Applying UChooBoost algorithm in Precision Agriculture

Anastasiya Kolesnikova
Chungnam National University
Daejeon, Korea
+82-42-821-5448
a.kolesnikova@gmail.com

Chi-Hwa Song
Chungnam National University
Daejeon, Korea
+82-42-821-5448
chsong@cnu.ac.kr

Won Don Lee
Chungnam National University
Daejeon, Korea
+82-42-821-5448
wondon@hotmail.com

ABSTRACT

Informatization is primary characteristic of current agriculture stage. Precision Agriculture is destination of new technologies and principles in agriculture using digital information.

We consider data analysis as an aspect of Precision Agriculture and introduce UChooBoost applied to classification problem in agriculture.

UChooBoost is a supervised learning ensemble-based algorithm for extended data, based on bootstrap technique. UChoo classifier is used as Weak Learner. Combining hypotheses by new weighted majority voting developed for extended results expression allows UChooBoost to achieve better performance level.

Categories and Subject Descriptors

I.2.6 **[Artificial Intelligence]**: Learning - *Knowledge acquisition.*

I.5.2 **[Pattern Recognition]**: Design Methodology - *Classifier design and evaluation.*

General Terms

Algorithms, Performance.

Keywords

Classification, boosting, ensemble-based system, precision agriculture.

1. INTRODUCTION

Agriculture is oldest human activity. While there are more and more people in the world, resources for each person become limited, the need of improving productivity of farmland is evident. In this way and assuming the complexity, informatization of agriculture is needed in order to increase effectiveness of resources utilization.

Precision Agriculture concept [3] requires the use of new technologies, such as global positioning, sensors, satellites or aerial images, and information management tools. As a result, a mass of data need to be analysed so as to make a decision. Data mining techniques can be applied to perform the decision making. We consider bootstrap technique as efficient Data Mining tools to solve classification problems.

Bootstrap technique [2] is powerful approach that has achieved fame in statistical and engineering sciences. Any bootstrap ensemble-based classification algorithm consists of iteratively learning weak classifiers with respect to a distribution and compounding them to a final strong classifier. First main point in using ensemble-based algorithm is diversity among weak classifiers. To achieve it, classifiers are learned on different training data subset obtained by resampling of the original data set. Second main point is compounding of weak classifiers. There are several different schemes for compounding. In *simple majority voting*, a commonly used combination rule, each classifier votes on the class it predicts, and the class receiving the largest number of votes is the ensemble decision. In *weighted majority voting*, each classifier has a voting weight inversely proportional to its classification error. The class with the largest total vote is then declared the winner. Algebraic combination of the class values can also be used, where the class receiving the highest combined support is then chosen by the ensemble [5].

UChoo [4,7] is a decision tree classifier derived from C4.5 using the extended data expression. That expression is existent and naturally based on probabilistic theory. With the view of increasing effectiveness of compounding procedure, we introduce new weighted majority voting developed for UChoo classifier. While UChoo classifier gives extended results expression, hypotheses are compounded using the expression.

First, an extended data expression is described. Second, UChoo, decision tree algorithm, is described. Next, UChooBoost, ensemble-based algorithm, founded on new weighted majority voting, is proposed.

2. EXTENDED DATA SET EXPRESSION

Table 1 shows a common expression of the training data set. It consists in two discrete attributes, one continuous attribute and a class. The discrete attribute "Outlook" has three outcomes, 'sunny', 'overcast' and 'rain'. "Windy?" has two outcomes, 'True' and 'False'. Continuous attribute "Temperature" has values range from 60 to 80. Finally, Class has two outcomes, 'Play' and 'Don't Play'.

ICAC3'09, January 23–24, 2009, Mumbai, Maharashtra, India.

Table 1. Training data set

Outlook	Temp(^{O}F)	Windy?	Class
Sunny	70	false	Don't Play
Sunny	60	true	Play
Overcast	80	false	Play
Rain	60	true	Don't Play
Rain	70	false	Play
Rain	80	true	Don't Play

To make an extended data expression, attributes have to be represented by probabilistic values. We put simply 0 and 1 into the entry of the data set according to outcome value.

Thus, if attribute of original data set has three outcomes, as 'Outlook' in Table 1, in an extended data expression the attribute has three entry values. In fact each entry is represented by probabilistic value ranging from 0 to 1 and each event has a weight value, which shows how much important the event is. For example, if expert assumes that each observed instance has weight of 1 for the importance; the weight value shows how much important the event is, compared with an instance with a weight of 1.

Table 2 shows the extended expression of the training data set of Table 1. Adjustment is made for first event and can be done by an expert in this field. An expert says that it is 'sunny' and the temperature is around 70 and the probability to play is 2/3 no matter if it is windy or not.

An event is a collection of instances with equal attribute values distribution and class value distribution. The event which weight is 1 can be considered as the instance itself. Therefore, the number of events may not be equal. For example, in Table 2 the number of all the instances is 35 while the number of the events is 6. The number of all the instances in a data set T is denoted as |T| and the number of the event is p.

3. UCHOO ALGORITHM

The measure equations of C4.5[6] had been modified in order to be able to operate with extended data set expression.

A is an attribute and k is the number of values of a class and n is the number of outcomes of the attribute.

Class membership weight:

$$C_1(m),\ C_2(m),\ \ldots,\ C_k(m).$$

$C_i(m)$ states how much the *m*th event belongs to the class C_i.

Here, $\sum_{i=1}^{k} C_i(m) = 1$

(1)

Outcome membership weight:

$$O_{A_1}(m),\ O_{A_2}(m),\ldots,\ O_{A_n}(m)$$

$O_{A_j}(m)$ states how much the outcome value *j* in the attribute *A* can happen in the *m*th event.

Here, $\sum_{j=1}^{n} O_{Aj}(m) = 1$

(2)

T_{A_j} is the subset of *T* that has the outcome value *j* about the attribute *A*. For example, as the attribute *Outlook* has three values, the set *T* is divided into three subsets.

$Weight(m,T)$: Weight value of the *m*th event in the set *T*.

$freq(C_i,\ T)$: It is the number of instances in the set *T*, which have the class value of C_i.

In this case,

$$freq(C_i,\ T) = \sum_{m=1}^{p(T)} Weight(m,T) \cdot C_i(m)$$

(3)

$freq(C_i,\ T_{A_j})$: The number of instances in the set T_{A_j}, which have the class value of C_i.

$$freq(C_i,\ T_{A_j}) = \sum_{m=1}^{p(T)} Weight(m,T)\ \cdot C_i(m) \cdot O_{Aj}(m)$$

(4)

$|T|$: The number of instances in the set *T*. An instance means the event which weight is 1 in this case. Therefore, if an event has the weight, *W*, it means that there are *W* number of instances with the equal distribution values of attributes and class.

$|\ T_{A_j}|$: The number of instances in the set T_{A_j} .

$$|\ T_{A_j}|\ =\ \sum_{m=1}^{p(T_{Aj})} Weight(m,T_{A_j}) \cdot O_{Aj}(m) \qquad (5)$$

Therefore, the best attribute for splitting has the biggest *Gain_ratio* in the node by using the existing entropy equations with these newly defined values.

Table 2. Extended data expression

Event#	Weight(i)	Outlook			Temp(^{O}F)			Windy?		Class	
		sunny	overcast	rain	60	70	80	true	false	Play	Don't Play
1	30	1	0	0	0	1	0	1/2	1/2	2/3	1/3
2	1	1	0	0	1	0	0	1	0	1	0

3	1	0	1	0	0	0	1	0	1	1	0
4	1	0	0	1	1	0	0	1	0	0	1
5	1	0	0	1	0	1	0	0	1	1	0
6	1	0	0	1	0	0	1	1	0	0	1

The measure equations of C4.5 are:

$$Gain\_ratio(A) = Gain(A) \,/\, Split\_info(A) \qquad (6)$$

$$Gain(A) = info(T) - info_A(T) \qquad (7)$$

$$info(T) = -\sum_{i=1}^{k}(freq(C_i,T)/|T|) \cdot \log_2(freq(C_i,T)/|T|) \qquad (8)$$

$$info_A(T) = \sum_{j=1}^{n}(|T_{A_j}| / |T|) \cdot info(T_{A_j})$$

(9)

$$info(T_{A_j}) = -\sum_{i=1}^{k}(\mathrm{freq}(C_i,T_{A_j}) / |T_{A_j}|)$$

$$\cdot \log_2(freq(C_i,T_{A_j}) / |T_{A_j}|) \quad (10)$$

4. ENSEMBLE-BASED ALGORITHM

Ensemble-based algorithms create collection of WeakLearners and compound them to StrongLearner. Bootstrap technique achieves diversity among classifiers by training classifiers with a subset of the training data resampled from the training dataset. The data distribution is adapted for each new classifier in way misclassified event are more likely to appear in the next subset.

UChoo classifier with parameter of pruning is used as WeakLearner. Parameter of pruning defines weakness of classifier. For example, pruning 30% means, that each leaf of tree can consist 30% of events of "wrong" classes. Hypothesis in leaf of UChoo decision tree is determined by summing weight of events in leaf for each class and normalizing:

$$h_i = \frac{\sum_{j:y_j=C_i \,\&\, x_j \in S_{leaf}} Weight(j, S_{leaf})}{\sum_{j:x_j \in S_{leaf}} Weight(j, S_{leaf})}, \qquad (11)$$

where dataset S_{leaf} is set of training instances at the leaf.

Define here training data $S = \{(Weight_i, x_i, y_i)\}_{i=1}^{n}$, where $Weight_i \geq 0$ is weight of instance, $x_i \in X$ is ith instance in extended attribute space X and $y_i \in Y$ is class label of x_i in extended class space Y. T is number of UChooLearners. Output of UChooLearner is hypothesis $h(x) = (h_1(x), \ldots, h_k(x))$, where $h_i(x)$ states how much the event x belongs to the class C_i and $\sum_{i=1}^{k} h_i(x) = 1$.

The UChooBoost algorithm is given in Fig.1.

At first, a distribution is initialized as uniform, so all events have equal probability to be chosen into training subset TR_t. At each iteration UChooLearner is training with subset TR_t to obtain hypothesis $h^t = \{h^t(x_i) \mid x_i \in S\}$. The weights $\{Weight(S)\}$ in an extended data expression of subset TR_t are used to define hypothesis h^t and are invariable through iterations.

AdaBoost rule [1] is applied to update weights $\{w_i\}_{i:x_i \in S}$, so that weights of misclassified events are increased. In AdaBoost weights are changed according to the hypothesis, which is result of current WeakLearner. Here the compounded hypothesis is used to update weights, so performance of resulting hypothesis is improved after each iteration. Also, resampling with replacement is used. So, difficult events can be chosen to training set more than one time. For UChoo, it means that instance will have higher weight $Weight$, i.e. instance become more important and classifier focuses on such event.

Compounding hypotheses is very important part of ensemble-based algorithms. This paper proposes the new compounding method based on an extended data expression. Application of an extended data expression allows using weights of classes given by hypotheses h^t. For every class i, weights h_i^t are summed embracing all hypotheses and normalized, so that $\sum_{i=1}^{k} H_i(x) = 1$, where $H(x)$ is compounded hypothesis. This method is efficient, as it is shown by experiments.

5. EXPERIMENTS

The algorithm, proposed in this paper, is tested on Eucalyptus Soil dataset obtained from agricultural researchers in New Zealand [1] and data set collected in UCI Machine Repository [9]: soybean disease database.

Goal of the experiments is to show advantages of new probabilistic majority voting, so algorithm is compared with Learn++ ensemble-based algorithm [8], using weighted majority voting. Using UChoo decision tree classifier as weak learner in experiments for Learn++ provides an equal state for both ensemble-based algorithms.

Each data set is randomly divided into two subsets. First (70%) is used to train and second (30%) to test UChooBoost and Learn++ algorithm. On each iteration training, a set is chosen from first subset according to distribution of data set. Pruning is the parameter of UChoo decision tree that determines possible level of dominant class in a leaf.

In first experiment Eucalyptus data is analyzed.

Eucalyptus Soil Conservation data set consists of 736 events. 95 events have missing values; so finally, we consider data set of 641 events. Events, described by 19 attributes, can be classified as one of 5 classes. The objective is to determine which seedlots in species are best for soil conservation in seasonally dry hill country.

The second experiment is about soybean disease.

Soybean Disease data set consists of 683 events considering training and testing data sets overall. 136 events have missing values and are not considered; so finally, we work with data set of 547 events. Events, described of 35 attributes, can be classified as one of 19 classes.

UChooBoost Algorithm

```
Initialize $w_i^1 = D_i^1 = \frac{1}{n}$
for $t = \overline{1,T}$
    1. $D_t(i) = \frac{w_t(i)}{\sum_{j=1}^{n} w_t(j)}$
    2. Choose training subset $TR_t$ according to $D_t$
    3. Train UChooLearn with $TR_t$ and obtain hypothesis $h^t = \{h^t(x_i) \mid x_i \in S\}$
    4. Compound hypothesis $\mathrm{H}_\mathrm{i}^\mathrm{t}(x) = \frac{\sum_{j=1}^{t} h_i^j(x)}{\sum_{l=1}^{k}\sum_{j=1}^{t} h_l^j(x)}$
    5. Compute error of $H^t$ :
        $e^\mathrm{t} = \sum_{i:\arg\max_j H_j^t(x_i) \neq \arg\max_j y^j(x_i)} D_i^t$
    6. Update weights :
        $w_i^{t+1} = \begin{cases} w_i^t, \text{if } (\arg\max_j H_j^t(x_i) \neq \arg\max_j y^j(x_i)); \\ \frac{e^t}{1-e^t} \cdot w_i^t, \text{otherwise.} \end{cases}$
end for
    $H_i^{final}(x) = \frac{\sum_{j=1}^{T} H_i^j(x)}{\sum_{l=1}^{k}\sum_{j=1}^{t} H_l^j(x)}$
    $H(x) = \arg\max_{C_i} H_i^{final}(x)$
```

Figure 1. UChooBoost algorithm

Table 3. Results for Eucalyptus Soil Data

pruning,%	UChooBoost, err(%)	Learn++(UChoo), err(%)
40	32.0755	32.0755
35	37.7358	36.7925
30	31.1321	32.0755
25	33.9623	33.0189
20	37.7358	39.6226
15	33.9623	38.6792
10	34.9057	35.8491
5	34.9057	41.5094

In Table 3 and Table 4, the error rates of UChooBoost and Learn++ are shown for different pruning level. A lower error range of UChooBoost underlines the advantage of new weighted majority voting.

Table 4. Results for Soybean Disease Data

pruning,%	UChooBoost,	Learn++(UChoo),
40	4.5872	11.926
35	6.4220	8.2569
30	5.5046	6.4220
25	2.7523	3.6697
20	5.5046	11.926

15	3.6697	4.5872
10	7.3394	8.2569
5	6.4220	6.4220

Figure 2 and Figure 3 show bootstrap capability of UChooBoost: error is decreasing through iteration.

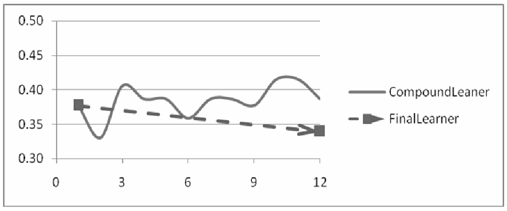

Figure 2. Error of compounded hypothesis through iterations for Eucalyptus Soil Data

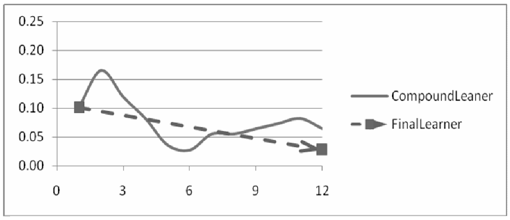

Figure 3. Error of compounded hypothesis through iterations for Soybean Disease Data

6. CONCLUSION

In this paper we have considered Data Mining as an aspect of Precision Agriculture. UChooBoost, the bootstrap ensemble-based algorithm for an extended data, shows good performance in experiments with agriculture data.

The main advantage of the algorithm is using an extended data expression to increase efficiency of compounding hypotheses which leads to improving algorithm performance.

7. REFERENCES

[1] Agricultural researchers in New Zealand. http://www.cs.waikato.ac.nz/ml/weka/

[2] B. Efron, 1979. Bootstrap methods: Another look at the jackknife, *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, 1979

[3] Dimitriadis, S., Goumopoulos. C.,2008. Applying Machine Learning to Extract New Knowledge in Precision Agriculture Applications. Panhellenic Conference on Informatics (PCI), 2008, pp. 100-104.

[4] Dong-Hui Kim, Dong-Hyeok Lee and Won Don Lee, 2006. Classifier using Extended Data Expression. *IEEE Mountain Workshop on Adaptive and Learning Systems*, July, 2006, pp. 154-159.

[5] J. Kittler, M. Hatef, R.P.W. Duin, and J. Mates, 1998. On combining classifiers, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[6] J. R. Quinlan, 1996. Bagging, Boosting, and C4.5. AAI/IAAI, vol. 1, 1996.

[7] Jung Min Kong, Dong-Hun Seo, Won Don Lee, 2007. Rule Refinement with Extended Data Expression, *IEEE Computer Society*, Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA), 2007,pp. 310-315.

[8] R. Polikar, 2007. Bootstrap-Inspired Techniques in Computational Intelligence. *IEEE Signal Processing Magazine*, July,2007, pp.59-72.

[9] UCI Repository of Machine Learning Databases, http://archive.ics.uci.edu/ml