A New Framework for Multiclass Classification Using Multiview Assisted Adaptive Boosting

Avisek Lahiri, Prabir Kumar Biswas

Indian Institute of Technology, Kharagpur

Abstract. Multiview representation of data is common in disciplines such as computer vision, bio-informatics, etc. Traditional fusion methods train independent classifiers on each view and finally conglomerate them using weighted summation. Such approaches are void from inter-view communications and thus do not guarantee to yield the best possible ensemble classifier on the given sample-view space. This paper proposes a new algorithm for multiclass classification using multi-view assisted supervised learning (MA-AdaBoost). MA-AdaBoost uses adaptive boosting for initially training baseline classifiers on each view. After each boosting round, the classifiers share their classification performances. Based on this communication, weight of an example is ascertained by its classification difficulties across all views. Two versions of MA-AdaBoost are proposed based on the nature of final output of baseline classifiers. Finally, decisions of baseline classifiers are agglomerated based on a novel algorithm of reward assignment . The paper then presents classification comparisons on benchmark UCI datasets and eye samples collected from FERET database. Kappa-error diversity diagrams are also studied. In majority instances, MA-AdaBoost outperforms traditional AdaBoost, variants of AdaBoost, and recent works on supervised collaborative learning with respect to convergence rate of training set and generalization errors. The error-diversity results are also encouraging.

1 Introduction

Adaptive Boosting[1, 2] yields a strong classifier by combination of several feeble rules of thumb. After each boosting round, AdaBoost updates a weight distribution based on difficulty of the sample space at that round. The final classifier is a weighted combination of individual trained classifiers. Though initially formulated for supervised learning, AdaBoost has been used extensively in semi supervised learning under multi view setting. In applications such as computer vision, biomedical signal processing, bioinformatics, data is acquired from different view points(feature spaces) [3, 4]. Each view has its own discriminative property which is absent in other views. Usually statistical methods [5, 6] are applied for extracting the best discriminative feature from each view. By dimensionality reduction, such methods yield a compact representation of data. But these methods tend to ignore the localized subtle features in each view which are usually beneficial for classification in presence of data perturbation.

Multiview learning in semi supervised learning tries to minimize manual labeling effort by iteratively learning on labeled and unlabeled instance spaces. The pioneering work of Query-by-Committee (QBC)algorithm[7] required that the views be independent of each other. The work was closely followed by Co-training [8] which still preserved the restrictive assumption of QBC. Since then there has been plethora of research on multiview semi supervised learning [9–14]. Recently, Co-Training-by-Committee [15] eliminates the requirement that the representative views be mutually orthogonal.

There is dearth of literature on collaborative multiview learning for multiclass classification in supervised learning. This is the primary motivation of the paper. In the era of Big Data we need a scalable learning framework to learn from different information sources. Previous works have focused on combining classifiers learnt on different views by fusion methods. Early fusion manifests data in a macro environment by combining discriminating features from the available views and then trains a single classifier. Late fusion trains separate classifiers on each view and ultimately combines them by plurality voting. Empirical studies performed on multimedia domain reveals that late fusion tends to perform better than early fusion [16]. But neither of the two methods encompasses assistive learning communications across views. Some recent works have focused on multiview learning in supervised learning arena. Liu et al. proposed Co-AdaBoost [17, 18] for classification of software document based on Co-training algorithm. Koco et al. proposed an algorithm (Mumbo)[19, 20] for multiclass classification with multiview collaborative learning. Recent attempts to model frameworks for assisted multiview learning in supervised learning is the primary invigoration of this paper.

2 Contributions of Proposed Work

The paper proposes a new supervised learning framework(MA-AdaBoost) for multiclass classification where a sample space is represented by multiple views. The paper places no restrictions on orthogonality of the feature spaces. The proposed work differs from Co-AdaBoost in the following aspects:

1. Co-AdaBoost mandates the views be disjoint for effective learning. The proposed MA-AdaBoost does not place any such restrictions on the views.
2. Co-AdaBoost is limited to binary classification with maximum two view representations. MA-AdaBoost is designed for multiclass classification and can be scaled for any finite cardinality view sets.
3. Co-AdaBoost conglomerates the boosted classifiers by simple weighted majority voting but MA-AdaBoost formulates a novel reward function for mixing ensemble learners.

Some of the key differences between Mumbo and MA-AdaBoost are:

1. For each view Mumbo maintains a cost matrix M(i,j) which denotes cost of assigning label j to training example x_i. For a V view problem, the storage requirement is $\mathcal{O}$(V*m*K), where m and K are cardinality of instance

space and label space respectively. Such space requirements are debatable in many real life problems. MA-AdaBoost calculates mislabeling cost on global basis after each round of assisted communication and hence need not store misclasification costs over local views.

2. Mumbo assumes the presence of a major view which is assisted by several minor views. The minor views intervene only if the average error on major view exceeds than that of random guessing. But selection of such major view from real life data is tedious and undermines the fundamental purpose of multiview learning. MA-AdaBoost adaptively rewards the views based on classification performances on each boosting round and thus the user is free from tediously selecting the best possible view.

At a particular boosting round (t), both Co-AdaBoost and Mumbo considers a sample-view space to be boostable if

$$\sum_{i:h_{t,v}(x_i)\neq y_i} W_{t,v}(i) \leq 0{\cdot}5 \tag{1}$$

whereas MA-AdaBoost deems a space boostable if

$$\prod_{v=1}^{V}\left[\sum_{i:h_{t,v}(x_i)\neq y_i} W_{t,v}(i)\right] \leq 0{\cdot}5 \tag{2}$$

where $W_{t,v}$ is weight distribution over view v and $h_{t,v}(x)$ is a local trained hypothesis on view v. Naturally MA-AdaBoost imposes less rigid restrictions on local hypotheses compared to Mumbo and Co-AdaBoost.

3 MA-ADABOOST ALGORITHM

This section formally introduces the MA-AdaBoost algorithm in the context of supervised learning for multiview assisted multiclass classification.

3.1 Initial Parameters

MA-AdaBoost initializes with learning space X=$\{(x_1,y_1),\ (x_2,y_2),..(x_m,\ y_m)\}$, label space L=$\{$1,2,.K$\}$ and view space VS=$\{v_1,v_2,.v_V\}$. Each $y_i \in$ L. An uniform weight $W_1(i)$=1/m is initiated over all x_i and v_i. A single learning algorithm such as ANN, SVM, C4.5, etc.. is selected as local hypothesis over each view.

3.2 Inter View Communication and Group Learing

Misclassification cost of a local hypothesis $h_{t,v}(x)$ over a view v on t^{th} boosting round is given by

$$P_{W_t}(h_{t,v}(x_i)) \neq y_i \tag{3}$$

After each boosting round, the local hypotheses communicate their classification performances and the overall misclassification cost of the entire learning group is assigned as:

$$\chi_t = \prod_{v_1}^{v_V} P_{W_t}(h_{t,v}(x_i) \neq y_i) \tag{4}$$

Such a cost function obviates the strict restrictions of Equation 1. MA-AdaBoost then updates the weight distribution W_t via a new scalable framework. During a boosting round, the easiest example x_i is given least weight if it is correctly classified across all views in previous iteration. Likewise, highest weight is assigned to the most onerous example. Intermediate weights of x_i are scaled according to cumulative misclassification occurances over the representative views. Difficulty of x_i is measured by the metric $\gamma_t(i)$, given by:

$$\gamma_t(i) = \frac{2j}{V} - 1; j \in [0, 1, 2, ...V] \tag{5}$$

where an example x_i has been misclassified on j views. Let

$$\beta_t = 0.5 * log\left(\frac{1-\chi_t}{\chi_t}\right) \tag{6}$$

represents the learning weight of entire group. A low misclassification cost χ_t ensures high β_t. Then the weight distribution $W_t(i)$ is updated as:

$$W_{t+1}(i) = \frac{W_t(i) * exp(\beta_t * \gamma_t(i))}{N_t} \tag{7}$$

where N_t is a normalization constant to preserve $W_{t+1}(i)$ as a distribution. Such a scalable weight distribution ensures that an arduous example is collaboratively learnt by all views.

3.3 Conglomerating Ensemble Decisions

The paper presents a novel framework for combining decisions of local view hypotheses. Let $S_{t,v}$ be a set such that,

$$S_{t,v} = \{x_i|\ h_{t,v}(x_i) = y_i\} \tag{8}$$

Classification accuracy of $h_{t,v}(x)$ is given by $\eta_{t,v} = |\ S_{t,v}|$/m. Performace reward of $h_{t,v}(x)$ is formulated as $R_{t,v}$:

$$R_{t,v} = \sum_{i:h_{t,v}(x_i)=y_i} W_t(i) * |h_{t,v}(x_i)| - \sum_{j:h_{t,v}(x_j)\neq y_j} (1 - W_t(j)) * |h_{t,v}(x_j)| \tag{9}$$

The overall performace metric of $h_{t,v}(x)$ is determined as :

$$P_{t,v} = \eta_{t,v} * (1 + R_{t,v}) \tag{10}$$

Such a reward based metric emphasizes those hypotheses which correctly classify difficult examples with high confidence than those which correctly classify easy examples with high confidence. High penalty is incurred on misclassifying an easy example with high confidence. Depending on the final classification space of $h_{t,v}(x)$, MA-AdaBoost has two variants: i. MA-AdaBoost.V1 and ii. MA-AdaBoost.V2.

3.4 MA-AdaBoost.V1

In this version the final output domain (D.V1) of $h_{t,v}(x)$ is defined as D.V1={1,2,3...K}. Final classifier is given by

$$F_{fin}(x) = \left\lfloor \frac{\sum_{t=1}^{T} \sum_{v=1}^{V} P_{t,v} * h_{t,v}(x)}{K * \sum_{t=1}^{T} P_{t,v}} \right\rfloor \tag{11}$$

where $\lfloor a \rfloor$ represents nearest integer to (a).

3.5 MA-AdaBoost.V2

In this version $h_{t,v}(x)$ yields confidence vector about each class instead of crisp labels. Thus $h_{t,v}(x) \in \mathbf{R}^{KX1}$ and output domain (D.V2) $\in$ [0,1]. Final classifier is given by

$$F_{fin}(x) = \underset{k \in L}{\operatorname{argmax}} \left[\frac{\sum_{t=1}^{T} \sum_{v=1}^{V} P_{t,v} * |h_{t,v}^{k}(x)|}{\sum_{t=1}^{T} P_{t,v}} \right] \tag{12}$$

where $|h_{t,v}^{k}(x)|$ is prediction confidence of class k. The sequential steps of MA-AdaBoost are showed in Algorithm 1.

Algorithm 1 MA-AdaBoost for Multiclass Classification

Input:

- Learning Space: X={(x_1,y_1), (x_2,y_2),..(x_m, y_m)}
- Label Space: L={1,2,.. K}
- View Space: VS={v_1, v_2,..v_V }
- Local View Hypothesis: $h_{t,v}(x)$
- Weight Distribution: W_1
- Total Boosting Rounds: T

For t=1 to T

1. Train local hypotheses on all views

2. Calculate misclassification cost of $h_{t,v}(x)$ from Equation (3)
3. Calculate χ_t from Equation (4)
4. Calculate β_t from Equation (6)
5. Calculate difficulty of x_i from Equation (5)
6. Update weight distribution $W_t(i)$ from Equation (7)
7. Evaluate performance of $h_{t,v}(x)$ from Equation (10)

END FOR

Output:
Use Equation (11) or (12) for final classifier $F_{fin}(x)$

---

4 Simulation Results

This section presents the classification performances of MA-AdaBoost and compares with Co-AdaBoost, Mumbo, and various boosting methods which use late fusion combination.

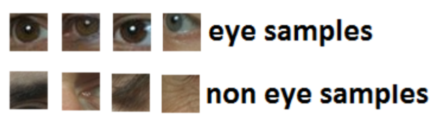

Fig. 1. Samples cut from FERET database for training ensemble classifiers.

4.1 Two Class Classification

For two class classification eye and non eye samples of 32X32 pixels are manually cut from FERET database[21, 22]. Total 5923 eye samples and 6123 non eye samples are collected. Few examples are shown in Fig.1. Each sample is transformed to gray scale space and 2D Haar transformation space. 2D Haar transformation of an image yields three matrices for emphasizing horizontal,vetical and diagonal edges and fourth matrix as an average image[23]. Without loss of generality we perform singular value decompositions of gray scale space and vertical edge space. 32 eigen values from gray scale space and 16 eigen values from vertical edge space act as two views of the system. We train a 2-layer back propagation ANN with one hidden layer on each view. We perform 10 rounds of boosting with 50 iterations of training ANN in each round.The training set, validation set and test set are divided in 60:20:20 proportions and we follow 5-fold validation for determing the optimum regularization parameter(λ) of ANN.

We report the training set performances in Table I. At onset of boosting, WNS[24] creates a sub sample space by identifying the most informative training examples.WNS speeds up AdaBoost but tends to compromise on accuracy.

Table 1. Comparison of training set accuracy rates of different ensemble classifiers for eye classification on FERET database.

T	Iterations	WNS	AdaBoost.M2	AdaBoost.Group	Mumbo	Co-AdaBoost	V1[a]	V2[b]
	05	66·0	67·8	69·3	67·6	62·4	60·0	64·0
2	25	73·0	74·6	75·0	75·9	73·0	75·0	78·4
	50	75·0	76·2	77·2	78·0	75·8	79·8	82·4
	05	68·2	70·2	71·2	69·2	65·6	63·4	67·8
5	25	79·8	81·4	82·0	80·7	78·8	80·8	85·0
	50	83·2	85·6	86·2	88·0	87·5	87·8	92·0
	05	74·8	77·9	80·0	79·0	75·2	75·0	76·4
10	25	85·3	87·8	88·6	93·8	90·6	96·2	98·0
	50	86·0	88·0	91·0	94·0	93·2	97·0	99·0

[a] **Proposed: MA-AdaBoost.V1**
[b] **Proposed: MA-AdaBoost.V2**

AdaBoost.Group[25] proposes to train independent hypotheses on each view. The hypotheses are optimized by maximizing F_1 scores on respective views. Finally the local hypotheses are combined by majority voting. From Table 1 we see that MA-AdaBoost outperforms the traditional fused based boosting algorithms by considerable margins. At low iterations, due to dearth of training, the total misclassification cost χ_t of MA-AdaBoost is high and thus hinders the learning rate by increasing β_t. So MA-AdaBoost manifests inferior learning compared to AdaBoost, WNS, and AdaBoost.Group at low iterations. But β_t increases rapidly with further training of ANN. Experiments show that after 15 rounds of training, MA-AdaBoost starts yielding superior performance compared to boosting. During final combination, MA-AdaBoost.V1 allows combination of crisp labels from each local hypothesis but MA-AdaBoost.V2 allows combination over entire label space and is therefore more expressive and superior. On avarage, after 50 training iterations, MA-AdaBoost.V1 outperforms WNS, AdaBoost.M2 and AdaBoost.Group by 8.·63%, 7·1%, and 6·93% respectively and the corresponding margins for MA-AdaBoost.V2 are 10·95%, 8·2%, and 7·4% respectively. Co-AdaBoost and Mumbo perform comparable to MA-AdaBoost.V1 and is outperformed by MA-AdaBoost.V2 by an average margin of 4·95% and 4·63% respectively. Mumbo tends to perform superior at low iterations. At low iterations, mislabelling cost is robustly dealt by Mumbo by maintaining 2 cost matrices and allowing only the best discriminative view to classify an example. Upper bound on training set error for MA-AdaBoost ϵ_{MA},:[26]

$$\epsilon_{MA} \leq \; 2^T \prod_{t=1}^{T} \sqrt{\chi_t \; (1 - \chi_t)} \tag{13}$$

while for AdaBoost.M2, WNS, and AdaBoost.Group, the error bound ϵ_A:

$$\epsilon_A \leq \; 2^T \prod_{t=1}^{T} \sqrt{\theta_t \; (1-\theta_t)} \tag{14}$$

where $\theta_t = \left[\sum_{v=v_1}^{v_V} \sum_{t=1}^{T} \sum_{j:h_{t,v}(x_j)\neq y_j} W_{t,v}(x_j)\right]$ is misclassification cost on round (t) and $W_{t,v}(x)$ is the weight distribution on view (v) during boosting round (t). Now, $\chi_t < \theta_t$, and thus $\epsilon_{MA} < \epsilon_A$. So the convergence rate of MA-AdaBoost is faster compared to traditional boosting.

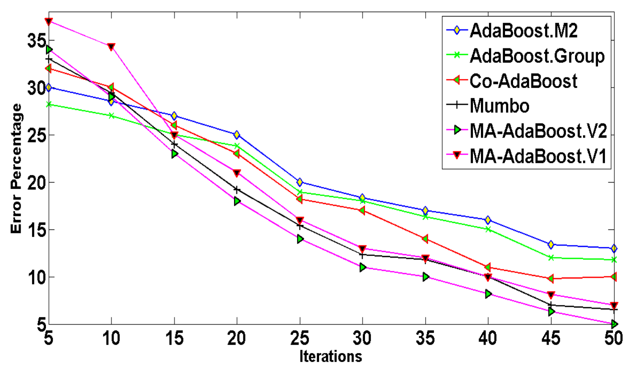

Fig. 2. Comparison of average generalization error rates of various ensemble classifiers trained for classification of eyes on FERET database. The classifiers are trained over 10 boosting rounds with 50 iterations of training per round.

In Fig.2 we report the generalization error rates of the ensemble classifiers. In each boosting round, we increment ANN training by 5 iterations and study the effect on test set performances. Due to space scarcity we report the average over 10 boosting rounds. Horizontal axis represents number of times ANN is trained per boosting round while vertical axis delineates generalization error rates. We see that MA-AdaBoost offers better test set performance comapared to fusion based boosting. On average MA-AdaBoost has 4·6% and 3·8% less error rate than AdaBoost.M2 and AdaBoost.Group respectively while the same margins for MA-AdaBoost.V2 are 6.·1% and 4.·2% respectively. For visual clarity we did not plot the curve for WNS. Specifically, MA-AdaBoost.V1 and MA-AdaBoost.V2 outperforms WNS by 11·5% and 13·2% respectively. Mumbo and Co-AdaBoost yield comparable results.

4.2 Multiclass Classification

In this section we test our model on several datasets from the benchmark UCI database[27]. Co-AdaBoost and AdaBoost.Group cannot be compared with as those models are apt only for binary classes. Weighted majority voting(WMV)[28] is an enhanced boosting algorithm. WMV is based on *boosting by resampling* AdaBoost which selects a subspace of original sample distribution and formulates a hypothesis. A correction factor is introduced while updating weight distribution for enhancing accuracy of AdaBoost.

We select 5 datasets from UCI repository as shown in Table 2. The datasets span over domains such as biology, commerce, game playing, and forensics. We randomly divide a dataset into 2 views and train 2-layer backpropagation ANN with one hidden layer on each view. For investigating the rate of convergence of training error, we stop simulation as soon as one of the algorithms achieve more than 90% training set accuracy; after this landmark, the convergence rate is sluggish for all algorithms. We report the training set performances in Table 3. In each round of boosting we increment the training iterations in steps of

Table 2. Datasets selected from UCI repository for training ensemble classifiers

Dataset	Instances	Attribues	Classes
Glass	214	10	7
Iris	150	4	3
Balance Scale	625	4	3
Car Evaluate	1728	6	4
Connect-4	67557	42	3

30 and investigate the classification spaces of competing algorithms. In Table 3, T denotes the boosting round at which we first achieve greater than 90% accuracy from any one of the ensemble classifiers while N denotes the number of times ANNs are trained per boosting round. We note that convergence rates of multiview based collaborative algorithms are significantly higher than that of non-cooperative boosting algorithms.

Let after (t) rounds of boosting the final classifier be $H_{final}(\boldsymbol{X}, t, v)$=f($h_{t,v}(\boldsymbol{X})$), where $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent training data set and label set respectively and $h_{t,v}(\boldsymbol{X})$ is the hypothesis on view (v). We define Γ as :

$$\Gamma = \underset{t}{\operatorname{argmin}} \; \{P_r(H_{final}(\boldsymbol{X}, t, v) \neq \boldsymbol{Y}) < 10\%\} \tag{15}$$

i.e. Γ represents the minimum boosting required to acquire 90% training set accuracy. If algorithms 1 and 2 have different Γ_1 and Γ_2 respectively, then we define *"edge-of-convergence"* e_{Γ_1,Γ_2} as:

$$e_{\Gamma_1,\Gamma_2} = \frac{\Gamma_1 - \Gamma_2}{\Gamma_2}; \quad \Gamma_1 > \Gamma_2 \tag{16}$$

$e_{\Gamma_1,\Gamma_2} \geq 0$ signifies faster convergence rate for algorithm 2. We report e_{Γ_1,Γ_2} in Fig.3 with MA-AdaBoost.V2 as algorithm 2 and the competitors as algorithm

1. Mumbo performs comparable to MA-AdaBoost and is thus not compared in Fig.3. We see that the boosting algorithms perform worst on Iris dataset. Due to scarcity of sufficient attributes on each view, the algorithms cannot train the independent hypotheses. The lack of proper training combined with dearth of collaboration leads to drastic drop of learing rate. Independent boosting algorithms fair best on Connect-4 dataset due to presence of plethora of data and attributes.

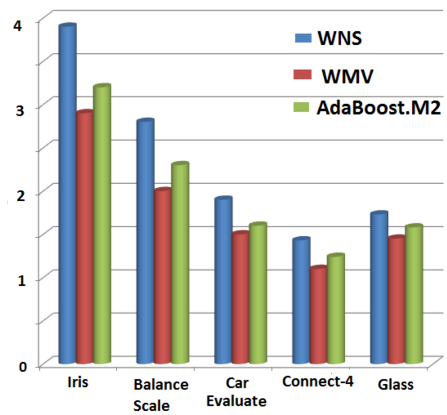

Fig. 3. Comparison of e_{Γ_1,Γ_2}(y axis) of MA-AdaBoost.V2 with variants of boosting on various UCI datasets(x axis).

Table 3. Study of training error convergence rates on UCI datasets. **T:** total boosting rounds elapsed before any one of the ensemble classifiers achieves 90% accuracy rate on a particular dataset. **N:** ANN training iterations per boosting round. The first classifier to achieve 90% accuracy on a particular dataset is marked in bold.

Dataset	(T,N)	WNS	AdaBoost.M2	WMV	Mumbo	V1[a]	V2[b]
Iris	(4,60)	55·0%	61·2%	63·4%	89·3%	88·0%	**93·2%**
Balance Scale	(3,30)	65·4%	70·8%	72·9%	91·0%	87·4%	**92·6%**
Car Evaluate	(5,60)	73·4%	81·8%	83·0%	89·3%	90·4%	**92·7%**
Glass	(4,30)	73·2%	78·3%	81·4%	**90·7%**	87·0%	88·7%
Connect-4	(7,90)	84·8%	88·2%	88·0%	90·5%	89·8%	**94·3%**

[a] **Proposed: MA-AdaBoost.V1**
[b] **Proposed: MA-AdaBoost.V2**

4.3 Kappa-Error Diversity Analysis

An ideal ensemble classifier should possess highly veracious members and should simultenously disagree within the group in most instances [29]. Such a requirement imposes a trade-off between miscellany and accuracy in a classifier space. Kappa-error diagram [30] is a visualization measure to study error-diversity trend of ensemble clasifiers. For two classifiers $f_a(x)$ and $f_b(x)$, a contingency table $\mathbf{A}$ is defined such that $\mathbf{A}(k,l)$=1 whenever $f_a(x_i)$=k and $f_b(x_i)$=l; x_i is a training example and (k,l) are class labels.. A high trace value of $\mathbf{A}$ manifests agreement between $f_a(x)$ and $f_b(x)$ on most instances. Define

$$\theta_1 = \frac{\sum_{l=1}^{K} \mathbf{A}(l,l)}{m}; K = number\ of\ classes \tag{17}$$

as probability of agreement between $f_a(x)$ and $f_b(x)$. Define

$$\theta_2 = \sum_{k=1}^{K} \left(\sum_{l=1}^{K} \frac{\mathbf{A}(k,l)}{m} \sum_{l=1}^{K} \frac{\mathbf{A}(l,k)}{m} \right) \tag{18}$$

as probabilty of random agreement between $f_a(x)$ and $f_b(x)$; m is the cardinality of sample space. Kappa statistic $\kappa_{a,b}$ is defined as

$$\kappa_{a,b} = \frac{\theta_1 - \theta_2}{1 - \theta_2} \tag{19}$$

$\kappa_{a,b}$=0 signifies that $f_a(x)$ and $f_b(x)$ agree by chance while $\kappa_{a,b}$=1 signifies agreement on every instance. Let $\epsilon_{a,b}$ represent mean misclassification cost on combined classification spaces of $f_a(x)$ and $f_b(x)$. Kappa-error diagram is a scatter plot of $\epsilon_{a,b}$ versus $\kappa_{a,b}$. After (n) rounds of boosting, ${}^{n}C_2$ combinations of pairwise classifiers can be selected from the ensemble classifier space .

In Fig.4 we report the error-diversity patterns of various ensemble classifiers over the 5 UCI datasets. We compare results using MA-AdaBoost.V2 which gives slightly better result than MA-AdaBoost.V1. The scatter clouds of the classifiers are highly overlapping; hence we plot only the centroids of the cluster clouds. The horizontal and vertical axis represent $\kappa_{a,b}$ and $\epsilon_{a,b}$ respectively; both axes are scaled between [0,1] for visualization. An ideal ensemble classifier has low values for both $\epsilon_{a,b}$ and $\kappa_{a,b}$ and thus its centroid of scatter cloud should occupy the third quadrant of error-diversity diagram.We deduce the following inferences from Fig.4.

– Ensemble spaces trained by collaborative learning possess more accurate member hypotheses compared to ensemble spaces trained by non communicating boosting algorithms. Conglomeration of more accurate member aids in better generalization ability and supports the results in Table 4 which reports test set performaces of ensemble classifiers.
– Scatter clouds of Mumbo and MA-AdaBoost.V2 tend to concentrate in the third quadrant of $\epsilon_{a,b}$-$\kappa_{a,b}$ space and thus tend towards realization of ideal ensemble learning.

- Inter hypothesis agreement is slightly more in Mumbo than in MA-AdaBoost.V2. In Mumbo, an arduous example x_i is removed from sample space of weak hypotheses and only the best hypotheses classify it; agreement thus increases among the best members. But as pointed out earlier, this increases computational and memory costs of Mumbo.
- Error clouds of WMV are usually concentrated below error clouds of AdaBoost.M2. WMV adaptively reduces probabilties of correctly classified examples so that classifiers can concentrate on hard examples. Such a modified weight distribution tends to enhance boosting classification accuracy.
- Members within WNS classifier space are least accurate. WNS forms a subspace from distribution space of AdaBoost by identifying the most discriminative examples. This reduced distribution subsapce reduces classification efficieny of WNS but aids in reduced execution time.

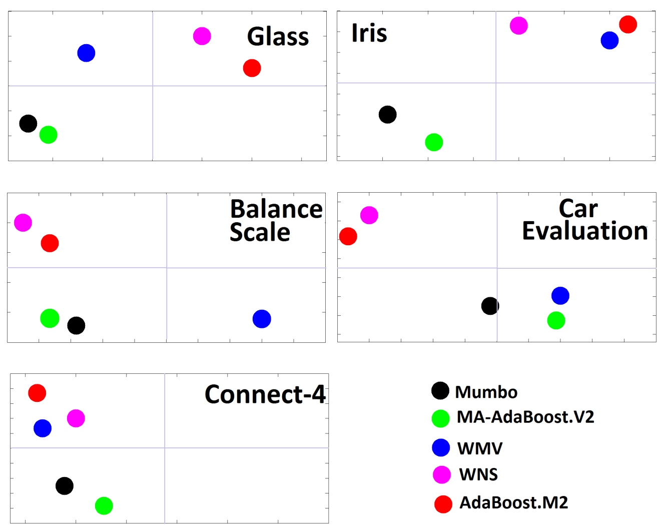

Fig. 4. Study of Kappaa-Error diagrams on 5 UCI datasets. Horizontal axis represents agreement metric $\kappa_{a,b}$. Vertical axis represents mean misclassification cost $\epsilon_{a,b}$. Both axes are scaled to span between [0,1]. The dots denote the centroids of error-diversity scatter clouds of various ensemble classifiers.

In Table 4 we report the generalization error rates of the ensemble classifiers after 10 rounds of boosting. The best results are marked in bold.

4.4 An Interesting Application in Computer Vision

" ONE HUNDRED SPECIES LEAVES" [31] is a challenging database in computer vision. The dataset contains 100 classes of leaves of different species. Sixteen different variants of each species are photographed as an RGB image against

Table 4. Generalization error rates of various ensemble classifiers on UCI datasets.

Datasets \ Algorithm	AdaBoost.M2	WNS	WMV	Mumbo	MA-AdaBoost.V2
Iris	33·5	35·1	20·3	7·3	**3·2**
Glass	19·8	21·3	15·8	**5·3**	5·9
Balance Scale	18·5	20·3	11·2	9·1	**4·1**
Connect-4	35·4	27·0	19·2	15·6	**14·3**
Car Evaluate	23·2	24·9	21·7	15·3	**9·2**

white background. Each of these samples is then characterized by three distinct 16D feature spaces: shape signature, texture histogram, and margin-feature histogram. Thus the sample space consists of 1600 samples; each sample is represented by a 64D feature vector.

The multiview setting of this dataset is apt to be applied on our model. We train 2-layer ANN with one hidden layer on each of these views for five rounds of boosting and hundred iterations of training ANN per round. We follow a sixteen fold evaluation to compare our results with [32], which reports the best classification accuracy of 99.3%, though their classification method is based on probabilistic K-NN. The results are reported in Table 5. After five rounds of

Table 5. Accuracy rates of various ensemble classifiers on the *" ONE HUNDRED SPECIES LEAVES"* dataset. Each ensemble classifier consists of ANN members which are trained 100 times per round of boosting on each of the three view spaces.

T	AdaBoost.M2	WMV	Mumbo	MA-AdaBoost.V1	MA-AdaBoost.V2
2	73·4	77·8	85·3	92·3	95·4
3	75·6	79·0	86·2	94·5	97·8
4	78·2	79·9	87·3	96·4	98·2
5	80·2	81·0	90·1	97·0	98·8

boosting, MA-AdaBoost.V1 achieves generalization accuracy rate of 97% while that achieved by MA-AdaBoost.V2 is 98.8%. Both results are comparable to the best reported accuracy rate of 99.3% [32]. At the same instant the accuracy rates of AdaBoost.M2 and WMV are 80.2% and 81.3% respectively. On average over five boosting rounds, MA-AdaBoost.V1 outperforms AdaBoost.M2 and WMV by an average margin of 18.2% and 15.5% respectively. The corresponding margins achieved by MA-AdaBoost.V2 are 20.4% and 18.1% respectively.

Note that performance of Mumbo deteriotes on this dataset compared to MA-AdaBoost. The three feature spaces are comparable to each other in clasification accuracy. But Mumbo requires a major view which will be assisted by several minor views. Absence of such a view arrangement minifies group learning in Mumbo, thereby reducing classification efficacy. MA-AdaBoost.V1 outperforms Mumbo by 7.8% while MA-AdaBoost.V2 outperforms Mumbo by 10.3%.

Experimental success on *"ONE HUNDRED SPECIES LEAVES"* dataset bolsters our claim that MA-AdaBoost is superior compared to other boosting based classification methods and is ready to be embraced in domains such as computer vision where an object of interst is frequently represented in multitude of view spaces.

5 Conclusion and Direction for Further Work

The paper presents a new algorithm, MA-AdaBoost in the context of multiview based multiclass classification for supervised learning. The core invigoration of MA-AdaBoost is to foster assistive learning across views. Importance of an example is ascertained based on its difficulties of classification on all the representative views. A single weight distribution is then updated based on importance of sample space across all view spaces. Such an update rule encourages that an example be learnt collaboratively by all views.

The paper then proposes a novel method for conglomerating decisions of hypotheses from multiple views. During combination, MA-AdaBoost assigns more importance to a hypothesis which correctly classifies a difficult example with high confidence than a hypothesis which correctly classifies an easy example with high confidence. Similarly, higher loss is suffered by a hypothesis if it misclassifies a naive example with high credence than misclassifying an arduous example with high conviction.

Experimental results confirm the boosting property of MA-AdaBoost; the training error decreases with increase of boosting rounds. The underlying assumption about the accuracy of an individual hypothesis trained on a view is much more relaxed in MA-AdaBoost compared to Mumbo and Co-AdaBoost. The rate of convergence of training set error is shown to be superior for MA-AdaBoost compared to Mumbo and Co-AdaBoost. Extensive simulations over samples from FERET, UCI and " ONE HUNDRED SPECIES LEAVES" databases manifiest the superior generalization capability of MA-AdaBoost. The paper also studies Kappa-Error diagrams for analyzing performances of ensemble classifiers on test sets. The diagrams reveal that ensemble space on MA-AdaBoost consists of more accurate members compared to ensemble spaces of other algorithms such as WMV,WNS, Mumbo, and Co-AdaBoost.

In future we wish to perform a thorough mathematical analysis to comprehend the changes MA-AdaBoost renders to traditional boosting. Another interesting area of investigation is to compare the performances of MA-AdaBoost using other learning platforms such as SVM, C4.5, Bayesian Networks, etc.

6 Acknowledgement

The work is financially supported by Space Technology Cell, ISRO,Ahmedabad under the *"SVD"* project. The work is dedicated to our respected Lt. Prof. Somnath Sengupta, Dept. of E&ECE, IIT Kharagpur.

References

1. Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence **14** (1999) 1612
2. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: ICML. Volume 96. (1996) 148–156
3. Zhang, H., Yang, Z., Gönen, M., Koskela, M., Laaksonen, J., Honkela, T., Oja, E.: Affective abstract image classification and retrieval using multiple kernel learning. In: Neural Information Processing, Springer (2013) 166–175
4. Masulli, F., Mitra, S.: Natural computing methods in bioinformatics: A survey. Information Fusion **10** (2009) 211–216
5. Duangsoithong, R., Windeatt, T.: Relevant and redundant feature analysis with ensemble classification. In: Advances in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference on, IEEE (2009) 247–250
6. Culp, M., Michailidis, G., Johnson, K.: On multi-view learning with additive models. The Annals of Applied Statistics (2009) 292–318
7. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proceedings of the fifth annual workshop on Computational learning theory, ACM (1992) 287–294
8. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory, ACM (1998) 92–100
9. Zhu, X.: Semi-supervised learning. In: Encyclopedia of Machine Learning. Springer (2010) 892–897
10. Li, G., Chang, K., Hoi, S.C.: Multiview semi-supervised learning with consensus. Knowledge and Data Engineering, IEEE Transactions on **24** (2012) 2040–2051
11. Cui, X., Huang, J., Chien, J.T.: Multi-view and multi-objective semi-supervised learning for hmm-based automatic speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on **20** (2012) 1923–1935
12. Wang, W., Zhou, Z.H.: On multi-view active learning and the combination with semi-supervised learning. In: Proceedings of the 25th international conference on Machine learning, ACM (2008) 1152–1159
13. Zhou, Z.H., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems **24** (2010) 415–439
14. Liu, W., Tao, D.: Multiview hessian regularization for image annotation. Image Processing, IEEE Transactions on **22** (2013) 2676–2687
15. Hady, M., Schwenker, F.: Co-training by committee: A new semi-supervised learning framework. In: Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on, IEEE (2008) 563–572
16. Snoek, C.G., Worring, M., Smeulders, A.W.: Early versus late fusion in semantic video analysis. In: Proceedings of the 13th annual ACM international conference on Multimedia, ACM (2005) 399–402
17. Liu, J., Li, J., Sun, X., Xie, Y., Lei, J., Hu, Q.: An embedded co-adaboost based construction of software document relation coupled resource spaces for cyber–physical society. Future Generation Computer Systems **32** (2014) 198–210
18. Liu, J., Li, J., Xie, Y., Lei, J., Hu, Q.: An embedded co-adaboost and its application in classification of software document relation. In: Semantics, Knowledge and Grids (SKG), 2012 Eighth International Conference on, IEEE (2012) 173–180
19. Koço, S., Capponi, C.: A boosting approach to multiview classification with cooperation. In: Machine Learning and Knowledge Discovery in Databases. Springer (2011) 209–228

20. Koço, S., Capponi, C., Béchet, F.: Applying multiview learning algorithms to human-human conversation classification. In: INTERSPEECH. (2012)
21. Philips, P., Wechsler, H., Huang, J., Rauss, P.: The feret database and evaluation procedure for face recognition algorithms. Image and Vision Computing **16** (1998) 295–306
22. Philips, P., Moon, H., Rizvi, S., Rauss, P.: The feret evaluation methodology for face recognition algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence **22** (2000) 1090–1104
23. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: Computer vision, 1998. sixth international conference on, IEEE (1998) 555–562
24. Seyedhosseini, M., Paiva, A.R., Tasdizen, T.: Fast adaboost training using weighted novelty selection. In: Neural Networks (IJCNN), The 2011 International Joint Conference on, IEEE (2011) 1245–1250
25. Ni, W., Huang, Y., Li, D., Wang, Y.: Boosting over groups and its application to acronym-expansion extraction. In: Advanced Data Mining and Applications. Springer (2008) 27–38
26. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences **55** (1997) 119–139
27. Bache, K., Lichman, M.: Uci machine learning repository (2013) http://archive.ics.uci.edu/ml.
28. Zhang, C.X., Zhang, J.S., Zhang, G.Y.: An efficient modified boosting method for solving classification problems. Journal of Computational and Applied Mathematics **214** (2008) 381–392
29. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28** (2006) 1619–1630
30. Margineantu, D.D., Dietterich, T.G.: Pruning adaptive boosting. In: ICML. Volume 97., Citeseer (1997) 211–218
31. Mallah, C., Cope, J., Orwell, J.: Plant leaf classification using probabilistic integration of shape, texture and margin features. Computer Graphics and Imaging/798: Signal Processing, Pattern Recognition and Applications (CGIM2013), Acta Press (2013) 2013–798
32. Mallah, C., Cope, J.: Probabilistic classification from a k-nearest-neighbour classifier. Computational Research **1** (2013) 1–9