge-loss* function Eq. (2) is convex of parameter $z = y_t(\mathbf{w_t} \cdot \mathbf{x_t})$, we denote $L(z) = (1 - z)_+ = \max\{0, 1 - z\}$ and easily obtain the following inequality:

$$\frac{L(z) + L(-z)}{2} \geq L(0) = 1, \quad \forall z \in \mathbb{R} \tag{14}$$

Then we bring up following theorem to bound the online cumulative mistakes E of all kinds of *coding matrix* M.

Theorem 1. *Assuming that for a sequence of samples* $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$*, where* $x_t \in \mathbb{R}^d$*,* $y_t \in [k]$*, at each round, we maintain a classifier* $\mathbf{f_t}$*, where* $\mathbf{f_t} = (f_{1t}, f_{2t}, \ldots, f_{lt})$*. And with a coding matrix* $\mathbf{M} \in \{-1, 0, +1\}^{k\times l}$ *and loss function* L *as defined above, let* ρ *be as Eq. (13), the cumulative number of mistakes made by the loss-based decoding scheme of an online algorithm satisfies:*

$$E = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t \neq y_t] \leq \frac{1}{\rho} \sum_{s=1}^{l} \sum_{t=1}^{T} L(M(y_i, s) f_{st}(x_i)) \tag{15}$$

Proof: In round t, if the incoming example (x_t, y_t) is incorrectly classified, then by the definition of loss decoding, there exists at least one label $r \neq y_t$ that satisfies:

$$d_L(M(r), f_t(x)) \leq d_L(M(y_t), f_t(x)) \tag{16}$$

Let $z_{st} = M(y, s) f_{st}(\mathbf{x_t})$ and $z_{st}{}' = M(r, s) f_{st}(\mathbf{x_t})$, then Eq. (16) can be rewritten as

$$\sum_{s=1}^{l} L(z_{st}{}') \leq \sum_{s=1}^{l} L(z_{st})$$

Let $S = \{s : M(r,s) \neq M(y_t,s), s = 1,\ldots,l\}$. An inequality can be attained:

$$\sum_{s=1}^{l} L(z_{st}) \geq \sum_{S} L(z_{st}) \geq \frac{1}{2}\sum_{S}(L(z_{st}^{'}) + L(z_{st})) \quad (17)$$

If $M(r,s) = -M(y_t,s)$ and $M(r,s)M(y_t,s) \neq 0$, $z_{st}^{'} = -z_{st}$, thus $L(z_{st}^{'}) + L(z_{st}) \geq 2L(0) = 2$; Otherwise at least one of z_{st} and $z_{st}^{'}$ equals 0, which indicates that $L(z_{st}^{'}) + L(z_{st}) \geq L(0) = 1$. So by the definition ρ and $\delta(u,v)$, we can derive that:

$$\sum_{s=1}^{l} L(M(y_t,s)f_{st}(x_t)) \geq \Delta(M(r), M(y_t)) \geq \rho \quad (18)$$

Therefore, by taking summation of Eq.(18) according to round t and considering the basic character of loss function $L(\cdot)$ that $L(\cdot) \geq 0$, we obtain the following bound of the number of cumulative mistakes E:

$$E = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_t \neq y_t] \leq \frac{1}{\rho}\sum_{t=1}^{T}\sum_{s=1}^{l} L(M(y_i,s)f_{st}(x_i))$$

Simply switching the two summations completes the proof. ■

Theorem 1 differs from the training error bound of [7] since we focus on deriving a bound for online algorithms, while [7] deals with its batch counterpart.

For the *one-versus-all others* reduction, the following *Corollary* holds:

Corollary 1. *If the condition is the same to* Theorem *1, for one-versus-all others reduction, the cumulative number of mistakes made by the loss-based decoding scheme of an online algorithm satisfies,*

$$E = \sum_{t=1}^{T} 1[\hat{y}_t \neq y_t] \leq \frac{1}{2}\sum_{s=1}^{l}\sum_{t=1}^{T} L(M(y_i,s)f_{st}(x_i)) \quad (19)$$

Proof: With $\rho = 2$ in Eq.(15) for *one-versus-all others* reduction, Eq.(19) is simply verified. ■

In [8], Crammer etc. have proven bounds for *PA* algorithm. And *PA-I* and *PA-II* updating method approximate the basic *PA* updating when aggressive parameter C goes to infinity. The following lemma for binary classifiers has been mentioned and carefully discussed in [8].

Lemma 1. *Assuming that for a sequence of samples, $(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T)$, we have $x_t \in \mathbb{R}^d$ and $y_t \in \{+1,-1\}$. Let $\mathbf{u}$ be an arbitrary fixed predictor and define*

$$l_t = l(w_t;(x_t,y_t)) \quad \textit{and} \quad l_t^* = l(\mathbf{u};(x_t,y_t)) \quad (20)$$

Then by setting α_t be as in Algorithm *2, we hold the bound for any $\mathbf{u} \in \mathbb{R}^d$,*

$$\sum_{t=1}^{T} \alpha_t(2l_t - \alpha_t||x_t||^2 - 2l_t^*) \leq ||\mathbf{u}||^2 \quad (21)$$

Lemma 2. *Assuming that for a sequence of samples, $(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T)$, we have $x_t \in \mathbb{R}^d$, $y_t \in \{+1,-1\}$ and $||x_t|| \leq R$ for all t. Then the cumulative squared loss of PA model on this sequence is bounded by,*

$$\sum_{t=1}^{T} l_t^2 \leq (R||u|| + 2\sqrt{\sum_{t=1}^{T}(l_t^*)^2})^2 \quad (22)$$

Lemma 3. *Assuming that for a sequence of samples, $(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T)$, we have $x_t \in \mathbb{R}^d$, $y_t \in \{+1,-1\}$ and $||x_t|| \leq R$ for all t. Then the cumulative squared loss of PA-II model on this sequence is bounded by,*

$$\sum_{t=1}^{T} l_t^2 \leq \left(R^2 + \frac{1}{2C}\right)\left(||\mathbf{u}||^2 + 2C\sum_{t=1}^{T}(l_t^*)^2\right) \quad (23)$$

Corollary 2. *Assuming that for a sequence of samples, $(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T)$, we have $x_t \in \mathbb{R}^d$, $y_t \in \{+1,-1\}$ and $||x_t|| \leq R$ for all t. Thus the derived bound for cumulative loss of PA model is,*

$$\sum_{t=1}^{T} l_t \leq \sqrt{T}R||\mathbf{u}|| + 2\sqrt{T\sum_{t=1}^{T}(l_t^*)^2} \quad (24)$$

Proof: The Corollary is simply verified by using Cauchy Schwarz inequality that $(\sum_{t=1}^{T} l_t)^2 \leq T\sum_{t=1}^{T} l_t^2$ to the left-hand side of Eq.(22) in Lemma 2. ■

The following *Corollary* 3 can be verified by using Cauchy Schwarz inequality, which is similar to the proof of *Corollary* 2.

Corollary 3. *Assuming that for a sequence of samples, $(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T)$, we have $x_t \in \mathbb{R}^d$, $y_t \in \{+1,-1\}$ and $||x_t|| \leq R$ for all t. Thus the derived bound for cumulative loss of PA-II is,*

$$\sum_{t=1}^{T} l_t \leq \sqrt{T\left(R^2 + \frac{1}{2C}\right)\left(||\mathbf{u}||^2 + 2C\sum_{t=1}^{T}(l_t^*)^2\right)} \quad (25)$$

Lemma 4. *Assuming that for a sequence of samples, $(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T)$, we have $x_t \in \mathbb{R}^d$, $y_t \in \{+1,-1\}$ and $||x_t|| \leq R$ for all t. Then the cumulative loss of PA-I on this sequence is bounded by,*

$$\sum_{t=1}^{T} l_t \leq \max\left(\frac{1}{C}\|u\|^2 + 2\sum_{t=1}^{T} l_t^*, \sqrt{2R^2\|\mathbf{u}\|^2 + 4CR^2\sum_{t=1}^{T} l_t^*}\right) \quad (26)$$

Proof: Because $\alpha_t = \min\left(C, \frac{l_t}{\|x_t\|^2}\right)$, we can know that $\alpha_t l_t^* \leq C l_t^*$ and $\alpha_t\|x_t\|^2 \leq l_t$, thus by using Lemma 1 directly, we obtain

$$\sum_{t=1}^{T} \alpha_t l_t \leq \|\mathbf{u}\|^2 + 2C\sum_{t=1}^{T} l_t^*$$

Then with definition of α_t of PA-I, Eq.(26) is derived. ■

Finally, plugging *Corollary* 2, 3 and *Lemma* 4 into *Theorem* 1, we provide a bound on the cumulative number of mistakes, E, which the *Conservative OVA with PA* algorithm makes.

Theorem 2. *Assuming that for a sequence of samples,* $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$*, where* $x_t \in \mathbb{R}^d$*,* $y_t \in [k]$ *and* $||\mathbf{x_t}|| \leq R$*, in each round, we maintain a classifier* $\mathbf{f_t}$*, where* $\mathbf{f_t} = (f_{1t}, f_{2t}, \ldots, f_{lt})$*. And with a coding matrix* $\mathbf{M} \in \{-1, 0, +1\}^{k \times l}$ *as in Eq.(7) and loss function L, let* ρ *be as (13), the cumulative number of mistakes made by the* Conservative OVA(one-versus-all others) Reduction with Online Passive-Aggressive Algorithm *satisfies: for basic PA model,*

$$E \leq \frac{1}{2} \sum_{s=1}^{l} \left(\sqrt{T} ||\mathbf{u_s}|| + 2\sqrt{T \sum_{t=1}^{T} (l_{st}^*)^2} \right) \tag{27}$$

for PA-I model,

$$E \leq \frac{1}{2} \sum_{s=1}^{l} \max \left(\frac{1}{C} \|\mathbf{u_s}\|^2 + 2 \sum_{t=1}^{T} l_{st}^* , \sqrt{2R^2 \|\mathbf{u_s}\|^2 + 4CR^2 \sum_{t=1}^{T} l_{st}^*} \right) \tag{28}$$

for PA-II model,

$$E \leq \frac{1}{2} \sum_{s=1}^{l} \sqrt{T \left(R^2 + \frac{1}{2C} \right)} \cdot \sqrt{\left(||\mathbf{u_s}||^2 + 2C \sum_{t=1}^{T} (l_{st}^*)^2 \right)} \tag{29}$$

In linearly separable cases, by the definition of $\mathbf{u_s}$ and l_{st}^*, we find that there exist $\mathbf{u_s}$, $s = 1, \ldots l$ which makes the corresponding l_{st}^* equals 0. Thus the cumulative mistakes made by the *Conservative OVA with PA* Algorithm would only be a function of $\mathbf{u_s}$. Therefore, the cumulative error rate would approach to 0 as the number of samples T goes to infinity. Here we see a very pleasing theoretical results of our algorithm.

V. Time Complexity

In our method, we deal with samples in a consecutive manner. For each sample, k binary classifiers are maintained, so our method would update at most k binary classifiers. With regard to each binary classifier, the updating procedure takes a closed form and takes time linear to the feature dimension d, no matter which *Online Passive-Aggressive Algorithm* is employed. Thus our method takes at most $O(kd)$ time for each sample, which leads to a time complexity linear to sample size n.

VI. Experiments

A. Overview

In this section, we will validate the performance of *Conservative OVA with PA* algorithm on several real world datasets and report the experimental results. First, we describe the datasets we used; then we compare our method with its counterparts, namely *Banditron* and *Realizable Offset Tree Reduction* algorithm. We implemented *Multiclass Perceptron*, an online algorithm for learning with full label feedback information, as well for comparison. All the experiments are performed with MATLAB R2008a on a Intel(R) Core(TM)2 Duo CPU E8400@3.00GHZ running Windows XP with 3.25GB main memory.

For *Conservative OVA with PA* algorithm, all three PA updating methods are implemented as base learner. In both PA-I and PA-II models, the aggressive parameters C is tuned by grid search. Actually, we found that $C = 1$ is a quite good parameter value, leading to a good classification performance in all datasets we used. So we arbitrarily set the aggressive parameters C equal 1. Although the randomness of our method is negligible, only including random guess at the beginning, we ran the algorithm 10 times also and report the average cumulative error rate.

For *Banditron*, as in [1], we ran the algorithm for different values of the exploration parameter γ to determine the best value for each dataset. Since this algorithm contains a stochastic scheme, the cumulative error rate we report is averaged over 10 independent runs.

For *Realizable Offset Tree Reduction* algorithm, instead of taking binary perceptron as a base classification algorithm[2], we embed the the *Online Passive-Aaggressive* algorithm to construct a more comparable counterpart. As *Conservative OVA with PA* algorithm, we found out that aggressive parameters C as 1 is good based on grid search. Also we ran the algorithm 10 times to get the average cumulative error rate.

For *Multiclass Perceptron*, we adopt a simple adaptation of Perceptron algorithm[13] for multiclass prediction in the full label feedback case, called Kesler's construction in [9], [11]. Since there are random guesses in first few steps, we also ran the algorithm 10 times to get the average cumulative error rate.

B. Datasets

We use four real world datasets in our experiments, covering a wide range of properties: **RCV1-v2**[14], **MNIST**[1], **20 Newsgroups**[2] and **USPS**[3].

RCV1-v2[14]: This dataset is an archive of newswire stories for research purposes by Reuters, Ltd. Documents in this data set can contain more than one label. Practicably

[1] http://yann.lecun.com/exdb/mnist/
[2] http://people.csail.mit.edu/jrennie/20Newsgroups/
[3] http://cervisia.org/machine_learning_data.php

we choose data samples with only one of the highest four topic codes (CCAT, ECAT, GCAT and MCAT) in the "Topic Codes" hierarchy in the data set. This process constructs data size of 704877 from the original 804414 samples, vectors of which are cosine-normalized, log TF-IDF vectors.

MNIST: This is a database of handwritten digits, which had been size-normalized and centered in a 28×28 image. The formal dataset maintains 60000 examples for training and 10000 for testing. For our usage, we simply combine these two into onedataset consisting of total 70000 examples, vectors of which are the gray value of pixels and scaled to $[0, 1]$.

20 Newsgroups: This dataset collects nearly 20000 newsgroup documents categorized by 20 different newsgroups. We use the whole dataset including 19928 examples, vectors of which are scaled to binary encoding.

USPS: This is an optical character recognition dataset with 9298 samples. Samples are digits of 10 different classes, vectors of which are of 256 dimensions.

C. Results and Discussion

Figure 1, 2, 3 and 4 show the experimental results on four real world datasets mentioned above. The parameter γ of *Banditron* for **RCV1-v2** is set as 0.03, for **MNIST** as 0.15, **20 Newsgroups** as 0.3, **USPS** as 0.3. For each figure, the left subgraph represents the cumulative error rate of four algorithms, *Multiclass Perceptron*, *Banditron*, *Conservative OVA with PA-I* and *Realizable Offset Tree with PA-I*, on consecutive samples of each data set; the middle one describes the cumulative error rate of *Conservative OVA Reduction* scheme with three kinds of *PA* algorithms as base classifiers; the right subgraph depicts the cumulative error rate of *Realizable Offset Tree* reduction scheme whose base learners are achieved by three kinds of *PA* algorithms.

Table I shows the final cumulative error rate of each algorithm on each dataset.

From the experimental results on four datasets, we can see that

1) For online multiclass prediction with bandit setting problem, Our *Conservative OVA Reduction* scheme with PA algorithm as base learning algorithm generally outperforms other algorithms when evaluating the cumulative error rate. Even prediction results of our methods quite comparable to those of *Multiclass Perceptron*, which is a learning algorithm for traditional problem with full label information as feedback.
2) Our algorithm performs far better in contrast to *Banditron* algorithm, with an improvement on cumulative error rate at least 7.96% on **RCV1-v2** dataset, while at most 47.71% on **20 Newsgroup** dataset.
3) Though on **RCV1-v2** dataset, the result of *Realizable Offset Tree Reduction* with *PA* as base learner is rather comparable to our method, its low performances on other dataset show the unstable characteristic of it. Thus it is not an executable algorithm for real world application.
4) From the results of either our *Conservative OVA Reduction* or *Realizable Offset Tree Reduction* scheme, we find out that the difference between the three models of *PA* algorithm is slight and usually *PA-I* would achieve an acceptable results. Thus in real world application, we can only embed the *PA-I* as a base learning algorithm.

VII. Conclusions and Future work

In this paper, we propose a novel algorithm for online multiclass prediction with bandit setting problem. The practicability of our algorithm is verified theoretically by showing a pleasing bound. Moreover, experimental evaluation on four real world datasets suggests that our *Conservative OVA with PA* algorithm performs better than other algorithms on the same problem.

For future work, we would like to think about online multiclass multilabel problem with bandit setting problem, where each sample has more than one label. It seems more practical that samples have multilabel in real world usages, especially in recommender systems. As well a cost sensitive online multiclass problem in such setting would be of great interest to us. Also this kind of paradigm seems related to Semi-supervised Learning(SSL)[15], [16], [17] if we consider the instance with positive feedback is the labeled instance and the one with negative feedback is the unlabeled instance for classes that it does not belong to. Another approach is to consider the interclass hypothesis sharing as in [18] to build a more accurate online classifier.

Acknowledgment

This work is supported by NSFC (Grant No. 60675009 and 60835002). The authors would like to thank anonymous reviewers for their valuable comments.

References

[1] S. Kakade, S. Shalev-Shwartz, and A. Tewari, "Efficient bandit algorithms for online multiclass prediction," in *Proceedings of the 25th international conference on Machine learning*. ACM New York, NY, USA, 2008, pp. 440–447.

[2] A. Beygelzimer, J. Langford, and T. Zhang, "The Offset Tree for Learning with Partial Labels," *Arxiv preprint arXiv:0812.4044*, 2008.

[3] V. Vapnik, *The nature of statistical learning theory*. springer, 2000.

[4] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proceedings of the Third IEEE International Conference on Data Mining*, vol. 2003. Citeseer, 2003, pp. 435–442.

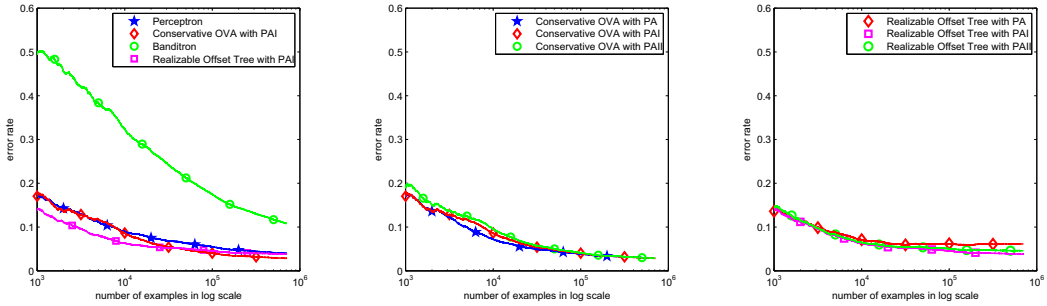

Figure 1. Cumulative Error Rate Comparison on RCV1-v2 dataset

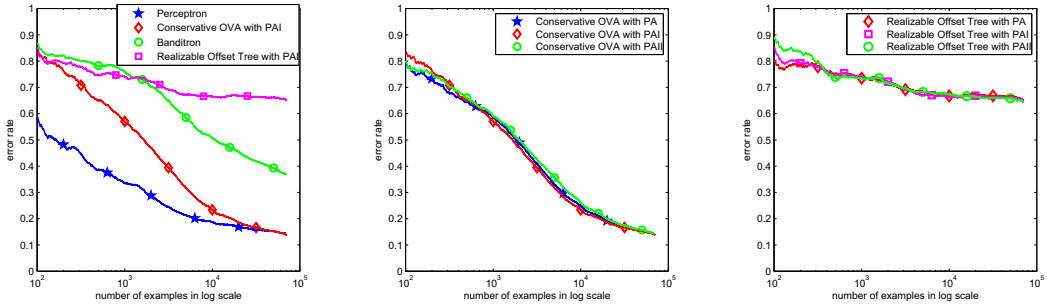

Figure 2. Cumulative Error Rate Comparison on Mnist dataset

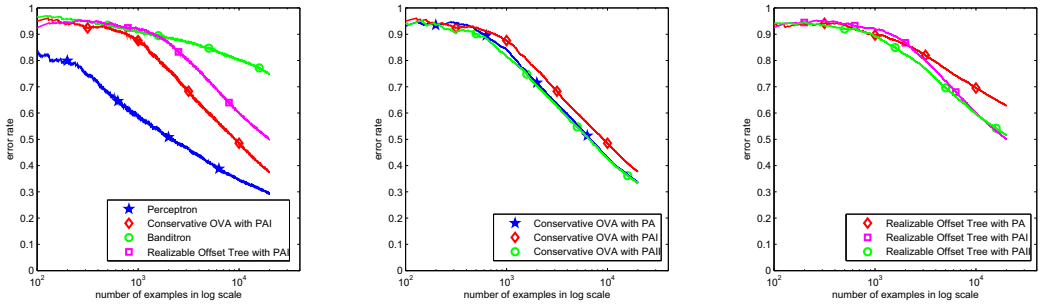

Figure 3. Cumulative Error Rate Comparison on 20 Newsgroups dataset

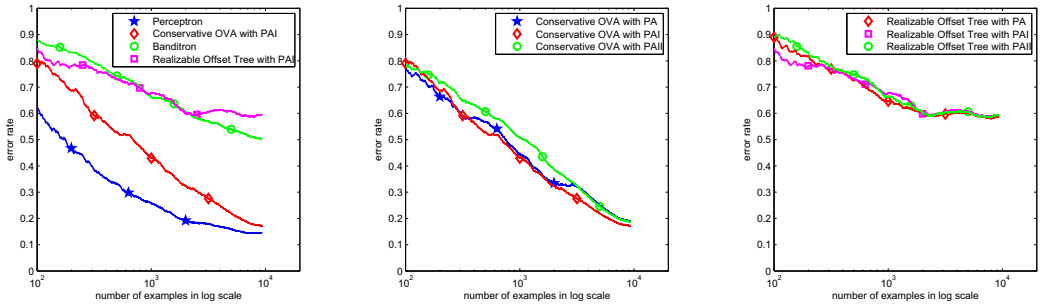

Figure 4. Cumulative Error Rate Comparison on USPS dataset

Table I
AVERAGE ERROR RATE(%) COMPARISON ON FOUR REAL WORLD DATASET. (*Multiclass Perceptron* DEALS WITH FULL LABEL FEEDBACK PROBLEM, WHILE OTHER METHODS DEAL WITH PARTIAL FEEDBACK PROBLEM.)

Algorithm	RCV1-v2	Mnist	20 Newsgroups	USPS
Banditron	10.80	36.79	75.07	50.28
Conservative OVA with PA	2.91	14.01	33.75	18.85
Conservative OVA with PAI	**2.84**	**13.92**	37.70	**17.70**
Conservative OVA with PAII	2.87	14.23	**33.36**	18.60
Realizable Offset Tree with PA	6.13	65.04	62.85	58.79
Realizable Offset Tree with PAI	3.79	64.97	50.04	59.50
Realizable Offset Tree with PAII	4.58	64.33	51.58	59.51
Multiclass Perceptron	3.97	14.19	29.50	14.47

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE*, vol. 36. IEEE Computer Society Press, 1995, pp. 322–331.

[6] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," *Advances in Neural Information Processing Systems*, 2007.

[7] E. Allwein, R. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *The Journal of Machine Learning Research*, vol. 1, pp. 113–141, 2001.

[8] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[9] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Wiley New York, 2001.

[10] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Annals of Statistics*, pp. 451–471, 1998.

[11] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *The Journal of Machine Learning Research*, vol. 3, pp. 951–991, 2003.

[12] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[13] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, pp. 386–408, 1958.

[14] D. Lewis, Y. Yang, T. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *The Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[15] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, 2006.

[16] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. The MIT Press, 2004, p. 321.

[17] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2008.

[18] M. Fink, S. Shalev-Shwartz, Y. Singer, and S. Ullman, "Online multiclass learning by interclass hypothesis sharing," in *Proceedings of the 23rd international conference on Machine learning*. ACM New York, NY, USA, 2006, pp. 313–320.