

# Efficient Active Learning with Boosting

Zheng Wang \*

Yangqiu Song \*

Changshui Zhang \*

## Abstract

This paper presents an active learning strategy for boosting. In this strategy, we construct a novel objective function to unify semi-supervised learning and active learning boosting. Minimization of this objective is achieved through alternating optimization with respect to the classifier ensemble and the queried data set iteratively. Previous semi-supervised learning or active learning methods based on boosting can be viewed as special cases under this framework. More important, we derive an efficient active learning algorithm under this framework, based on a novel query mechanism called query by incremental committee. It does not only save considerable computational cost, but also outperforms conventional active learning methods based on boosting. We report the experimental results on both boosting benchmarks and real-world database, which show the efficiency of our algorithm and verify our theoretical analysis.

## 1 Introduction

In classification problems, a sufficient number of labeled data are required to learn a good classifier. In many circumstances, unlabeled data are easy to obtain, while labeling is usually an expensive manual process done by domain experts. Active learning can be used in these situations to save the labeling effort. Some works have already been done for this purpose [21, 5, 23, 8]. Many methods have been used for querying the most valuable sample to label. Recently, the explosive growth in data warehouse and internet usage has made large amount of unsorted information potentially available for data mining problems. As a result, fast and well performed active learning methods are much desirable.

Boosting is a powerful technique widely used in machine learning and data mining fields [14]. In boosting community, some methods have been proposed for active learning. Query by Boosting (QBB) [1] is a typical one. Based on the *Query By Committee* mechanism [21], QBB uses classifier ensemble of boosting as the query committee, which is deterministic and easy to

handle. For each query, a boosting classifier ensemble is established. Then the most uncertain sample, which has the minimum margin for current classifier ensemble, is queried and labeled for training the next classifier ensemble. [15] generalizes QBB for multiclass classification problems. Besides these, there are also other well established practical boosting based active learning algorithms for different applications, including combining active learning and semi-supervised learning under boosting (COMB) for spoken language understanding [12] and adaptive resampling approach for image identification [17].

However, there still remains some problems for this type of methods.

- There lacks more theoretical analysis for these boosting based active learning methods. There is no explicit consistent objective function, which unifies both the base learner and the query criterion.

- Their computational complexity is high. Since for each query, sufficient iterations should be made until boosting converges. This is a critical problem limiting the practical use of this type of methods.

- Their initial query results are not very satisfying. Sometimes, they are even worse than random query. It is a common problem for most of the active learning methods [3]. This is because they may get very bad classifiers based on only a few labeled samples at the beginning. The bad initial queries, based on these classifiers, will make the whole active learning process inefficient.

- The number of classifiers in the committee is fixed among all above methods. It is hard to determine a suitable size of the committee in practice. This will limit the query efficiency and obstruct the algorithm from getting the optimal result.

To solve above problems and make this type of methods more consistent and more practical. In this paper, we propose a unified framework of Active Semi-Supervised Learning (ASSL) boosting, based on the theoretical explanation of boosting as a gradient decent process [14, 18]. We construct a variational objective function for both semi-supervised and active learning boosting, and solve it using alternating optimization. Theoretical analysis is given to show the convergence condition and the query criterion.

\*State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, P. R. China, {wangzheng04@gmail.com, songyq99@mails.thu.edu.cn, zcs@mail.thu.edu.cn}

What is more important is that, to solve the latter three problems, a novel algorithm with incremental committee members is developed under this framework. It can approximate the full data set AdaBoost good enough after sufficient iterations. Moreover, it runs much faster and performs better than conventional boosting based active learning methods.

The rest of this paper is organized as follows. In section 2, the unified framework Active Semi-Supervised Learning Boost (ASSL-Boost) is presented and analyzed. The novel efficient algorithm is proposed in section 3. Experimental results for both boosting benchmarks and real world applications are shown in section 4. Finally we give some discussions and conclude in section 5.

## 2 A Unified View of ASSL Boosting

**2.1 Notations and Definitions** Without loss of generality, we assume there are  $l$  labeled data,  $D_L = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\}$ , and  $u$  unlabeled data,  $D_U = \{(\mathbf{x}_{l+1}), \dots, (\mathbf{x}_{l+u})\}$ , in data set  $D$ ; typically  $u \gg l$ .  $\mathbf{x}_i \in R^d$  is the input point and the corresponding label is  $y_i \in \{-1, +1\}$ . We focus on binary classification problems.

In our work, we treat the boosting-type algorithm as an iterative optimization procedure for a cost functional of classifier ensembles, which is also regarded as a function of margins [18]:

$$(2.1) \quad C(F) = \sum_{(\mathbf{x}_i, y_i) \in D} c_i(F) = \sum_{(\mathbf{x}_i, y_i) \in D} m_i(\rho).$$

$C : \mathcal{H} \rightarrow [0, +\infty)$  is a functional on the classifier space  $\mathcal{H}$ .  $c_i(F) = c(F(\mathbf{x}_i))$  is the functional cost of the  $i$ th sample, and  $F(\mathbf{x}) = \sum_{t=1}^T \omega_t f_t(\mathbf{x})$ , where  $f_t(\mathbf{x}) : R^d \rightarrow \{1, -1\}$  are base classifiers in  $\mathcal{H}$ ,  $\omega_t \in R^+$  are the weighting coefficients of  $f_t$ , and  $t$  is the iteration time when the boosting algorithm is running.  $\rho = yF(\mathbf{x})$  is the margin.  $m_i$  is the margin cost of the  $i$ th sample.

To introduce the unlabeled information for semi-supervised data set, we consider to add the effect of unlabeled data into the cost, using pseudo margin  $\rho^U = y^F F(\mathbf{x})$  with pseudo label  $y^F = \text{sign}(F(\mathbf{x}))$  as in Semi-Supervised MarginBoost (SS-MarginBoost) [6]. Note that other types of pseudo label are also feasible here. In this case, unlabeled data get pseudo labels based on the classifier  $F$ , then elements in  $D_U$  become  $(\mathbf{x}_i, y_i^F)$ . The corresponding cost of  $D_U$  is

$$(2.2) \quad \sum_{\mathbf{x}_i \in D_U} m(-y_i^F F(\mathbf{x}_i)).$$

For active learning, we only focus on the myopic mode. In this case, only one sample is moved from  $D_U$

to  $D_L$  after each query. After  $n$  queries, these two sets become  $D_{U \setminus n}$ , which has  $u - n$  unlabeled data, and  $D_{L \cup n}$ , which has  $l + n$  labeled data. The queried samples compose the set  $D_n$ . The whole data set now is denoted by  $S_n = \{D_{L \cup n}, D_{U \setminus n}\}$ . We call it **semi-supervised data set**. Initially  $S_0 = D$ . After all unlabeled data are labeled, the data set is called **genuine data set**  $G$ ,  $G = S_u = D_{L \cup u}$ .

We define the cost functional on semi-supervised data set after  $n$  queries, for combined classifier  $F$  as  $C_{S_n}(F)$ :

$$(2.3) \quad C_{S_n}(F) = \sum_{(\mathbf{x}_i, y_i) \in D_{L \cup n}} m(-y_i F(\mathbf{x}_i)) + \alpha \sum_{\mathbf{x}_i \in D_{U \setminus n}} m(-y_i^F F(\mathbf{x}_i)) = \frac{1}{l+u} \left( \sum_{(\mathbf{x}_i, y_i) \in D_{L \cup n}} e^{-y_i F(\mathbf{x}_i)} + \alpha \sum_{\mathbf{x}_i \in D_{U \setminus n}} e^{-|F(\mathbf{x}_i)|} \right),$$

where  $\alpha$ ,  $0 \leq \alpha \leq 1$ , is a trade-off coefficient between the effect of labeled and unlabeled information. It can make our method more flexible. And the cost based on the genuine data set is  $C_G(F)$ :

$$(2.4) \quad C_G(F) = \frac{1}{l+u} \sum_{(\mathbf{x}_i, y_i) \in G} e^{-y_i F(\mathbf{x}_i)}.$$

It is the classical cost of boosting. For convenience of following analysis, the negative exponential margin expression is chosen for above cost. And we denote the corresponding optimal classifiers as  $F_{S_n}$  and  $F_G$  respectively, which can minimize the cost of semi-supervised data set boosting and genuine data set AdaBoost [6, 18, 9].

**2.2 The Framework of ASSL-Boost** With initial scarce labeled data, it is incapable to minimize  $C_G(F)$  directly to get the optimal  $F_G$ . Therefore, we aim at finding the best possible semi-supervised data set to approximate the genuine one, with the difference of their cost as the measurement. Then the optimal classifier  $F_{S_n}$  on this semi-supervised data set is the current best approximation we can get for  $F_G$ .

Now, we establish our algorithm framework, ASSL-Boost. In this framework, only one objective is optimized for both the learning and the querying process. It is to find the best classifier  $F$  and the most valuable queried data set  $D_n$  to minimize the distance between the cost  $C_{S_n}(F)$  on semi-supervised data set and the optimal cost  $C_G(F_G)$  on the genuine data set:

$$(2.5) \quad \min_{F, D_n} \text{Dist}(C_{S_n}(F), C_G(F_G)),$$

where the distance between two costs is defined as:

$$(2.6) \quad \text{Dist}(C_1(F_1), C_2(F_2)) = |C_1(F_1) - C_2(F_2)|.$$

Here,  $C_1(F_1)$  and  $C_2(F_2)$  are two cost functionals with classifiers  $F_1$  and  $F_2$ . The distance is within the range  $[0, +\infty)$ .

It is not easy to directly optimize (2.5) w.r.t.  $F$  and  $D_n$ , which affects  $C_{S_n}(\cdot)$ , simultaneously. Thus, we get an upper bound to separate this two variables:

$$\begin{aligned} & \text{Dist}(C_{S_n}(F), C_G(F_G)) \leq \\ & \text{Dist}(C_{S_n}(F), C_{S_n}(F_{S_n})) + \text{Dist}(C_{S_n}(F_{S_n}), C_G(F_G)). \end{aligned}$$

Minimizing (2.5) can be achieved by alternately minimizing the two terms of its upper bound w.r.t  $F$  and  $D_n$  individually. As a result, we solve this problem using alternating optimization in two steps.

**Step 1.** Fix the semi-supervised data set, and find current optimal classifier. This is,

$$(2.7) \quad \min_F \text{Dist}(C_{S_n}(F), C_{S_n}(F_{S_n})),$$

which tends to zero when we approximately get the optimal classifier  $F_{S_n}$ . We adopt Newton-Raphson method to find the optimal solution  $F_{S_n}$  of cost functional  $C_{S_n}(F)$ , as in [11]:

$$(2.8) \quad F \leftarrow F + \frac{\partial C_{S_n}(F)/\partial F}{\partial C_{S_n}(F)^2/\partial^2 F}.$$

This also can be viewed as the Gentle Adaboost for semi-supervised data set with pseudo labels under our cost.

**Step 2.** Fix the suboptimal classifier  $F_{S_n}$ , and query the most valuable unlabeled sample, which will change the cost of the current semi-supervised data set most towards the cost of the genuine data set. This is,

$$(2.9) \quad \min_{D_n} \text{Dist}(C_{S_n}(F_{S_n}), C_G(F_G)).$$

This procedure is moving the most valuable term from the unlabeled part to the labeled part in (2.2). With constant  $C_G(F_G)$ , which is the upper bound of  $\{C_{S_n}(F_{S_n})\}$  given by *Corollary 1* in the next subsection, minimizing (2.9) is equivalent to finding the data point  $(\mathbf{x}_q, y_q)$  that maximizes:

$$(2.10) \quad \max_{\mathbf{x}_q \in D_{U \setminus n}} (e^{-y_q F_{S_n}(\mathbf{x}_q)} - e^{-|F_{S_n}(\mathbf{x}_q)|}).$$

The coefficient  $\alpha$  for the second term does not affect the choice of optimal  $\mathbf{x}_q$  and can be ignored. The most useful sample we need is the one causing maximum cost increase.

In  $D_{U \setminus n}$ , the sample causing the biggest change is the one with the maximum margin among the samples that learn a wrong label by current classifier  $F_{S_n}$ . When we are not sure which one is mislabeled by  $F_{S_n}$ , finding the most uncertain one is a reasonable choice. So

unlabeled data with the minimum margin was queried as in [1]. This criterion usually decreases (2.9) more rapidly than random query, though it may not be the optimal one. We still use the same query criterion in this paper, as our main focus is the efficient query structure of the incremental committee, which will be introduced in section 3. The analysis of other criteria is left for future study.

The above two steps iterate alternately and get the best approximation for the optimal classifier that can be learnt by genuine data set boosting. Under this framework, QBB is a special case, with the unlabeled data having initial zero weights,  $\alpha = 0$ . And SS-MarginBoost is the first step to optimize our objective w.r.t. only one variable  $F$ . COMB is still under this framework, which uses classification confidence instead of margin.

**2.3 Analysis of the Framework** In this subsection, we analyze the characteristics of the cost functional during the active learning process. These properties guarantee that our objective is feasible and the framework can find the optimal solution of genuine data set cost based on this objective.

Following *Theorem 1* shows that the cost functionals,  $C_{S_n}(F)$  ( $n = 1, 2, \dots$ ), compose a monotonically non-decreasing series tending to genuine data set cost,  $C_G(F)$ , when we continuously query and label data.  $C_G(F)$  is the upper bound. *Corollary 1* shows the same characteristic for the optimal cost series. It was used to get the query criterion in the last subsection, which guarantees that our objective tends to zero. It also gives the reason why we use the convex negative exponential margin cost. *Corollary 2* shows the convergence property of the derivatives for the cost series. It will be used in the next section. *Theorem 2* shows that our optimization procedure can get the optimal classifier if the objective tends to zero. All proofs can be found in appendix.

**Theorem 1.** *The cost  $C_{S_n}(F)$  after  $n$  queries composes a monotonically non-decreasing series of  $n$  converging to  $C_G(F)$ , for any classifier ensemble  $F$ . We have  $C_{S_1}(F) \leq C_{S_2}(F) \leq \dots \leq C_{S_n}(F) \leq \dots \leq C_{S_u}(F) = C_G(F)$ .*

**Corollary 1.** *If the cost function is convex for margin, the minimum value of  $C_{S_n}(F)$ ,  $C_{S_n}(F_{S_n})$ , composes a monotonically non-decreasing series of  $n$  converging to  $C_G(F_G)$ , which is the minimum cost for genuine data set boosting. That is  $C_{S_1}(F_{S_1}) \leq C_{S_2}(F_{S_2}) \leq \dots \leq C_{S_n}(F_{S_n}) \leq \dots \leq C_{S_u}(F_{S_u}) = C_G(F_G)$ .*

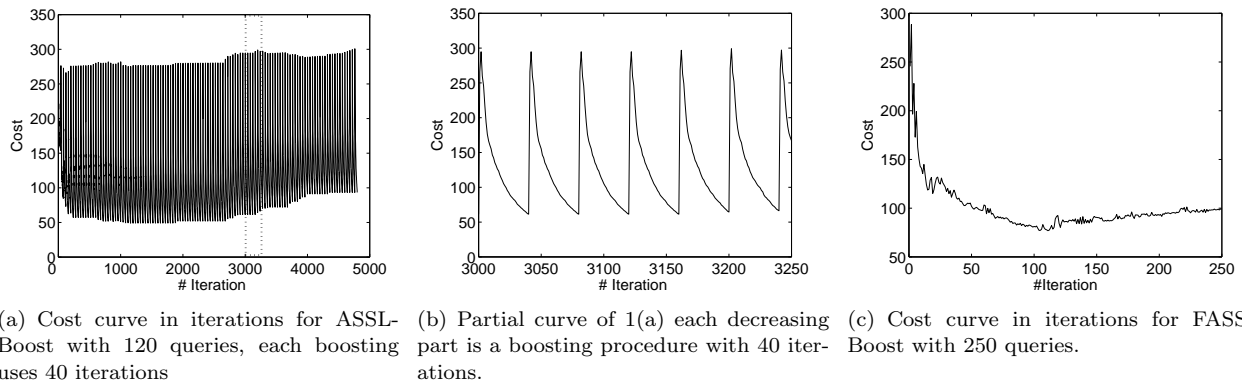


Figure 1: Learning curves of cost in iterations for previous ASSL-Boost methods and FASSL-Boost, FASSL-Boost in (c) describes a similar path and achieve the same final cost as the lower envelope of (a), which is about 100 in this example. In (a) the algorithm runs approximate 5000 iterations, while in (c) FASSL-Boost uses only 100 for the same result.

**Corollary 2.** *If the  $k$ th partial derivative of any cost functional exists and is finite, a series of  $\{\frac{\partial C_{S_n}(F)^k}{\partial^k F}\}$  can be composed, which converges to the  $k$ th partial derivative of cost functional  $C_G(F)$ ,  $\frac{\partial C_G(F)^k}{\partial^k F}$ .*

**Theorem 2.** *If the minimum of  $C_{S_n}(F)$ ,  $C_{S_n}(F_{S_n})$ , is equal to the final genuine data set minimum cost  $C_G(F_G)$  for certain  $F_{S_n}$ , this function  $F_{S_n}$  is also an optimal classifier for genuine data set boosting.*

### 3 The Efficient Algorithm of ASSL Boosting

**3.1 Query By Incremental Committee** The previous boosting based active learning methods [1, 12] under the ASSL-Boost framework have expensive computational cost. For each query, at least tens of iterations should be handled. However, only the last boosting classifier ensemble is used for final classification. This is a waste of previous classifier ensembles. The cost curve in iterations is shown in Fig. 1 (a) and (b). The convergence of the cost is only represented by the lower envelope in Fig. 1 (a), which is composed by the optimal cost series of semi-supervised data set in *Corollary 1*, while the other search iterations seem redundant. On the contrary, the big and complex classifier ensemble at the beginning of the query process may lead to poor query result, with such few labeled samples. As stated in following theorems, the complex ensemble seriously over-fits to the initial semi-supervised data set, which maybe far from the genuine one. As a result, the sample queried by this committee maybe far from the real

desired one. And the whole active learning process cannot be efficient. The above problems limit the usefulness of conventional ASSL-Boost type methods.

*Theorem 3* is the general risk bound in statistical learning theory, which shows the bound of the generalization risk dominated by empirical risk and expresses the over-fitting issue for typical learning problems.

**Theorem 3.**[4] *In inference problems, for any  $\delta > 0$ , with probability at least  $1 - \delta$ ,  $\forall F \in \mathcal{H}$ ,*

$$(3.11) \quad R_T(F) \leq R_l(F) + \sqrt{\frac{h(\mathcal{H}, l) - \ln(\delta)}{l}}.$$

$R_T(F)$  is the true risk and  $R_l(F)$  is the empirical risk for  $F$ , based on data distribution  $q(x)$ .

$$(3.12) \quad R_T(F) = \int r(F(\mathbf{x}), y)q(\mathbf{x})d\mathbf{x}.$$

and

$$(3.13) \quad R_l(F) = \sum_l r(F(\mathbf{x}), y).$$

$r(F(\mathbf{x}), y)$  is the risk function for sample  $(\mathbf{x}, y)$ .  $l$  is the number of the labeled samples, which are iid sampled with the distribution density  $q(\mathbf{x})$ .  $h$  is the capacity function, which describes the complexity of the hypothesis space  $\mathcal{H}$  for the learning problem. It can be VC-entropy, growth function, or VC-dimension.

To make *Theorem 3* held, it has a constrain that the available data are iid sampled from the true distribution,  $q(\mathbf{x})$ . In active learning, the queried samples are selected from a distribution with density  $p(\mathbf{x})$ , which is

often different from the original  $q(\mathbf{x})$ . The distribution density  $p(\mathbf{x})$  becomes higher in the queried area, where the sample has higher expected risk, and lower in other area. This is a covariance shift problem, which is a common scenery in active learning [22].

Thus, *Theorem 3* cannot be applied directly to ASSL-Boost. Luckily, the conclusion can be conditionally preserved for the optimal classifiers w.r.t each query, based on the cost function (2.3), which is also a risk function. This result is summarized in *Theorem 4*.

**Theorem 4.** *In ASSL-Boost, if the active learning procedure is efficient than random iid sampling, which means  $C_l(F_{al}) \leq C_{al}(F_{al})$ , for any  $\delta > 0$ , with probability at least  $1 - \delta$ ,*

$$(3.14) \quad C_T(F_{al}) \leq C_{al}(F_{al}) + \sqrt{\frac{h(\mathcal{H}, l) - \ln(\delta)}{l}}.$$

$C_T(F)$  is the true cost for  $F$ .  $C_{al}(F)$  is the empirical cost based on the selected samples under active learning.  $C_l(F)$  is the empirical cost based on the iid samples.  $F_{al}$  is the optimal classifier for  $C_{al}(F)$ .  $l$  is the number of the labeled samples for both active learning and random sampling.  $h$  is the capacity function.

The validity of the precondition  $C_l(F_{al}) \leq C_{al}(F_{al})$  is the key issue for *Theorem 4*. It means the active learning result need to be better than the learning result based on random sampling with the same number of labeled samples, as higher optimal cost leads to smaller objective (2.5).

This issue is analyzed in many works both theoretically [10, 7, 8, 2] and empirically [23]. It is known that active learning can save sufficient learning effort for certain learning result compared with random sampling in many situations. As a result, the presupposition can be achieved and the conclusion is realistic.

From *Theorem 4*, we realize that to alleviate the initial over-fitting, we should keep the term,  $h(\mathcal{H}, l)/l$ , in the upper bound relatively small. Though, as far as we know there is no explicit expression for the change of  $h(\mathcal{H}, l)$  during the boosting process, it is usually considered the complexity of the classifier ensemble becomes higher, such as the VC-dimension, expressed by the upper bound [14, 9]. In active learning process, when there are few labeled samples we should use a relatively simple classifier ensemble with small size, which has a small  $h(\mathcal{H}, l)$ . With the increase of the labeled samples, more classifiers can be added on. Thus, we make the boosting query committee varying in an incremental manner. This can improve the query efficiency. Besides, it also saves considerable running time.

---

#### Algorithm 1 FASSL-Boost Algorithm

---

**Input:** data  $D = \{D_L, D_U\}$ , base classifier  $f$ , trade-off coefficient  $\alpha$

**Initial:** distributions  $W_0(\mathbf{x}_i) = \frac{1}{l+\alpha u}$  for samples in  $D_L$ ,  $W_0(\mathbf{x}_j) = \frac{\alpha}{l+\alpha u}$  for samples in  $D_U$ , semi-supervised data set  $S_0 = D$ ,  $t = 1$

**repeat**

**Step 1:**

  Fit  $f_t$  using  $S_{t-1}$  and  $W_{t-1}$

**if** error for  $f_t$ ,  $\varepsilon_t > \frac{1}{2}$  **then**

    stop

**end if**

  Compute  $\omega_t = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t}$

  Update  $W_t(\mathbf{x}_i) = W_{t-1}(\mathbf{x}_i) e^{-\omega_t y_i f_t(\mathbf{x}_i)}$ .

**Step 2:**

  Query the most valuable data using (2.9)

  Update  $S_{t-1}$  using current classifier ensemble  $F_t$

$t \leftarrow t + 1$

**until** error for  $F_t$  not decrease or  $t = T$

**Output:** classifier  $F = \sum_t \omega_t f_t$ .

---

### 3.2 The Implementation of The Algorithm

We propose the algorithm, Fast ASSL-Boost (FASSL-Boost) under the same framework, based on the query by incremental committee mechanism. Nevertheless, we solve the original Newton update process in another way. In this algorithm, the series  $\{C_{S_n}(F)\}$  is still used to approximate  $C_G(F)$ , while active learning is carried out as soon as the semi-supervised boosting procedure finds a new classifier. At last, it combines every classifier for final ensemble. The flowchart is shown in table Algorithm 1. Moreover, a typical cost curve in iterations is shown in Fig. 1 (c). This curve describes a similar path and achieve the same optimal cost as the lower envelope of Fig. 1 (a).

The solution of the optimal problem using Newton iteration becomes:

$$(3.15) \quad F = F_1 + \frac{\frac{\partial C_{S_1}(F)}{\partial F}|_{F_1}}{\frac{\partial^2 C_{S_1}(F)}{\partial^2 F}|_{F_1}} + \dots + \frac{\frac{\partial C_{S_n}(F)}{\partial F}|_{F_n}}{\frac{\partial^2 C_{S_n}(F)}{\partial^2 F}|_{F_n}} + \dots$$

From *Corollary 2*, we know that after sufficient queries, the partial derivatives of the semi-supervised data set cost approximate the partial derivatives of the genuine data set cost as good as possible. So there exists some  $N$  such that it is reasonable to use cost model  $C_{S_n}(F)$  to approximate  $C_G(F)$ , for  $n > N$ . We sum up all first  $(N + 1)$  terms in (3.15),

$$(3.16) \quad F_{N1} = F_1 + \frac{\frac{\partial C_{S_1}(F)}{\partial F}|_{F_1}}{\frac{\partial^2 C_{S_1}(F)}{\partial^2 F}|_{F_1}} + \dots + \frac{\frac{\partial C_{S_n}(F)}{\partial F}|_{F_N}}{\frac{\partial^2 C_{S_n}(F)}{\partial^2 F}|_{F_N}}.$$

Then the solution is rewritten as:

$$(3.17) \quad F \approx F_{N1} + \frac{\frac{\partial C_G(F)}{\partial F}|_{F_{N1}}}{\frac{\partial^2 C_G(F)}{\partial^2 F}|_{F_{N1}}} + \dots$$

We consider it as a new Newton procedure with initial point  $F_{N1}$  and objective functional  $C_G(F)$ .

With sufficient queries and iterations, (3.17) converges to the genuine data set optimal solution. It means the FASSL-Boost will converge to the optimal solution of the genuine data set boosting.

**3.3 Complexity** The time complexity for previous ASSL-Boost methods are of order  $O(NTQF(N))$ , as in [1, 12].  $N$  is the number of data.  $F(N)$  is the time complexity of the “base learner”.  $Q$  is the size of candidate query set, which approximates  $N$  in this paper.  $T$  is the iteration times for each boosting algorithm. Algorithms can be parallelizable w.r.t.  $N$  and  $Q$ , but not  $T$  [1]. The time complexity of our new algorithm is of order  $O(NQF(N))$ , which reduces the time complexity a lot.

## 4 Experiments

**4.1 Boosting Benchmarks Learning Results** In these experiments, the comparison is performed on six benchmarks of the boosting literature: twonorm, image, banana, splice, german, flare-solar. Every data set is divided into training and test sets<sup>1</sup>. The training set is used for transductive inference. We set that the initial training set has 5% labeled data, and the query procedure stops when 80% data are queried. The test set is composed by unseen samples. It is used for comparing the inductive learning result. We adopt the decision stump [16] as the base classifier for boosting, which is a popular and efficient base classifier. Experiments are conducted among random query Adaboost (RBoost), QBB [1], COMB [12] and FASSL-Boost.

In QBB, we set the iteration times  $T = 20$  for each boosting, according to [1]. The size of candidate query set is  $Q = |D_U|$ , which means we search among all the unlabeled data for the next query. In RBoost and COMB, we use the same parameters, which are  $T = 20$  and  $Q = |D_U|$ . For COMB and FASSL-Boost, we can initialize the semi-supervised data set using any classifier. The nearest neighbor classifier is used here. We use minimum margin query criterion for QBB, COMB and FASSL-Boost. All our report is averaged over 100 different runs<sup>2</sup>.

<sup>1</sup>The date and relative information are available at <http://www.first.gmd.de/~raetsch/>.

<sup>2</sup>20 different runs for experiments on splice and image.

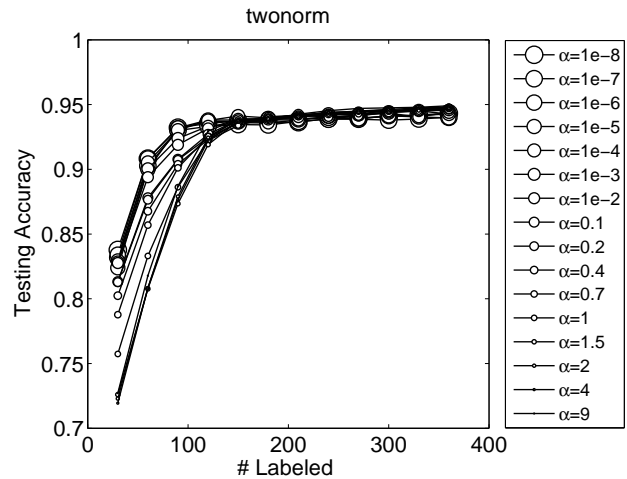


Figure 2: Representative test curves of FASSL-Boost for different  $\alpha$  on twonorm, each averaged over 100 trials. The curve with bigger marker represents a smaller  $\alpha$ .

**4.1.1 The Effect of  $\alpha$ :** We have the experiments demonstrate the effect of different  $\alpha$  for the learning result in Fig 2. It shows that  $\alpha$  should be small enough in our experiments. Thus it limits the effect of unlabeled data. If the effect is not limited, the initial labeled data will be submerged in the huge amount of unlabeled data, as there are too many unlabeled data with complex distribution. We want to use the manifold information from unlabeled data and prevent the harmful over-fitting to them. We can also use the parameter adjustment method as in [13], dynamically change  $\alpha$  w.r.t. the iteration steps. We only use a fix small  $\alpha$  in our experiments for convenience. In COMB and FASSL-Boost, we set  $\alpha = 0.01$ .<sup>3</sup>

**4.1.2 The Comparison of Learning Results:** We give both transductive and inductive learning results in our experiments. Transductive inference is a main performance, as all active learning methods compared are pool-based [19, 23]. Fig. 3 shows FASSL-Boost has the best transductive learning results. More important, it has better performance from the beginning most of the time. The result also shows that the conventional methods perform worse than random query in some situations, while FASSL-Boost does not.

Strong induction ability is a good characteristic for boosting, so we also compare the inductive inference results in Fig. 4. It shows that FASSL-Boost performs rel-

<sup>3</sup>However the algorithms are not very sensitive to the choice of  $\alpha$ , when  $0 \leq \alpha < 0.1$ .

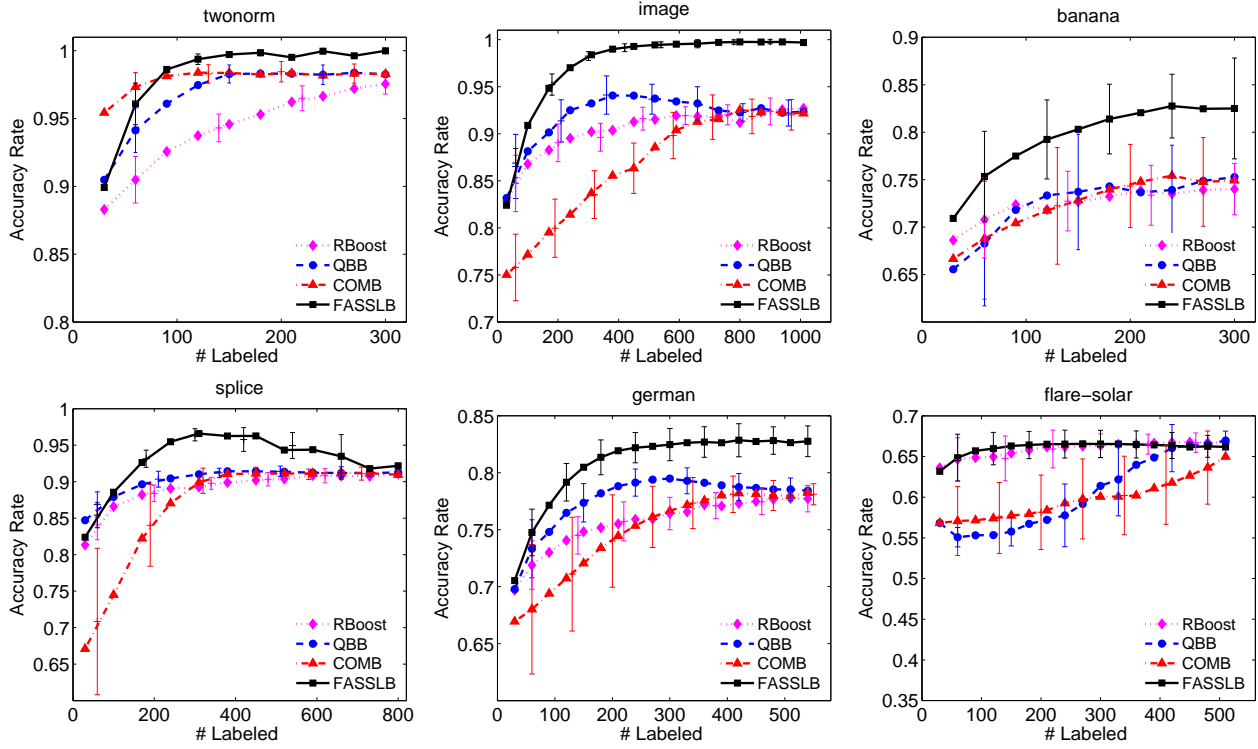


Figure 3: Transductive inference results for RBoost, QBB, COMB and FASSL-Boost (FASSLB) on boosting benchmarks.

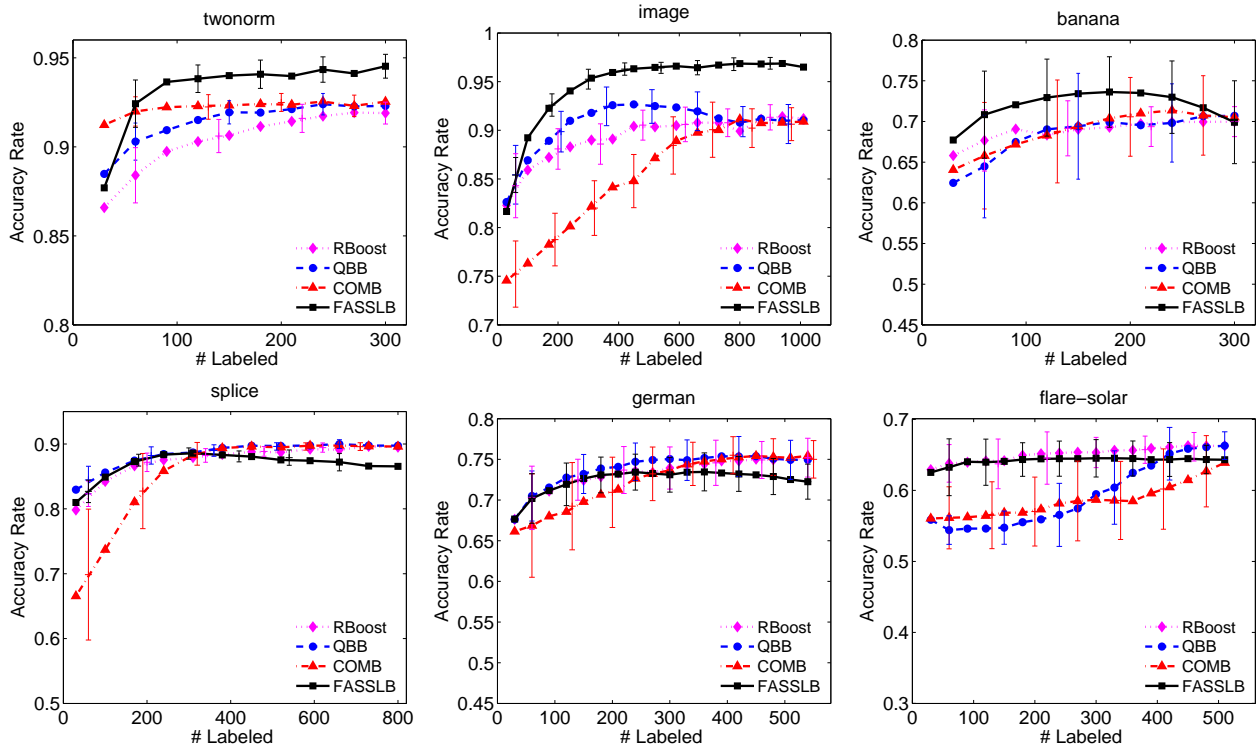


Figure 4: Inductive inference results for RBoost, QBB, COMB and FASSL-Boost (FASSLB) on boosting benchmarks.

atively best among all methods. However, the inductive accuracy decreases with too many queries for FASSL-Boost on some data sets. It may have two causes. One is that the query is too abundant to decrease the error, which means the useful data are queried out and the left query is only to apply to useless samples. The other is that boosting may slightly over-fit with too many iterations sometimes [14]. However, querying 80% samples is not practical and just to show the full query and learning processes. For real problems, users seldom query so many data, then it is naturally an early stop for FASSL-Boost. Moreover, we also could use other early stop methods for boosting to control the query process and the number of committee members.

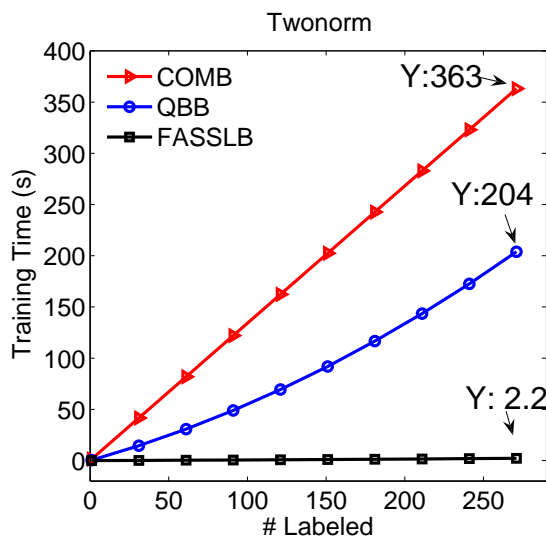


Figure 5: Training time comparison on twonorm. It is to point out for the same learning result FASSL-Boost needs not query so many data, which means its running time is shorter.

**4.1.3 The Running Time:** Fig. 5 shows the transductive learning time for the three active learning methods, labeled by machine. The experiments are running under Matlab R2006b, on a PC with Core2 Duo 2.6GHz CPU and 2G RAM. The curves show the FASSL-Boost is much more economic.

**4.2 MNIST Learning Results** In above experiment, the data sets used are all benchmarks for boosting methods. Next, we give both transductive and induc-

tive learning results on MNIST<sup>4</sup>, which is a real-world data set for handwritten digits with 70,000 samples.

The comparison experiments are performed on six binary classification tasks: 2 vs 3, 3 vs 5, 3 vs 8, 6 vs 8, 8 vs 9, 0 vs 8, which are more difficult to classify than other pairs, as the two digits in each task are much similar to each other.

All our report is averaged over 20 different runs. In each run, the samples for each digit are equally divided into two sets at random, one training set and one test set. It is for the same use as previous experiments, for comparing both the transductive and inductive learning results. As there are much more samples in this problem, we set the initial training set has 0.1% labeled data, and the query procedure stops when 10% data are queried. Experiments are conducted among random query Adaboost (RBoost), QBB [1], COMB [12] and FASSL-Boost. All the other settings are the same with last experiments.

The results are shown in Fig. 6 and Fig. 7. Our efficient method still gives the best learning performance.

**4.3 The Comparison of ASSL Methods** There are also some other well defined active semi-supervised learning methods [19, 20, 24]. [19] and [20] are proposed for specific applications and difficult to be generalized into a common learning problem as in our experiments. So we compare FASSL-Boost with *Zhu's* label propagation method with active learning [24], which is a state-of-the-art method. Label propagation originally is a transductive approach. Though [25] explained that it could be extended to unseen data, this needs plenty of extra computation, which limits its usefulness. As in *Zhu's* method and other methods based on graph, a weight matrix should be build up. It is a costly work. On the other hand, data may not satisfy the “cluster assumption” [4] very well. In this situation, label propagation with active learning cannot get satisfying result. We compare active learning label propagation with FASSL-Boost for data sets with complex distributions.

We set the same parameters for FASSL-Boost as in section 5.1. For *Zhu's* method, we establish the weight matrix in different ways and use the best result we have gotten to compare with FASSL-Boost. Results in Fig. 8 show our method performs better. And it is less dependent on data distributions.

## 5 Conclusion and Discussion

In this paper we present a unified framework of active and semi-supervised learning boosting, and develop a

<sup>4</sup>The original data and relative information are available at <http://yann.lecun.com/exdb/mnist/>



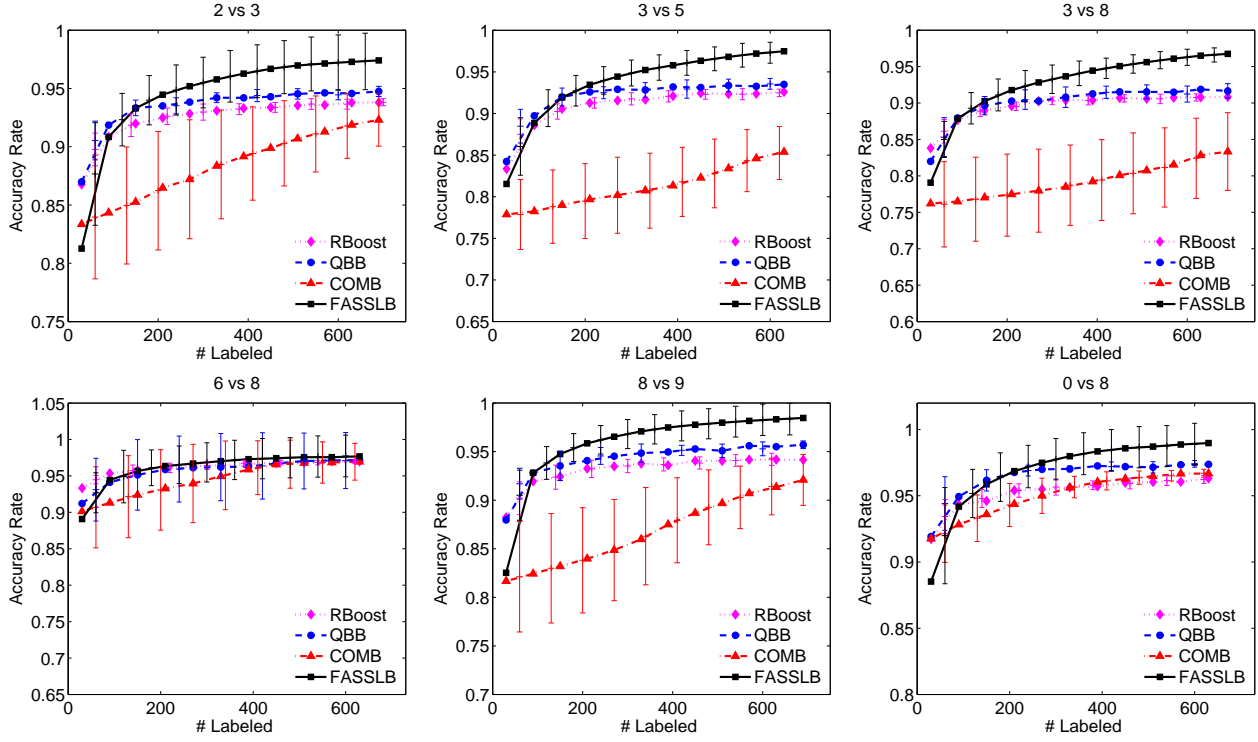


Figure 6: Transductive inference results for RBoost, QBB, COMB and FASSL-Boost (FASSLB) on MNIST.

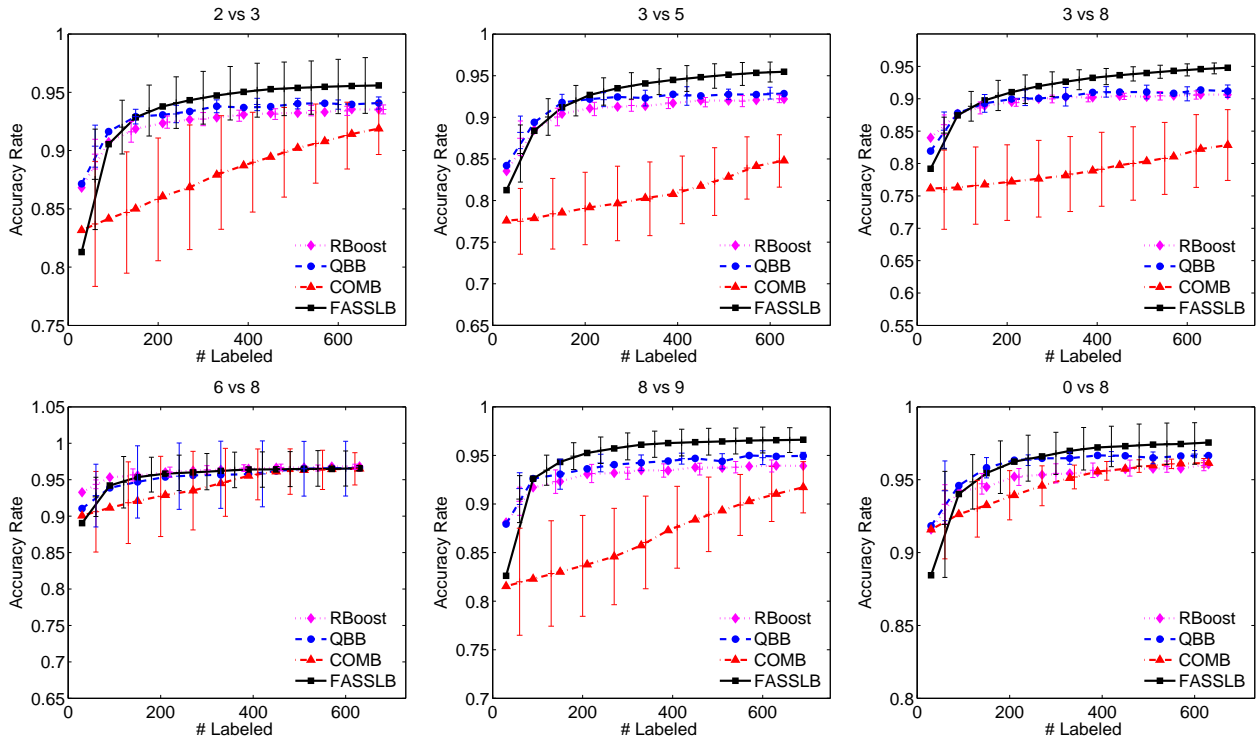


Figure 7: Inductive inference results for RBoost, QBB, COMB and FASSL-Boost (FASSLB) on MNIST.

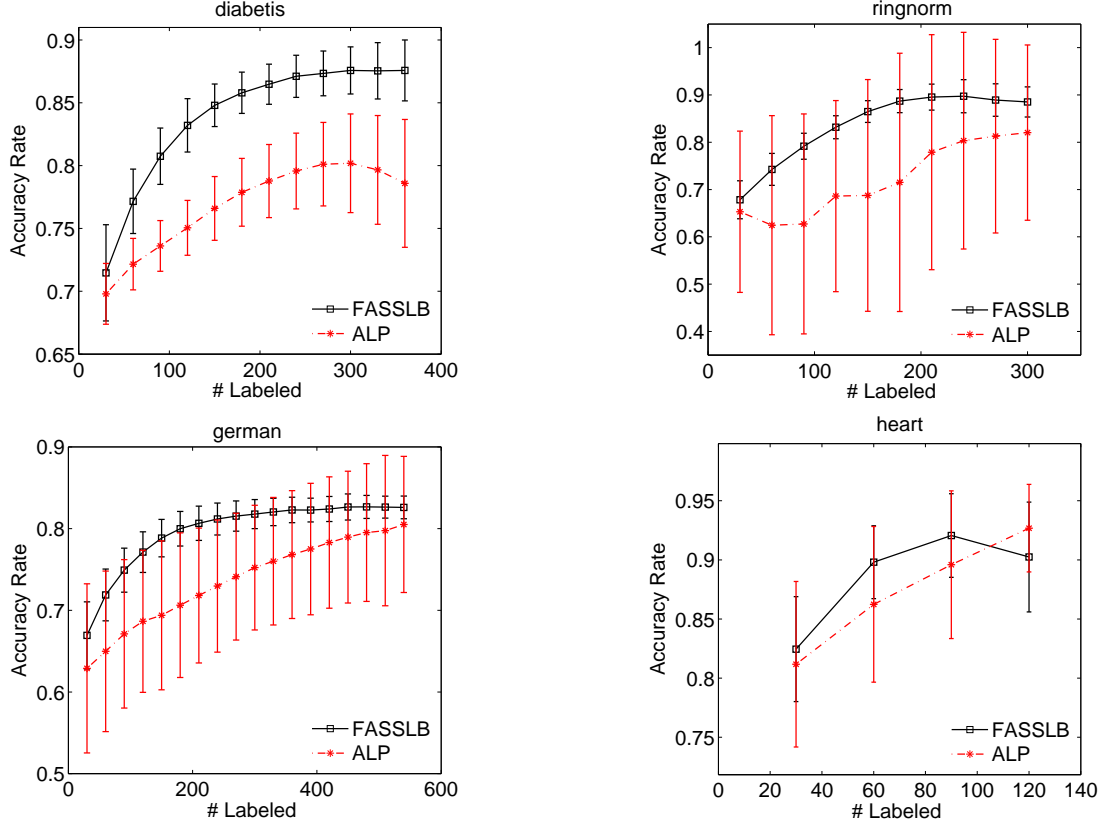


Figure 8: Comparison results of FASSL-Boost (FASSLB) and label propagation with active learning (ALP).

practical algorithm FASSL-Boost based on query by incremental committee mechanism, which rapidly cuts the training cost with improved performance. Previous SS-MarginBoost, QBB and COMB are all special cases in this framework.

Though our algorithm is in myopic mode, they can be easily generalized to batch mode active learning methods. We can select several data having large margins in different margin clusters. Using different  $C_{S_n}(\cdot)$  in different iteration step to approximate  $C_G(\cdot)$ , we can find other active semi-supervised learning boosting methods, which may lead to new discovery. Moreover, our framework can be extended to general active semi-supervised learning process. For any “meta-method” with cost functional satisfying the conditions in our theorems and corollaries, we can develop a corresponding ASSL algorithm. The novel explanation for semi-supervised and active learning combination may be found. This framework shows that the minimum margin sample is not always the best choice. We would like to work on finding a more efficient query criterion for future study.

## Appendix A: Proof of Theorem 1

**Lemma 1.** *For a certain classifier  $F$ , the cost of boosting for genuine data set is no less than the cost of boosting for semi-supervised data set under any queries. That is*

$$C_{S_n}(F) \leq C_G(F), \forall n \text{ and } F.$$

*Proof :* For  $y \in \{-1, +1\}$ ,  $e^{-|F(\mathbf{x}_i)|} \leq e^{-y_i F(\mathbf{x}_i)}$ ,  $\forall (\mathbf{x}_i, y_i) \in G$ .

We have:

$$\begin{aligned} & \sum_{(\mathbf{x}_i, y_i) \in D_{L \cup n}} e^{-y_i F(\mathbf{x}_i)} + \alpha \sum_{\mathbf{x}_i \in D_{U \setminus n}} e^{-|F(\mathbf{x}_i)|} \\ & \leq \sum_{(\mathbf{x}_i, y_i) \in D_{L \cup n}} e^{-y_i F(\mathbf{x}_i)} + \alpha \sum_{\mathbf{x}_i \in D_{U \setminus n}} e^{-y_i F(\mathbf{x}_i)}, \\ & \forall n \text{ and } F. \end{aligned}$$

Then  $C_{S_n}(F) \leq C_G(F)$ ,  $\forall n$  and  $F$ , as  $\alpha \leq 1$ .

□

**Lemma 2.** *The cost  $C_{S_n}(F)$  for semi-supervised data set with  $n$  queries is no more than the cost  $C_{S_{n+1}}(F)$  with  $n+1$  queries, for any classifier  $F$ .*

*Proof:*

$$\begin{aligned} & \alpha (\sum_{\mathbf{x}_i \in D_{U \setminus (n-1)}} e^{-|F(\mathbf{x}_i)|} + e^{-|F(\mathbf{x}_q)|}) \\ & \leq \alpha (\sum_{\mathbf{x}_i \in D_{U \setminus (n-1)}} e^{-|F(\mathbf{x}_i)|} + e^{-y_q F(\mathbf{x}_q)}). \end{aligned}$$

adding

$$\sum_{(\mathbf{x}_i, y_i) \in D_L} e^{(-y_i F(\mathbf{x}_i))} + \alpha \sum_{(\mathbf{x}_i, y_i) \in D_n} e^{(-y_i F(\mathbf{x}_i))}$$

to each side, and using  $\alpha \leq 1$ , we get Lemma 2,

$$C_{S_n}(F) \leq C_{S_{n+1}}(F) \quad \forall F.$$

□

**Theorem 1.** *The cost  $C_{S_n}(F)$  after  $n$  queries composes a monotonically non-decreasing series of  $n$  converging to  $C_G(F)$ , for any classifier ensemble  $F$ . We have  $C_{S_1}(F) \leq C_{S_2}(F) \leq \dots \leq C_{S_n}(F) \leq \dots \leq C_{S_u}(F) = C_G(F)$ .*

*Proof:* Using Lemma 1 and 2, we get directly Theorem 1.

□

**Corollary 1.** *If the cost function is convex for margin, the minimum value of  $C_{S_n}(F)$ ,  $C_{S_n}(F_{S_n})$ , composes a monotonically non-decreasing series of  $n$  converging to  $C_G(F_G)$ , which is the minimum cost for genuine data set boosting. That is  $C_{S_1}(F_{S_1}) \leq C_{S_2}(F_{S_2}) \leq \dots \leq C_{S_n}(F_{S_n}) \leq \dots \leq C_{S_u}(F_{S_u}) = C_G(F_G)$ .*

*Proof:* As in [18], if the cost function is convex for margin, boosting under this cost can get a global minimum solution  $F_{S_n}$ . So

$$C_{S_n}(F_{S_n}) \leq C_{S_n}(F_{S_{n+1}}) \leq C_{S_{n+1}}(F_{S_{n+1}}) \leq \dots \leq C_{S_{u-1}}(F_{S_u}) \leq C_G(F_G).$$

□

**Corollary 2.** *If the  $k$ th partial derivative of any cost functional exists and is finite, a series of  $\{\frac{\partial C_{S_n}(F)^k}{\partial^k F}\}$  can be composed, which converges to the  $k$ th partial derivative of cost functional  $C_G(F)$ ,  $\frac{\partial C_G(F)^k}{\partial^k F}$ .*

*Proof:* The partial derivatives for two costs are the same in labeled part. The only difference will appear in unlabeled part. As in AnyBoost [18], cost is additive among all data:

$$C(F) = \frac{1}{l+u} \sum_{i \in D} c_i(F).$$

It is the same with the  $k$ th order partial derivatives,

$$\frac{\partial C(F)^k}{\partial^k F} = \frac{1}{l+u} \sum_{i \in D} \frac{\partial c_i(F)^k}{\partial^k F}.$$

Thus the difference of derivatives between genuine data set cost and semi-supervised data set cost is:

$$\delta = \frac{1}{l+u} \sum_{\mathbf{x}_i \in D_{U \setminus n}} \left| \frac{\partial c_{S_n i}(F)^k}{\partial^k F} - \frac{\partial c_{G i}(F)^k}{\partial^k F} \right| \leq \frac{n-u}{l+u} \Delta,$$

where  $\Delta$  is the biggest gap of the derivative for any given  $F$  among the data set. Its finiteness can be ensured by the finiteness of the derivative.

For any  $\epsilon > 0$ ,  $\frac{n-u}{l+u} \Delta < \epsilon$  needs

$$n > u - (l+u) \frac{\epsilon}{\Delta}.$$

As  $\frac{\epsilon}{\Delta} > 0$ , then we get  $n \leq u$ . So there exists feasible  $n$  making the difference small enough. And the result is the same for any order partial derivatives, including zero which is the cost itself as in Theorem 1.

□

## Appendix B: Proof of Theorem 2

**Theorem 2.** *If the minimum of  $C_{S_n}(F)$ ,  $C_{S_n}(F_{S_n})$ , is equal to the final genuine data set minimum cost  $C_G(F_G)$  for certain  $F_{S_n}$ , this function  $F_{S_n}$  is also an optimal classifier for genuine data set boosting.*

*Proof:* We have already known from Corollary 1,

$$C_{S_n}(F_{S_n}) \leq \dots \leq C_{S_u}(F_{S_u}) = C_G(F_G),$$

If  $C_{S_n}(F_{S_n}) = C_G(F_G)$ , the equality is easily got:

$$C_{S_n}(F_{S_n}) = \dots = C_{S_u}(F_{S_u}) = C_G(F_G).$$

This means that the queries after  $n$  get the same label as pseudo labels for the unlabeled data, so

$$C_{S_n}(\cdot) = \dots = C_{S_u}(\cdot) = C_G(\cdot),$$

and  $F_{S_n}$  is also an optimal classifier for  $C_G(\cdot)$ .

□

## Acknowledgments

This research was supported by National Science Foundation of China ( No. 60835002 and No. 60675009 ).

## References

- [1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of Fifteenth International Conference of Machine Learning*, pages 1–9, 1998.
- [2] M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *In: Proc. The 21st Annual Conference on Learning Theory (COLT)*, pages 45–56, 2008.
- [3] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. In *Proceedings of 20th International Conference on Machine Learning*, pages 19–26, 2003.
- [4] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
- [5] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.
- [6] F. d’Alche Buc, Y. Grandvalet, and C. Ambroise. Semi-supervised marginboost. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Proceedings of the Advances in Neural Information Processing Systems 14*, pages 553–560, 2002.

- [7] S. Dasgupta. Coarse sample complexity bounds for active learning. In *Neural Information Processing Systems 2005*, pages 235–242, 2005.
- [8] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Neural Information Processing Systems 2007*, pages 353–360, 2007.
- [9] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- [10] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2):133–168, 1997.
- [11] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28:337–374, 2000.
- [12] T. Gokhan, H.-T. Dilek, and R. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005.
- [13] Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 593–600, 2007.
- [14] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer-Verlag, Berlin, Germany, 2001.
- [15] J. Huang, S. Ertekin, Y. Song, H. Zha, and C. L. Giles. Efficient multiclass boosting classification with active learning. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, pages 297–308, 2007.
- [16] W. Iba and P. Langley. Induction of one-level decision tree. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 233–240, 1992.
- [17] V. S. Iyengar, C. Apte, and T. Zhang. Active learning using adaptive resampling. In *Proceedings of the ACM SIGKDD*, pages 91–98, 2000.
- [18] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 221–246. MIT Press, Cambridge, MA, USA, 2000.
- [19] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. In *In: Proc. Internat. Conf. on Machine Learning (ICML)*, pages 359–367, 1998.
- [20] I. Muslea, S. Minton, and C. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *In: Proc. Internat. Conf. on Machine Learning (ICML)*, pages 435–442, 2002.
- [21] H. S. Seung, M. Oppor, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Workshop on Computational Learning Theory*, pages 287–294, 1992.
- [22] M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *The Journal of Machine Learning Research*, 7:141–166, 2006.
- [23] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proceedings of the 17th International Conference of Machine Learning*, pages 999–1006, 2000.
- [24] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference of Machine Learning Workshop*, pages 58–65, 2003.
- [25] X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From gaussian field to gaussian processes. Technical Report CMU-CS-03-175, School of Computer Science, Pittsburgh, PA, 2003.