ISSN: 2321-7782 (Online)
Volume 3, Issue 1, January 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Survey on Cost-Sensitive Classification-Boosting

Kaushikkumar Jadav[1]
Computer Science department
Parul Institute of Technology
Affiliated to GTU
Gujarat, India

Asst. Prof. Hiren V. Mer[2]
Computer Science department
Parul Institute of Technology
Affiliated to GTU
Gujarat, India

Abstract: ***The paper basically survey on cost sensitive boosting. It contains summary of different meta classifier like Boost,CostBoost, UBoost, Cost-UBoost, Adacost, Adaboost, CSE1, CSE2, CSE3, CSE4, CSE5. It contains the comparative study of different Meta classifier by different inputs and collects the results according to it. Experimental evaluation based on base classifier as K-Nearest Neighbor and Naïve Base with all Meta Learners. As a result of experiment we found that Cost-Boost with K-nearest neighbor that gives minimum misclassification cost and Cost-Sensitive Extension 4 with Naïve base gives the minimum number of high-cost error.***

Keywords: Boost, Cost-Boost, UBoost, Cost-UBoost, CS extensions version.

I. INTRODUCTION

Classification is techniques of identify data instances for particular class or not. It is an analysis technique in mining. Most of the Classification techniques are of error predicated framework but in some of the authentic world examples does not used error predicated alone they additionally consider the cost with them. Cost sensitive classifier is a Meta learner to make its base classifier cost-sensitive. For example medical diagnosis like someone has decease but they classified in normal then it has higher cost. In cost sensitive classification that has two main parameters are misclassification cost and high cost error reduced by some of the existing algorithm.

So this paper considering comparative study on different base classifiers like K-nearest neighbor and Naïve Base using all other existing Meta classifier. Classification is form of data analysis that extracts models describing important data classes. An accuracy of a classifier on given test set is the percentage of test set tuple that are correctly classified by classifier. Many classification techniques are developed under the error-based framework. This is not the case in real world application like fraud detection, medical diagnosis. Error based framework do not consider the cost of misclassification in building classifier model so the resulting model having large number of misclassification error and high cost error. Boosting has been shown to be an effective method of combining multiple models in order to enhance the predictive accuracy of single model.

In the second section of the paper its literature survey which describe the all existing algorithm for cost-sensitive classification. Describe the different Meta learners which are like Boost, Cost-Boost, UBoost, Cost UBoost, Adacost, Adaboost, CSE1, CSE2, CSE3, CSE4, CSE5.

In the third section it shows that experimental evaluation of all other existing Meta learners and collects the results of it. Then identify which algorithm gives the best result among all them. In that evaluation used input as ten different dataset, six cost matrix, base learner and Meta learner.

II. LITERATURE SURVEY

A survey of existing algorithm of cost sensitive classification like Boost, Cost-Boost, UBoost, Cost-UBoost, AdaCost, Adaboost, CSE1, CSE2, CSE3, CSE4, CSE5 is performed.

Boosting trees has been proven an effective method of reducing the number of high cost errors as well as the total misclassification cost. The authors of this paper have proposed two boosting techniques to incorporate the misclassification cost. The important thing to note is that both algorithms incorporates the cost of misclassification. The author nicely answered that why boosting is considered more powerful than other techniques in cost-sensitive classification.

Intuitively combining multiple models shall give more robust prediction than a single model under the situation where misclassification costs are considered. Boosting has been shown to be an effective method of combining multiple models in order to enhance the predictive accuracy of a single model Thus it is natural to think that boosting might also reduce the misclassification costs.

–**Boosting[2]** in simple words it is boosting with decision trees but as it utilizes the misclassification cost at minimum expected cost criteria while classification thus boosting becomes cost-sensitive. [2]

–**Cost-Boosting[2]** weights are updated according to misclassification cost associated with each sample. All trees are cost-sensitive. In this there have initial weights. [2]

$$W_{(t+1)}(x) = nW_{(t+1)}(x) / \sum_{y=1}^{N} nW_{(t+1)}(Y)\, Ny$$

Weights are normalized to ensure total weight is N

The only difference is that Cost-Boosting incorporates the misclassification cost during induction of the trees. It also considers the minimum expected cost criteria of boosting while classification of an example.

Discussion: boosting technique with respect to cost-boosting performs slightly down the line. It is natural because the trees generated in cost-boosting are cost-sensitive (except the first one) in terms of reduced misclassification cost and reduced number of high cost errors cost-boosting is better. The only disadvantage of the cost-boosting is that in case of change in misclassification cost all the trees have to be regenerated.

AdaCost[8]: Incorporates the misclassification cost on cost in algorithmic step with the weight update rule. Misclassification cost is considered as a part of algorithmic step. For the same a new parameter β is incorporated. Reward of correct classification cost is low when the cost is high and vice-versa. New parameter is incorporated to consider the cost of misclassification as a part of algorithmic step total cost is recuced.

UBoost [7](Boosting with unequal initial weights**)**[6] **:** Similar to AdaCost this uses C4.5S as abse classifier. Misclassification cost is incorporated only in first induction of decision tree using initial weight setting. Weights are normalized to ensure total number of examples in training is N. Algorithm also incorporates the misclassification cost at the classification stage. An error rate of each decision tree is calculated. Weights of samples are multiplied by a variable to reflect correctness of prediction of example. Means, sample is assigned '1' –correct classification '0'-incorrect classification. Weight of a sample is increased if it is misclassified else it it reduced.

Coat-[7] (UBoost with Cost-Sensitive adaption**)**[6]**:**weights are updated according to misclassification cost associated with each sample. Starting with unequal initial weights. Misclassification cost is considered in weight update method. Plus minimum expected cost criterion is used. All boosted trees are cost-sensitive except first.

Cost-Sensitive Extension versions [9]: Learning with Cost-Sensitive algorithm is essential to consider cost of misclassification into account apart from measure like accuracy and speed. Two way exist, to incorporate misclassification cost. First, the algorithm which incorporates misclassification cost into algorithmic steps. Second, cost in preprocessing stage cost-sensitive.

Author extends the Adaboost by using different constant parameter which was useful for the weight update equation in every round of building stage of model. Somehow the

Author main focus on the weight update equation and reduced misclassification cost using known Meta learners. They extend three version of Adaboost for best results. CS1,CS2,CS3 that all are the extension of Adaboost meta learner.

CSE1 first extension of Adaboost weight updates equation as follow

$$W_{(k+1)}(n) = C_\delta W_k(n)$$

$$C_\delta = cost\ of\ classification$$

CSE2 third extension of Adaboost considers the weight update rule as follow:

$$W_{(k+1)}(n) = C_\delta W_k(n) \exp(-\delta H_k(x_n))$$

CSE3 second extension of Adaboost consider the weight update rule as follow

$$W_{(k+1)}(n) = C_\delta W_k(n) \exp(-\delta H_k(x_n)\alpha_k$$

In this extension consider another α_k parameter it computes as $\alpha_k = \frac{1}{2}\ln\left(\frac{1+r_k}{1-r_k}\right)$

The author extension of Adacost which is is same as focus on initial weight is high in some costly high examples. CSE4 and CSE5 are two extension of Adacost which was also reduce the misclassification cost and number of high cost error.

CSE4 first extension of Adacost contains the weight update rule as follow:

$$W_{(k+1)}(n) = W_k(n) \exp(-\delta H_k(x_n)\beta_\delta \alpha_{k)}$$

$$\text{Where } \beta_\pm = C_n$$

CSE5 another extension of Adacost contains the weight update rule as follow:

$$W_{(k+1)}(n) = W_k(n) \exp(-\delta H_k(x_n)\beta_\delta \alpha_k)$$

$$\beta_+ = -0.5\, C_n + 0.5 \ \ and\ \beta_- = 0.5\, C_n + 0.5$$

Paper also contain the experimental evaluation of all this extensions and comparing its result with best output. Future scope of this paper said by author that above all algorithms decision tree as base classifier one can further show comparative study of choosing K-Nearest neighbor and Naïve base as base classifier and reduced the misclassification cost and number of high cost error.

III. EXPERIMENTAL EVALUATION

3.1 Experiment Setup :

In this chapter, we empirically evaluate the performances of the cost-sensitive boosting procedures (Firstly, AdaBoost, Boost, Cost-Boost, UBoost and Cost-UBoost with AdaCost and secondly, AdaBoost, CSExtension1, CSExtension2 and CSExtension3 with AdaCost).Ten two-class natural data sets from the UCI machine learning repository are used in the experiments shows the characteristics of the data sets. They are breast cancer(Wisconsin), liver disorder, credit screening, echocardiogram, solar flare, heart disease(Cleveland), hepatitis, horse colic, house-voting84, hypothyroid, king-rook versus king-pawn, pima Indian diabetes, sonar and tic-tac-toe datasets. Only two-class data sets are used because AdaBoost, in its basic form, is designed for two-class problems only.

For each of the data sets, were port the sum of six averages .The six cost matrices used are [01;20], [02;10], [01;50], [05;10], [010;10] and [01;100]. Thus, in each set of experiments (having six cost factors and Ten datasets), there are altogether 60 runs.

The key measure to evaluate the performance of the algorithms for cost-sensitive classification is the *total cost of misclassifications* made by a classifier on a test set

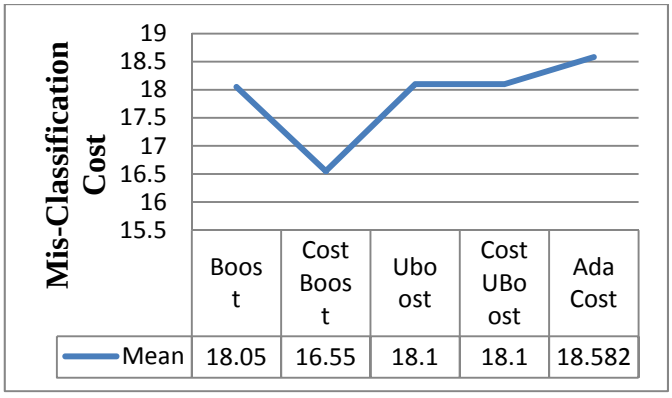
$$(\text{i.e., } \sum m = cost\,(actual\,(m), predicted\,(m))).$$

Number of high cost errors. It is the number of misclassifications associated with costs higher than unity made by a classifier on a test set. The second measure is used primarily to explain the behavior of the algorithms.

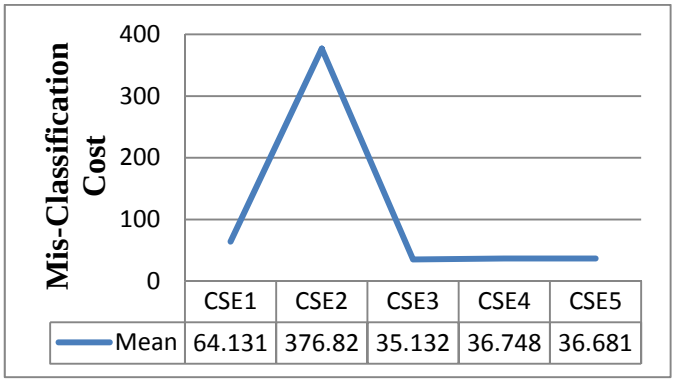
3.1 Experiment - K-NN as base classifier

3.1.1 Misclassification cost of K-Nearest Neighbour with mean value of boosting algorithm

	Boost	CostBoost	Uboost	CostUBoost	AdaCost
bcw	10	10	10	10	10
bupa	0	0	0	0	0
crx	13.33	13.33	13.33	13.33	13.33
echo	6.67	6.67	6.67	6.67	15.33
hepa	30	30	30	30	39.33
horse	120	101.67	120	120	106.33
hv84	0.5	3.83	1	1	1.5
pima	0	0	0	0	0
sonar	0	0	0	0	0
ttt	0	0	0	0	0
Mean	18.05	16.55	18.1	18.1	18.582

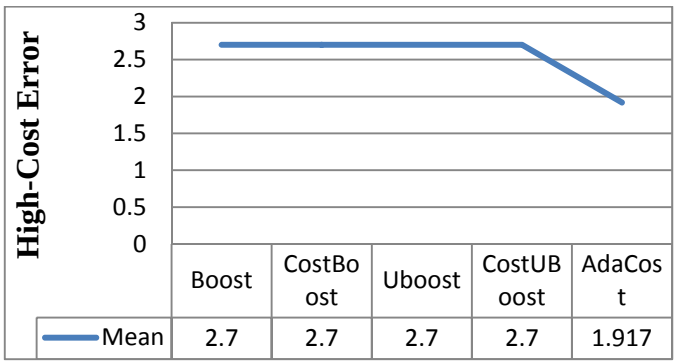


	cs1	cs2	cs3	cs4	cs5
bcw	82.33	1190.33	45.16	45.16	45.16
bupa	0	0	0	0	0
crx	232	2286.67	121	121	121
echo	117.16	60.67	25.67	25.66	25.66
hepa	91.16	77.5	56.16	56.16	56.16
horse	116.5	151	101.5	118	118
hv84	2.16	2	1.833	1.5	0.833
pima	0	0	0	0	0
sonar	0	0	0	0	0
ttt	0	0	0	0	0
Mean	64.131	376.817	35.1323	36.748	36.6813

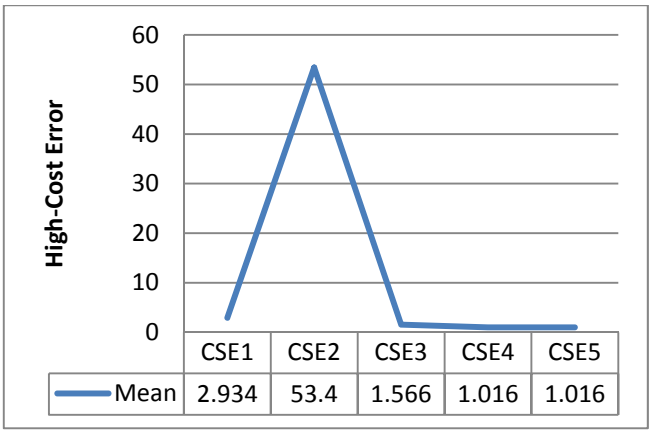


3.1.2 Number of High Cost Error of K-NN with mean value of boosting algorithm

	Boost	CostBoost	Uboost	CostUBoost	AdaCost
bcw	1.5	1	1.5	1.5	1.5
bupa	0	0	0	0	0
crx	2	2	2	2	2
echo	1	1	1	1	0.67
hepa	4.5	4.5	4.5	4.5	3
horse	18	18	18	18	12
hv84	0	0.5	0	0	0
pima	0	0	0	0	0
sonar	0	0	0	0	0
ttt	0	0	0	0	0
Mean	2.7	2.7	2.7	2.7	1.917



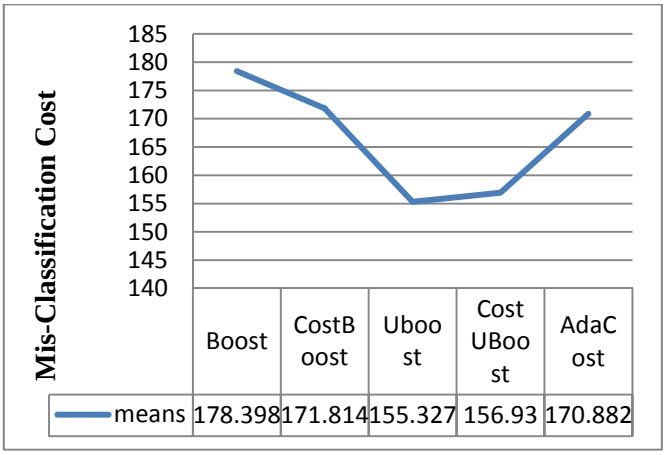
	cs1	cs2	cs3	cs4	cs5
bcw	0.5	178.5	1	1	1
bupa	0	0	0	0	0
crx	0.67	343	1.33	1.33	1.33
echo	8.5	12	0.33	0.33	0.33
hepa	13.67	0	1.5	1.5	1.5
horse	6	0.5	11.5	6	6
hv84	0	0	0	0	0
pima	0	0	0	0	0
sonar	0	0	0	0	0
ttt	0	0	0	0	0
Mean	2.934	53.4	1.566	1.016	1.016



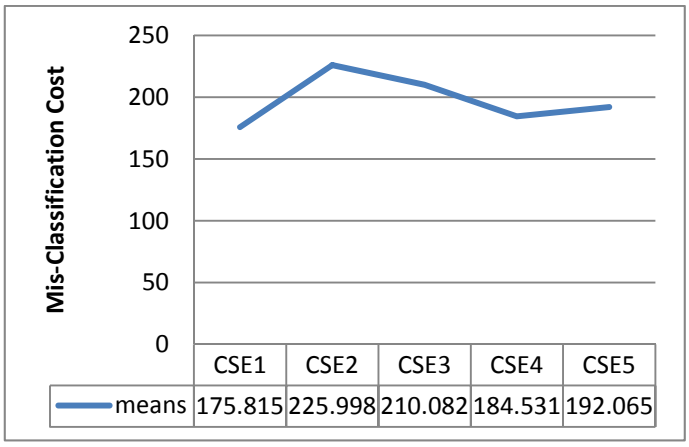
3.2 Experiment - Naïve-Base as base classifier

3.2.1 Misclassification cost of Naïve Base with mean value of boosting algorithm

	Boost	CostBoost	Uboost	CostUBoost	AdaCost
bcw	66.67	62.16	122.833	30.16	93.83
bupa	199.5	179.5	163.16	197.33	171
crx	252.833	329.33	220	269.16	277.16
echo	0	1	0.33	1	1.67
hepa	73.5	77.83	68.13	69.83	68.33
horse	308.5	219.33	188	164	107.5
hv84	83.16	75.5	78.83	72.5	82.33
pima	395.5	291.16	295.16	300.16	368.5
sonar	66.16	64	76	81.83	70.67
ttt	338.16	418.33	340.83	383.33	467.83
means	178.3983	171.814	155.3273	156.93	170.882

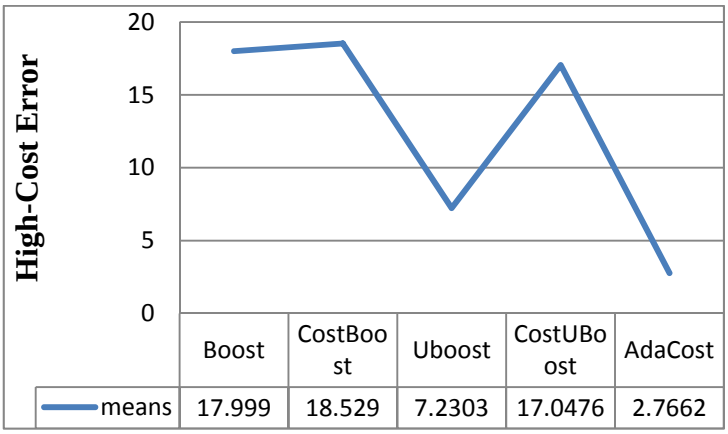


	cse1	cse2	cse3	cse4	cse5
bcw	89.5	345.16	273.33	104.33	152.16
bupa	172.16	177.16	172.5	172.16	172.33
crx	288	363	330.5	324.33	315
echo	2.5	1.166	0.833	1.33	0.166
hepa	74.33	80.5	77	76.33	75.67
horse	102.667	148	132	104.5	104.83
hv84	119.67	175	160	151.833	158.33
pima	376.83	383.33	383.33	384	383.5
sonar	64.33	107.66	92.33	47.5	79.667
ttt	468.16	479	479	479	479
means	175.8147	225.998	210.0823	184.531	192.0653

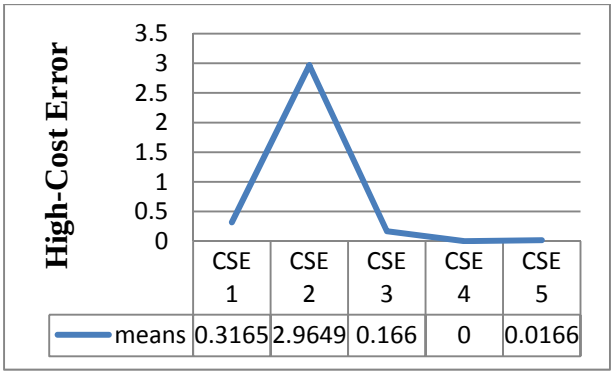


3.2.2 Number of High Cost Error of K-NN with mean value of boosting algorithm :

	Boost	CostBoost	Uboost	CostUBoost	AdaCost
bcw	3	9.16	2.33	10.33	0.67
bupa	19.83	8.16	4	10.66	0
crx	32.67	50.16	16.16	40.83	8
echo	0	0.16	0	0.166	0
hepa	8.67	10.33	2.16	8.33	2
horse	46.16	34.33	30	23.83	10.5
hv84	1.5	11.33	0.833	11.83	1.16
pima	52.16	28.83	14.16	30.67	0.166
sonar	0	8.67	0	9.67	5.166
ttt	16	24.16	2.66	24.16	0
means	17.999	18.529	7.2303	17.0476	2.7662



	cs1	cs2	cs3	cs4	cs5
bcw	0.833	0.33	0	0	0
bupa	0	3.33	0	0	0
crx	2.166	14.66	1.66	0	0.166
echo	0	0	0	0	0
hepa	0	1.833	0	0	0
horse	0	1.5	0	0	0
hv84	0.166	0	0	0	0
pima	0	0.166	0	0	0
sonar	0	7.83	0	0	0
tic-tac-toe	0	0	0	0	0
means	0.3165	2.9649	0.166	0	0.0166



IV. CONCLUSION

This paper provides details discussion on different meta learners of cost sesnsitive boosting. At best of our knowledge we believe most of cost-sensitive boosters are studied. Moreover it consider comparative study of those learnes. Identify the cost-boost learner with K-nearest neighbor gives the best performance amongst the all in misclassification cost. The reason for its better performance is weights are updated according to mislcassification cost associated with each sample. It also consider the minimum expected cost-criterion of boosting while classification of an examples. Hence it shows that different graphs of miscalssification cost and number of high cost error with two base classifier and number of different meta classifier. Analysis based on that graphs found the meta classifier are Cost-Boost with K-nearest neighbor in minimum misclassification cost and CSE4 with Naïve base in minimum number of high-cost error.

References

1. Charles Elkan, " The Foundations of Cost-Sensitive Learning " ,The International Joint Conference on AI (IJCAI'01)
2. Kai Ming Tine and Zijian Zheng," Boosting Tree for Cost-Sensitive Classification " ,Eighth International Conference on DS, Singpore 2005.
3. Susan Lomax and Sunil Vadera ," An empirical comparison of cost-sensitive decision tree in duction algorithms", July 2011.
4. Geoffrey I. Webb, "Cost-Sensitive Specialization", Proceedings of the 1996 Pacific Rim International Conference on Artificial Intelligence, Cairns, Springer-Verlag, pp. 23-34.
5. P. Domingos. "Metacost: A general method for making classifiers cost- sensitive",In KDD, pages 155–164, 1999.
6. Art ur Ferreira, "Survey on boosting algorithms for supervised and semi- supervised learning", oct. 2007.
7. Kai ming Ting and Zijian Zheng ," Boosting Cost-sensitive trees", Tenth International Conference on Discovery Science, LNAI-1038 (pp.134-145) Japan: Springer- 2007.
8. WieFan, Salvatore J. Stolfo, Junxin Zhang and Philip k. chan "AdaCost: Misclassification Cost-sensitive Boosting", 27thInternationalConferenceon Machine Learning, july2010.
9. An Empirical Evaluation of Adaboost Extensions : July 2012, By Ankit Desai IJCA'2012
10. Desai, Ankit B., and Jalpesh Vasa. "Cost-Sensitive Decision Tree Induction for Feature Selection and Sequential Minimal Optimisation for Classification: CSattribSelectorC4.5()."International Journal Of Data Mining And Emerging Technologies 3.2 (2013): 58-62.

