ISSN : 2165-4069(Online)
ISSN : 2165-4050(Print)

# IJARAI

## International Journal of Advanced Research in Artificial Intelligence

Volume 5 Issue 9

www.ijarai.thesai.org

A Publication of
The Science and Information Organization

# Editorial Preface

*From the Desk of Managing Editor…*

Artificial Intelligence is hardly a new idea. Human likenesses, with the ability to act as human, dates back to Geek mythology with Pygmalion's ivory statue or the bronze robot of Hephaestus. However, with innovations in the technological world, AI is undergoing a renaissance that is giving way to new channels of creativity.

The study and pursuit of creating artificial intelligence is more than designing a system that can beat grand masters at chess or win endless rounds of Jeopardy!. Instead, the journey of discovery has more real-life applications than could be expected. While it may seem like it is out of a science fiction novel, work in the field of AI can be used to perfect face recognition software or be used to design a fully functioning neural network.

At the International Journal of Advanced Research in Artificial Intelligence, we strive to disseminate proposals for new ways of looking at problems related to AI. This includes being able to provide demonstrations of effectiveness in this field. We also look for papers that have real-life applications complete with descriptions of scenarios, solutions, and in-depth evaluations of the techniques being utilized.

Our mission is to be one of the most respected publications in the field and engage in the ubiquitous spread of knowledge with effectiveness to a wide audience. It is why all of articles are open access and available view at any time.

IJARAI strives to include articles of both research and innovative applications of AI from all over the world. It is our goal to bring together researchers, professors, and students to share ideas, problems, and solution relating to artificial intelligence and application with its convergence strategies. We would like to express our gratitude to all authors, whose research results have been published in our journal, as well as our referees for their in-depth evaluations.

We hope that this journal will inspire and educate. For those who may be enticed to submit papers, thank you for sharing your wisdom.

**Editor-in-Chief**
**IJARAI**
**Volume 5 Issue 9 September 2016**
**ISSN: 2165-4069(Online)**
**ISSN: 2165-4050(Print)**

# Editorial Board

# Reviewer Board Members

# CONTENTS

# A Rank Aggregation Algorithm for Ensemble of Multiple Feature Selection Techniques in Credit Risk Evaluation

Shashi Dahiya
Deptt. Of Computer Science and Engineering
Manav Rachna International University, (MRIU)
Faridabad, India

S.S Handa
Deptt. Of Computer Science and Engineering
Manav Rachna International University, (MRIU)
Faridabad, India

N.P Singh
Management Development Institute, (MDI)
Gurgaon, India

***Abstract*—In credit risk evaluation the accuracy of a classifier is very significant for classifying the high-risk loan applicants correctly. Feature selection is one way of improving the accuracy of a classifier. It provides the classifier with important and relevant features for model development. This study uses the ensemble of multiple feature ranking techniques for feature selection of credit data. It uses five individual rank based feature selection methods. It proposes a novel rank aggregation algorithm for combining the ranks of the individual feature selection methods of the ensemble. This algorithm uses the rank order along with the rank score of the features in the ranked list of each feature selection method for rank aggregation. The ensemble of multiple feature selection techniques uses the novel rank aggregation algorithm and selects the relevant features using the 80%, 60%, 40% and 20% thresholds from the top of the aggregated ranked list for building the C4.5, MLP, C4.5 based Bagging and MLP based Bagging models. It was observed that the performance of models using the ensemble of multiple feature selection techniques is better than the performance of 5 individual rank based feature selection methods. The average performance of all the models was observed as best for the ensemble of feature selection techniques at 60% threshold. Also, the bagging based models outperformed the individual models most significantly for the 60% threshold. This increase in performance is more significant from the fact that the number of features were reduced by 40% for building the highest performing models. This reduces the data dimensions and hence the overall data size phenomenally for model building. The use of the ensemble of feature selection techniques using the novel aggregation algorithm provided more accurate models which are simpler, faster and easy to interpret.**

***Keywords—Classification; Credit Risk; Feature Selection; Ensemble; Rank Aggregation; Bagging***

## I. INTRODUCTION

The data size is increasing regarding records and dimensions both. It presents challenges to the machine learning community which is working on new methods and techniques to fasten the data exploration, analysis, and validation tasks. One way of handling this problem is by using an effective sampling methodology to choose a subset of samples describing the dataset as a whole. This method results in a reduced dataset having less number of instances. Another way of handling this problem is to use an appropriate dimensionality reduction/ feature selection method to reduce the dimensions of the dataset.

In a vital machine learning problem of classification, the accuracy of a classifier plays an important role. The accuracy of the classifier depends on many factors such as – the single, hybrid or an ensemble method used for modelling; the base models used for the ensemble; the learning algorithm used for model training; the feature selection method used for selecting the relevant features; the sampling technique used for sampling the data; the evaluation method used for testing the model and many more.

Feature selection is an important pre-processing step in machine learning and pattern recognition problems. It has been an active area of research since past three decades [1]. Feature selection increases the performance of classification models by eliminating redundant and irrelevant features and thus reducing the dimensionality of datasets [2]. This study uses the feature selection approach for the enhancement of accuracy of credit risk evaluation models.

### *A. Credit Risk Evaluation*

Quantifying the credit risk is a typical bank decision problem of classification in which the new loan applicants are to be classified accurately into either a creditworthy or a non-creditworthy category based on the historical dataset of loan applicants. This historical dataset is used for training the classifier, and the new loan applicant's data is tested on this trained classifier. The Class labels i.e. creditworthy or non-creditworthy are automatically assigned to the new applicants records during testing phase. The credit dataset contains the features mainly describing the financial status, demographic details of the applicant and his personal profile. Some features of the dataset may provide more significant information needed for classifying a new loan applicant than others. While some of the features are not required, some may contain redundant or irrelevant information and don't provide any additional information during the model development task. They don't contribute to the accuracy of the model and sometimes even decrease it by slowing down the classifier learning process. The big feature set can make a more complex model whose interpretation also becomes

cumbersome. It can make a classifier overfitting the training data [3].

### B. Feature Selection for Credit Evaluation

In credit risk evaluation the accuracy of the classifier is very crucial. Even a small increase in model accuracy may result in huge profit for the bank. For performance enhancement of single models, the literature proposed the hybrid and ensemble based models. In credit risk evaluation. Many of the ensemble based and hybrid models are developed using feature selection methods during the initial stage [4]. Feature selection is crucial for the selection of significant and appropriate features for model development. If the number of features is large, more computation is required, and the accuracy and interpretation of the classification model decrease [5], [6]. A large number of features in credit evaluation implies that there are a large number of questions for the loan applicants, which will be time-consuming and confusing. According to [7], exploring a big number of features lead to identifying a relevant subset of features for building the credit model.

The relevance of the features needs to be identified before the model development task so that the undesired, redundant and irrelevant features are not used as input to the model. Supervised feature selection determines relevant features by their relations with the corresponding class labels and discards irrelevant and redundant features. The subset of features identified as important will help in reducing the size of the hypothesis space and allows the algorithms to operate faster and more effectively [8]. This smaller feature subset will help in building simplified models reducing the time and space complexity of the algorithms and hence improving the accuracy with well interpreted results.

The purpose of this paper is the enhancement of classification accuracy of the credit risk evaluation models. This study uses the ensemble of multiple feature selection techniques for ranking and selecting the significant features.

## II. Feature Selection Criteria for Filter based Feature Selection

The filter approach to feature selection works independently of learning/Induction algorithm (Fig. 1.). It operates as a pre-processing step and selects and presents the important features to the learning algorithm as input. Filter approach makes use of the complete training data for its operation. It ranks the features in accordance with their importance w.r.t selecting a class. A threshold has to be then defined for selecting the number of most important features from the ranking.


Fig. 1. The Filter based method of Feature Selection

There are several features ranking methods [9] available in the literature, some of them are - correlation based, mutual information based and methods based on decision tree and the distance between probability distributions. Any of the predefined measures such as – the Dependency measures, Information measures, distance measures [10] [11], independent component analysis [12], class separability measure [13], or variable ranking [14] are the basis of these feature ranking methods.

### A. Dependency measures

As discussed by [15] and [2], the dependency measures or correlation measures quantify the ability to predict the value of one variable based on the value of the other. The Pearson's correlation coefficient (PCC) is very useful for feature selection [16] [17], as it quantifies the relationship of a feature with its corresponding class label and with other features in the dataset. As per [18], PCC for continuous features is a simple measure but can be effective in a wide variety of feature selection methods.

A uniform manner is used to treat the features and the class, then the feature-class correlation and feature-feature inter-correlations are calculated according to the following equation:

$$CC(X_j, c) = \frac{\left[\sum_{i=1}^{m}(x_i^j - \bar{X}^j)(c_i - \bar{c})\right]}{\sigma_{X^j} \cdot \sigma_c}$$

$\bar{X}^j$ and $\sigma_{X^j}$ are the mean and standard deviation of $j^{th}$ feature and $\bar{c}$ and $\sigma_c$ are the mean and standard deviation of vector c of class labels). The ranking values are absolute values of CC:

$$J_{CC}(Xj) = |\, CC(Xj, c)\,|$$

This ranking has a low complexity of the order of *O (mn)* and is very simple to implement for numerical variables.

For, nominal or categorical variables the popular feature selection method used is Pearson's chi-squared ($\chi$ 2) test. The numerical variables can also be converted into nominal or categorical types for applying the $\chi$ 2 test. First, a contingency table is made by converting the raw data. Then, the independence between each variable and the target variable is measured using the contingency table. $\chi^2$ is defined by :

$$\chi^2 = \sum_{i=1}^{c} \frac{(O_i - E_i)^2}{E_i}$$

where $O_i$ is the observed frequency; $E_i$ is the expected theoretical frequency, asserted by the hypothesis of independency and c the number of cells in the contingency table.

Correlation-based feature selection is the base for symmetrical uncertainty (SU) also. It is a symmetric measure and can be used to measure feature-feature correlation. The value of symmetrical uncertainty ranges between 0 and 1. The value of 1 indicates that one variable (either X or Y) completely predicts the other variable [19] .The value of 0 indicates that both variables are completely independent.

### B. Information Measures

Information theory has been proved to be very successful in solving many problems [20]. It provides a theoretical

framework for measuring the relation between the classes and a feature or more than one feature. Mutual Information (MI) is a filter-based feature selection metric used to find the relevance of features. It works on the principle of information shared by two features using MI [20], the relevance of a feature subset on the output vector C can be quantified. Formally, the MI is defined as follows:

$$I(x; y) = \sum_{i=1}^{n}\sum_{j=1}^{n} p(x(i), y(j)).\log\left(\frac{p(x(i),y(j))}{p(x(i)) \cdot p(y(j))}\right)$$

Where MI is zero when x and y are statistically independent, i.e., p(x(i), y(j)) = p(x(i))·p(y(j)).

Large values of MI indicate a high correlation between the two features and zero indicates that two features are uncorrelated. Many feature selection methods are proposed based on MI such as [20] [21].

Information Gain (IG) and Gain Ratio (GR) are feature ranking methods based on information measures. IG is the reduction in entropy of the class variable when the value of the independent variable is known. The IG of an attribute X with respect to class variable Y is given by:

$$I(Y; X) = H(Y) - H(Y|X)$$

Where *H(Y)* is the entropy of Y,

*H (Y|X)* is the uncertainty about Y for a given X

The information gain measure is biased towards tests with many outcomes. Therefore C4.5 uses Gain Ratio (GR) for overcoming this bias and is an extension of IG.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

Where Gain(A) is the encoding information gained by branching on A and SplitInfo(A) is the information got by splitting the dataset into 'n' distinct values of the attribute A. The maximum GainRatio attribute is subject to splitting.

### C. Distance Measures

Distance measures, also known as separability, divergence, or discrimination measures, study the difference between the two-class conditional probabilities in a binary context [15] [22]. In other words, a feature $X_j$ is chosen over another feature $X_j$' if it induces a greater difference between the two-class conditional probabilities than $X_j$'. In the case where the difference is zero then the two features are identical. Relief is one of the most famous feature selection method based on distance measures. Relief algorithm has been given by [23]. It is a multivariate method which is sensitive to interactions [24]. It estimates the features relevance according to how well their values distinguish between the instances of the same and different classes that are near each other. It performs well on small sample size datasets having a large collection of features. Its computational complexity is O (mn), which is linear in comparison to other multivariate methods often having quadratic complexity in the number of features.

### D. Feature Ranking

Feature ranking uses the above discussed filter based measures to compute a scoring function from the values ($x^j_i$ ; $y_i$). It is considered that a high score indicates a valuable feature and the features are sorted in decreasing order of the scoring function [25]. It is computationally efficient since it requires only the computation of d scores and sorting them. It is statistically robust against overfitting because it introduces bias, however it may have considerably less variance [26]. Therefore, feature ranking can be preferable than any other feature selection method.

## III. Background

In general the feature ranking criteria for filter based feature selection discussed above have one or the other limitation in their performance. The distance based measures like - Relief are good in capturing the relevance of features to the target variable but doesn't capture the redundancy among the features. The dependency measure such as PCC is not able to capture the correlations that are not linear [2]. The dependency measures and information measures suffer from time complexity issues since they have to evaluate all possible subsets. Therefore they are not practical to deal with high dimensional data.

Due to these limitations of the filter based methods, it is difficult to find out the best criteria for a particular problem.

According to [27] this problem is called the selection trouble. The best approach is to independently apply a combination of the available methods and evaluate the results.

Aggregating the ranked lists from individual rankers into a single better ranking is called as rank aggregation. Rank aggregation method is an Ensemble based feature selection method which is considered as an upcoming important tool for combining information with the purpose of getting higher accuracy.

## IV. Ensemble Method for Feature Selection

An ensemble of classifiers is a set of base Classifiers that are individually trained. For classifying new instances, the decisions of these classifiers are combined using weighted or un-weighted majority voting [28] [29]. According to [30], the ensemble model could outperform the single base models when weak/ unstable models are combined. Looking at advantage of ensemble based classifiers over individual ones, the concept of ensemble can be applied for performance enhancement in the feature selection process also.

### A. Ensemble of a Single Feature Ranking Technique

Ensemble of a single feature ranking technique involves Bagging (Bootstrap Aggregation) or some other Algorithms to generate various bags of data. For each bag the feature ranking is done and the ensemble is formed by combining the individual bag rankings by weighted voting, using linear aggregation [31].

### B. Ensemble of Multiple Feature Ranking Technique

In this method, multiple feature ranking techniques are used for ranking the features in order of their relevance for building an ensemble. The same training data is used by the ranking methods and the results of these methods i.e. the ranking lists are combined in a certain way to obtain a final

ranked list of the features. Thus, multiple feature ranking lists creates a single feature ranking list in the following two steps: First a set of different ranking lists are created using corresponding rankers and secondly these ranking lists are combined using rank ordering of features [32].

Suppose a dataset 'D' has 'I' instances and 'k' features. During the first step a set of n ranking lists {F1, F2, F3…Fn} are obtained (one for each 'n' feature selection methods used).

In the second step, a rank aggregation method R is used for combining the ranks of individual features from n ranking lists obtained in first step. Let $f_i^j$ be the rank of feature i from ranking list j, then the set of rankings of feature i is given by:

$R_i = f_i^1, f_i^2, f_i^3, \ldots\ldots, f_i^n$

The new rank obtained by feature i using the combination method C is

$\breve{F}_i = R\ (f_i^1, f_i^2, f_i^3, \ldots\ldots, f_i^n)$

### C. Rank Aggregation

There are different combination or rank aggregation methods used for creating an aggregated feature ranking list from various individual feature ranking lists for the ensembles of multiple feature selection techniques. Recently, there have been studies applying the ensemble concept to the process of feature selection [33]. The results of this technique are more stable and accurate as the different ranking methods explore different important qualities of the data. A combination of these qualities in one ranking scheme will outperform each ranking method.

Research in the field of feature selection proposed some rank aggregation methods such as the sum, mean, median, highest rank or lowest rank aggregation and some are more difficult [33]. Moreover, research is on to give more weight to top ranking features or combining well-known aggregation methods in search of finding the best list which is an optimization problem.

## V. Methodology

In this paper, the ensemble of multiple feature selection methods has been used for the selection of important features for the classifier. For the combination of ranks of individual feature selection methods the ensemble uses the fusion based rank aggregation method. For, the FS ensemble, five individual filter based methods of FS were chosen based on different measures of feature ranking. These were – Chi Square and Symmetrical Uncertainty methods of FS based on Dependency Measures; Information Gain and Gain Ratio FS methods based on Information Theory Measures; and Relief FS method based on Distance Measures.

In the first step, the five filter-based feature selection methods were used for ranking the features by their importance. The result of the first step is five ranked lists from the five individual feature selection methods.

The results of the first step are five ranked lists from the five individual feature selection methods.

The individual feature selection methods used are the Chi-Square, Information Gain, Gain Ratio, ReliefAttributeEval and SymmetricalUncertaintyAttributeEval from the WEKA software environment for knowledge analysis [34]. The study conducts experiments for ranking features using each feature selection method.

The second step proposes a new fusion based Rank Aggregation Algorithm for an ensemble of multiple feature selection techniques. The algorithm is described in Fig. 2. This method makes use of both rank score and ranks order of each feature in the ranked lists for rank aggregation. Fig. 2. describes the rank aggregation algorithm and its operation as follows:

First, the **k** individual feature selection methods rank the **n** features in order of their importance in descending order. Hence, each feature selection method generates a ranked list depicting the rank score (the value of a feature in the ranked list) and a sequence number **m** of each feature in the descending ordered ranked list.

In the second step, most of the rank aggregation methods use a combination of the ranked scores of multiple feature selection techniques in a certain way such as the sum, mean, median or taking the highest or lowest rank scores. But the rank score alone can't depict the importance of a feature in the ranked list. The order of the feature in the ranked list is also crucial for considering the importance of a feature. The proposed novel aggregation algorithm considers both rank scores and the rank orders of the features. This aggregation will give more weight to the features which not only have higher rank scores but also have higher rank orders in the ranked list. Equation (1) computes the rank order of a feature having sequence no. 'm' in a ranked list of 'n' features. Therefore, for a feature having sequence number 1, in a ranked list of 20 features, the aggregation finds the rank order of this feature as 20 by using (1).

$$rankorder = n - m + 1 \quad (1)$$

**Algorithm: A Novel Rank Aggregation Algorithm for Ensemble of Multiple Feature Selection Techniques**

---

**Input:**
Dataset m*n containing m instances and n features $f_j$, where j = 1, 2, -----, n

Initialize Ensemble Rank List E = φ
Suppose $F_1$, $F_2$, -------, $F_k$ be the feature selection techniques used for the ensemble
For each $F_i$, i= 1, 2, ---, k
Calculate rank score of each feature and construct ranked lists $R_i$, i = 1, 2, ----, k
Sort each $R_i$ in descending order of rank scores
Give a sequence number m=1, 2, ------, n; to all the features in each $R_i$ starting from top.
ENDFOR

For each feature $f_j$, j = 1, 2, -----, n
For each sorted ranked list $R_i$, i = 1, 2, ------, k
For Sequence no. m = 1, 2,-------,n;
*rankorder* = n – m +1

Ensemble *rankscore* $E_j = \sum_{i=1}^{k} (rankscore_{ji} * rankorder_{ji})$
E = E U $E_j$
ENDFOR
ENDFOR
ENDFOR
Sort the Ensemble rank list E using ensemble rank scores in descending order

**Output:** A sorted ensemble ranked list E containing features and their corresponding ensemble rank scores.

Fig. 2. A Novel Rank Aggregation Algorithm for Ensemble of FS

## VI. EXPERIMENTS

### A. Data Used

The data set chosen for this experiment is the German dataset from UCI repository [35]. It is a credit dataset having 1000 loan applicants' records and 20 predictor variables. There is one class variable having two classes - Good and Bad. Most of the features are qualitative, and few are numerical.

### B. Feature Selection

For ranking the features in order of their importance, the experiments considers the ensemble of multiple feature ranking techniques and five individual rank based feature selection methods. Those feature selection methods are used which perform better on qualitative data since the data is mostly qualitative. The novel rank aggregation algorithm uses the rank scores and rank orders of the individual rank based feature selection methods. The threshold values of 80%, 60%, 40% and 20% i.e. 16, 12, 8 and four features are used for selecting the features from the top of the sorted, ranked lists. In this way, only the highly ranked features identified as important and relevant by the individual and ensemble feature selection methods have been selected for building the classification models.

The performance of the classifiers is compared to find out the best threshold, best model and the best feature selection method which yielded the highest ROC value. The best threshold value indicates that the features selected using it are the most important ones which best described the dataset.

The best model is the one whose average classification performance across all the feature selection methods is the highest. The best feature selection method is the one which yields best average performance across all models built over the features selected by it.

### C. Classifier

For testing the impact of the new rank aggregation algorithm on the accuracy of classifiers, the features selected from the aggregated ranked list are taken as inputs to the classifiers. The individual and ensemble based classifiers are used for model building and performance assessment. The individual classifiers used are the C4.5 and the MLP, while Bagging is used as the ensemble classifier.

The ensemble based bagging technique is used since the use of bootstrapping with replacement in bagging creates diversity within the data being used by the classifier hence impacting the performance of the classifier. The base classifiers used for bagging are the C4.5 and the MLP. These classifiers are considered acceptable to use at the cost of time

and complexity of the system, since the focus of the study is the enhancement of classification accuracy of the credit risk evaluation models using the proposed rank aggregation method. For data sampling using bootstrapping, 20 iterations are used, as the classifier didn't show any increase in performance using more iterations. More iterations would rather have slowed down the classification process by increasing data samples and hence time.

### *D. Accuracy Assessment*

The Area under the Receivers Operating Curve (ROC) popularly known as AUC, is used for accuracy assessment. The ROC Curve is a graph of True Positive Rate (TPR) versus False Positive Rate (FPR). The models are built using 70% training and 30% test partitions. A random sampling of 70% of training data is done from the dataset for training the classifier. The classifier uses the remaining 30% of data for testing the classifiers. The correctly classified instances were taken from the test data for classification. A ROC graph was plotted using TPR against the FPR for assessing the accuracy.

## VII. Results and Discussion

Four classifier models - C4.5, MLP, C4.5 based Bagging and MLP based Bagging were built on the German credit dataset using a different number of features selected by each FS method. Each model was generated on four different threshold percentages (80%, 60%, 40% and 20%) i.e. (16, 12, 8 and 4) features selected from the sorted, ranked lists of the five individual feature selection methods and an ensemble of multiple FS methods. The performance of the classifiers has been observed using the ROC measure which is considered as a true measure of accuracy. For comparison of accuracy, each model has also been built using all the features. The average performance of six FS methods using four different thresholds across four different classifier models is depicted in Table I.

TABLE I. Average Performance of Rank based Feature Selection Methods

| FS Ranking Methods | 80% | 60% | 40% | 20% | Ranking Methods Average Performance |
|---|---|---|---|---|---|
| **Chi-square** | .766 | .754 | .747 | .740 | **0.752** |
| **Gain Ratio** | .766 | .771 | .734 | .733 | **0.748** |
| **Information Gain** | .766 | .754 | .747 | .740 | **0.752** |
| **Relief-F** | .762 | .766 | .729 | .734 | **0.748** |
| **Symmetrical Uncertainty** | .766 | .759 | .741 | .748 | **0.753** |
| **Ensemble** | .769 | .772 | .741 | .740 | **0.756** |

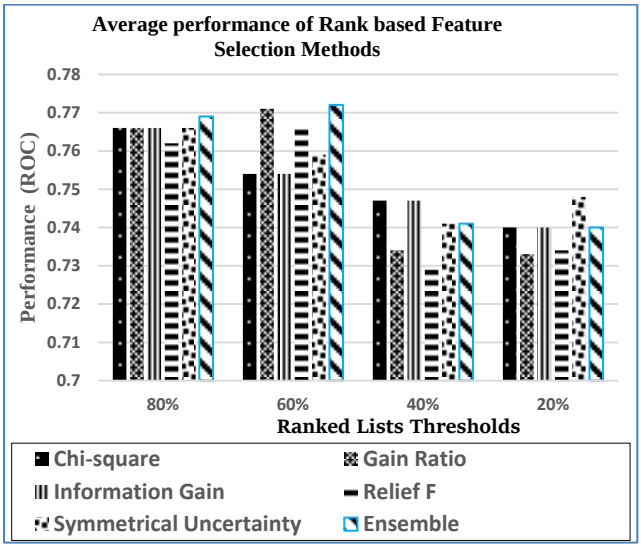



Fig. 3. Average performance of Rank based Feature Selection Methods across all models

The performance of each FS based ranking method is recorded for the four models for all thresholds. An average of performances of all the models on the features selected by the FS methods using a particular threshold is observed. Similarly, the average performances of all FS methods including Ensemble FS method have been calculated across all models using different thresholds. The comparative performance of these FS methods is depicted in Fig. 3.

The graph of Table I. summarizes that the performance of the ensemble of multiple FS methods is higher than all individual FS methods for the thresholds of 80% and 60%, while the performance of FS methods Chi-square and Information gain is higher than others for the 40% threshold. The symmetrical uncertainty method outperforms the others for 20% threshold. It is clearly observed from the graph that, for 40% and 20% thresholds (i.e. small no. of features), the performance of all the FS methods is substantially lower than that for 80% and 60% thresholds.

By looking at the graph, it can also be inferred that the performance of the Ensemble of FS methods is the highest for the 60% threshold followed by 80% threshold. Also, the performance of all FS methods including the ensemble of multiple FS techniques started declining drastically after the 60% threshold.

The individual model performance based on different thresholds using the ensemble of FS method is depicted in Table II. It can be seen across all thresholds, the performance of bagging models based on C4.5 and MLP as the base classifiers is much better than the individual C4.5 and MLP models. Moreover, the average model performance for the bagging model based on MLP as the base classifier is the best. It can also be observed that the average performance of all the models is the best for 60% threshold. The graph depicting the average performance of the individual models in Fig. 4, shows that the performance of bagging based on MLP classifier is the highest followed by bagging based on C4.5 classifier at 60% threshold. While the individual models C4.5 and MLP performed best at 80% threshold, the individual C4.5 model performed the worst of all for all thresholds.

TABLE II. INDIVIDUAL MODEL PERFORMANCE ON DIFFERENT THRESHOLDS USING THE ENSEMBLE BASED ON FS METHOD

| Classification Models | 100% | 80% | 60% | 40% | 20% | Avg. Model Performance |
|---|---|---|---|---|---|---|
| **C4.5** | .730 | .730 | .727 | .707 | .726 | **0.728** |
| **Bagging (C4.5)** | .773 | .773 | .787 | .774 | .752 | **0.775** |
| **MLP** | .725 | .770 | .765 | .711 | .716 | **0.743** |
| **Bagging (MLP)** | .788 | .801 | .809 | .773 | .767 | **0.794** |
| **Avg. performance** | **.754** | **0.769** | **0.772** | **0.741** | **0.740** | |

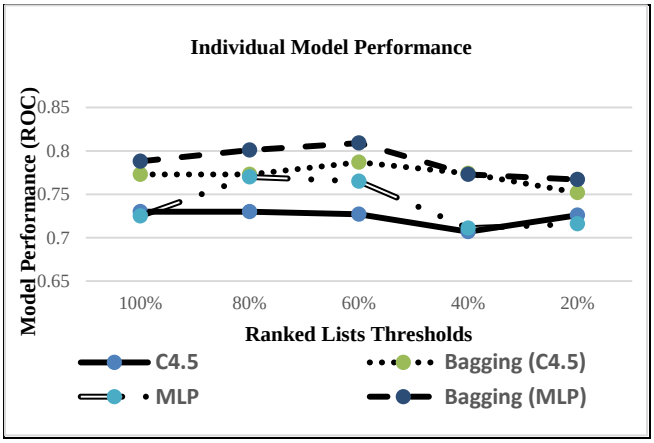



Fig. 4. Average performance of Individual Models using Ensemble based FS method

## VIII. CONCLUSION

In credit risk evaluation the accuracy of a classifier is very crucial. Even a small increase in model accuracy may result in huge profit for the bank. For accuracy enhancement, this study uses the ensemble of multiple feature selection techniques for ranking and selecting the important features. A novel rank aggregation algorithm has been proposed using the rank scores and rank orders of the individual rank based feature selection methods. The ensemble of FS technique uses the novel rank aggregation algorithm for ranking the features in order of their importance and relevance. The ranked lists of 5 FS methods and 1 Ensemble based FS method were used to select the top 16, 12, 8 and 4 features. The Ensemble based FS method attained the best performance for the threshold of 12 top features with an average ROC value of .772 followed by the threshold of 16 giving an average ROC value of .769 while the average ROC value for the dataset without FS is .754. Moreover, these ROC values for the ensemble method are higher than all other individual FS methods used. On comparing the ROC values it is inferred that using the Ensemble based FS method, the average performance of the four models increased by a ROC of .018 using the 60% threshold.

The results also concluded that the bagging based models outperformed the individual models using the ensemble of FS methods for all thresholds. The performance of Bagging using MLP as the base classifier is the highest with a ROC of .809 followed by Bagging using C4.5 as the base classifier with a ROC of .787 at 60% threshold, while the individual MLP and C4.5 models performed with an ROC value of .765 and .727 respectively for the same threshold. By using Bagging, there is an average performance enhancement of .044 and .060 respectively for individual MLP and C4.5 models across all thresholds. One more inference drawn from the results is that the average performance of Bagging model with MLP as the base classifier is the best across all thresholds with a ROC of .794 followed by .775 for the Bagging model with C4.5 as the base classifier.

Therefore, the study concluded that, using an ensemble of multiple feature selection techniques with the novel rank aggregation algorithm proposed in the study, a significant enhancement in the performance of credit risk evaluation models is observed. The accuracy of the models is enhanced with the selection of top 80% and 60% features from the ranked list of the ensemble. Although, the accuracy of the models declined with the selection of top 40% and 20% features. It may be attributed to the rejection of many relevant features required for building the accurate model.

By using the ensemble of multiple feature selection techniques, the bagging based models outperformed the individual models for all thresholds but most significantly for the 60% threshold.

This increase in performance is more significant from the fact that the number of features reduces by 40% for building the highest performing models which indicates a phenomenal reduction in the instance size and hence the overall data size. The reduction of irrelevant features simplifies the model building task and hence the time and space complexity of running the models. A simpler and faster model would be helpful for the bankers in a quick and precise overall assessment of the risk involved in granting the loan to a customer. Moreover, the irrelevant features with very low ranks are identified which do not contribute to the model building process. These features can be ignored by the banks in the loan application forms, making them simpler and faster for the applicants to fill in and for the banks to get them verified quickly.

Future studies can focus on testing the novel rank aggregation algorithm on other high dimensional credit datasets collected from the real world. The algorithm may

prove to be more useful for such data with a large number of attributes by selecting only a small number of relevant attributes contributing to the accuracy and simplicity of the model. Even a small enhancement in the accuracy of credit risk evaluation models is very beneficial as the financial risk associated with the credit defaulters get assessed accurately on time.

## REFERENCES

[1] H. Liu, H. Motoda, R. Setiono, Z. Zhao, "Feature Selection: An Ever Evolving Frontier in Data Mining", JMLR: Workshop and Conference Proceedings, vol. 4, Publisher: Citeseer, pp. 4-13, 2010.

[2] L. Yu, H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation- Based Filter Solution", Proceedings of the Twentieth International Conference on Machine Leaning, ICML-03, Washington, D.C., August, 2003, pages 856-863.

[3] M. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," IEEE Transactions on Knowledge and Data Engineering, vol. 15(3), November/December 2003.

[4] Dahiya, S., Handa, S.S., Singh, N.P. (2015). Credit Evaluation using Ensemble of various classifiers on reduced feature set", Industrija, Vol.43, No.4.pp 163-174.

[5] Y. Liu, M. Schumann, "Data mining feature selection for credit scoring models," Journal of Operations Research Society, vol. 56(9), pp. 1099–1108, 2005.

[6] T. Howley, M. G. Madden, M. L. O'Connell, and A. G. Ryder, "The effect of principal component analysis on machine learning accuracy with high dimensional spectral data," Knowledge Based Systems, vol. 19 (5), pp. 363-370, 2006.

[7] Hand, D. J., Henley, W. E. (1997).Statistical Classification Methods in Consumer Credit Scoring: A Review. Journal of the Royal Statistical Society: Series A (Statistics in Society), Vol. 160, No. 3, pp. 523-541.

[8] M. A. Hall, "Feature Selection for Discrete and Numeric Class Machine Learning," In Proceedings of the 17th international conference on Machine Learning (ICML-2000).

[9] W. Duch, "Filter methods," Feature extraction, foundations and Applications, pp. 89–117. Studies in fuzziness and soft computing, Springer (2006).

[10] T. W. S. Chow, D. Huang, "Using Mutual information for Feature selection with bioinformatics applications," Neural Networks Applications in Information Technology and Web Engineering, 2005, Borneo Publishing Co.

[11] V. Sindhwani, S. Rakshit, D. Deodhare, D. Erdogmus, P J, Niyogi, "Feature selection in MLPs and SVMs based on maximum output information". IEEE Trans Neural Netw ork, vol.15(4), pp.937–948, 2004.

[12] MD Plumbley, E Oja, "A nonnegative PCA" algorithm for independent component analysis," IEEE Transactions on Neural Networks vol. 15 (1), pp. 66-76, 2004.

[13] K.Z. Mao, "Orthogonal forward selection and backward elimination algorithms for feature subset selection," IEEE Trans Syst Man Cybern B Cybern, vol. 34(1), pp.629–634, 2004.

[14] R. Caruana, De Sa V , "Benefitting from the variables that variable selection discards," Journal of Machine Learning, Res 3, pp. 1245–1264, 2003.

[15] M. Dash and H. Liu, "Consistency-based search in feature selection," Artificial Intelligence, vol. 151 (1-2), pp. 155-176, 2003.

[16] K. Grabczewski, N. Jankowski, "Mining for complex models comprising feature selection and classification," Feature extraction, foundations and Applications, pp. 473–489, Studies in fuzziness and soft computing, Springer, 2006.

[17] Guyon, A. Elisseef, "An introduction to variable and feature selection," Journal of Machine Learning Research pp. 1157–1182, (2003).

[18] Rodriguez-Lujan, R. Huerta, C. Elkan and C.S. Cruz, "Quadratic programming feature selection". Journal of Machine Learning Research, vol. 11, pp. 1491–1516, 2010.

[19] Ienco, R.G. Pensa, R. Meo, "Context-based Distance Learning for Categorical Data clustering", IDA 2009, LNCS 5772, Springer, Berlin, pp. 83 – 94, 2009.

[20] Kumar, and K. Kumar, "A novel evaluation function for feature selection based upon information theory", In Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 395–399, 2011.

[21] Al-Ani, A. and M. Deriche, "An optimal feature selection technique using the concept of mutual information," In Proceedings of the Sixth International Symposium on Signal Processing and its Applications, pp. 477–480, 2001.

[22] Liu and L. Yu, "Towards integrating feature selection algorithms for classification and clustering," IEEE Transactions on Knowledge and Data Engineering, vol.17 (4), pp. 491-502, 2005.

[23] K. Kira, and L. Rendell, "A Practical Approach to Feature Selection," Proceedings of the Ninth International Workshop on Machine Learning, pp.249-256, 1992.

[24] Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF," In: Proceedings of the 7th European Conference on Machine Learning (ECML-94), pp. 171–182, New York: Springer, 1994.

[25] W. Bouaguel, "On Feature Selection Methods for Credit Scoring" Doctoral Thesis, ISG University of Tunis, 2015.

[26] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," Springer series in statistics, Springer New York Inc. 2001.

[27] O. Wu, H. Zuo, W. Zhu, M. Hu, J. Gao, and H. Wang, "Rank aggregation based text feature selection," In: Proceedings of the Web Intelligence, pp. 165-172, 2009.

[28] J. Kittler, "Combining classifiers: A theoretical framework," Pattern Analysis & Applications, vol. 1 (1), pp. 18-27, 1998.

[29] L.I. Kuncheva, "Combining Pattern Classifiers: Methods and Algorithms,". John Wiley & Sons, (2004).

[30] T.G. Dietterich, "Ensemble methods in machine learning," In Proceedings of the First International Workshop on Multiple Classifier Systems, London, UK, pp. 1-15. Springer-Verlag, 2000.

[31] Y. Saeys, T. Abeel, Y. V. Peer, "Robust feature selection using ensemble feature selection techniques", ECML PKDD 2008, Part II, LNAI 5212, pp. 313–325.

[32] S.H. Vege, "Ensemble of Feature Selection Techniques for High Dimensional Data". Master's Thesis, Western Kentucky University, 2012.

[33] Dittman, D. J., T. M. Khoshgoftaar, R. Wald, and A. Napolitano (2013). Classification performance of rank aggregation techniques for ensemble gene selection. In C. Boonthum-Denecke and G. M. Youngblood (Eds.), Proceedings of the International Conference of the Florida Arti_cial Intelligence Research Society (FLAIRS). AAAI Press.

[34] http://www.cs.waikato.ac.nz/, accessed on 6 June'2016.

[35] https://archive.ics.uci.edu/ml/datasets/ accessed on 3, June'2016.

# Pursuit Reinforcement Competitive Learning: PRCL based Online Clustering with Tracking Algorithm and its Application to Image Retrieval

Kohei Arai[1]
[1]Graduate School of Science and Engineering
Saga University
Saga City, Japan

***Abstract*—Pursuit Reinforcement guided Competitive Learning: PRCL based on relatively fast online clustering that allows grouping the data in concern into several clusters when the number of data and distribution of data are varied of reinforcement guided competitive learning is proposed. One of applications of the proposed method is image portion retrievals from the relatively large scale of the images such as Earth observation satellite images. It is found that the proposed method shows relatively fast on the retrievals in comparison to the other existing conventional online clustering such as Vector Quatization: VQ. Moreover, the proposed method shows much faster than the others for the multi-stage retrievals of image portion as well as scale estimation.**

***Keywords*—Pursuit Reinforcement Guided Competitive Learning; Reinforcement Guided Competitive Learning; Sustained Reinforcement Guided Competitive Learning Vector Quantization; Learning Automata***

## I. INTRODUCTION

Clustering is an exploratory data analysis tool that deals with the task of grouping objects that are similar to each other [1,2,3]. For many years, many clustering algorithms have been proposed and widely used. It is commonly used in many fields, such as data mining, pattern recognition, image classification, biological sciences, marketing, city-planning, document retrieval, etc.

Many cases of clustering commonly used the static data. It means that the clustering can be made after the entire data have been collected, then grouped into clusters whose members are similar in some way. In the data mining, there is a kind of data which comes every time so that we cannot stop it in a while in order to make clustering.

Online clustering is a kind of clustering that is used for dynamic data. It is not considering a number of data, but only focus on a new data and previous centroids. However, determining position of each centroid because of a new data attracted some approaches. Vector Quantization (VQ) was a very simple approach to do online clustering. It is derived from concept of competitive learning network [4],[5]. Likas (1999) proposed Reinforcement Guided Competitive Learning (RGCL) [6] as an approach for on-line clustering based on reinforcement learning. It utilized the concept of reward in the reinforcement learning from winning unit in the Learning Vector Quantization. The Sustained RGCL (SRGCL) was modification of RGCL in considering a sustained exploration in reinforcement learning. On the other hand, other approaches such as modified ISODATA, k-means clustering, Self-Organization Mapping: SOM based clustering, spatial feature utilizing clustering, Fisher distance measure utilizing clustering, GA based clustering and so on are proposed in order to improve clustering performance [7]-[25].

A new approach for online clustering based on reinforcement learning, called Pursuit Reinforcement Guided Competitive Learning. PRCL which is derived from pursuit method in reinforcement learning that maintain both action-value and action preferences, with the preferences continually pursuing the action that is greedy according to the current action-value estimates together with learning automata is proposed. PRCL can be used as online clustering method. One of the applications is, then introduced for evacuation simulation.

The following section describes the proposed PRCL with learning automata together with the existing conventional online clustering methods of RGCL, SRGCL and VQ. Then preliminary experiments are described followed by its application of image retrievals. After all, conclusion is described with some discussions.

## II. THEORETICAL BACKGROUD

### A. Reinforcement Learning

Reinforcement Learning is learning what to do---how to map situations to actions---so as to maximize a numerical reward signal [4]. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards. These two characteristics (trial-and-error search and delayed reward) are the two most important distinguishing features of Reinforcement Learning.

Reinforcement Learning is defined not by characterizing learning algorithms, but by characterizing a learning problem. Any algorithm that is well suited to solving that problem we consider to be a Reinforcement Learning algorithm. Clearly

such an agent must be able to sense the state of the environment to some extent and must be able to take actions that affect that state. The agent must also have a goal or goals relating to the state of the environment.

*B. Competitivet Learning*

Competitive learning is defined as unsupervised learning method of which one of the output neurons firing through competition among the neurons without training sample. It has both of the features of supervised and unsupervised learning methods. It can be applicable for clustering. In such case, inputs are the data in concern while outputs are clusters. The following cost function is usually used.

$$J = \sum_{i=1}^{n} \min_{r} d(x_i, w_r) \tag{1}$$

In the clustering based on competitive learning, $J$ implies the minimum distance $\min.d(x,w)$ between input data $x_i$ and cluster center $w_r$. Therefore, J is sum of dissimilarity within cluster.

*C. Simple Competitivet Learning*

SCL (Simple Competitive Learning) is the simplest competitive learning. The basic idea of the SCL is WTA (Winner Take All). Namely, winner of the neuron gets all. The winner of the neuron is determined with the following equation,

$$\begin{aligned} W_m^T x &= \max_i (W_i^T x) \\ \Delta W_m^T &= \alpha (x - W_i^T x) \end{aligned} \tag{2}$$

where *Wi* denotes weight of the neuron i while *x* denotes input data, and $\alpha$ is the coefficient for determine its convergence speed.

*D. Reinforcement Competitivet Learning*

The following reinforcement competitive learning based online clustering is typical.,

VQ (Vector Quantization),

RGCL (Reinforcement Guided Competitive Learning),

SRGCL (Sustained Reinforcement Guided Competitive Learning).

VQ (Vector Quantization)

Process flow of the VQ is as follows,

*1)* weighting vector is defined as the selected sample vector of each cluster.

*2)* input data belongs the sample vector of cluster $i^*$ which shows the shortest distance between input data and the sample vector.

*3)* sample vector is updated based on the following equation,

$$\Delta w_{ij}^{t+1} = \begin{cases} \alpha (x_j - w_{ij}^t) & if\, i = i^* \\ 0 & if\, i \neq i^* \end{cases} \tag{3}$$

where *t* denotes leaning number of which the number is incremented for each input data. The weight is increased with $\Delta w_i$ when the input data $x_j$ is matched with *t*-th sample vector. Meanwhile, weight is not changed when the input data does not match to any sample vector. Repeating these process, representative vector is updated and then most appropriate clusters are formed.

RGCL (Reinforcement Guided Competitive Learning)

All the clusters of output neurons are represented with Bernoulli units where the weight is assumed to be vector. The distance between input data and weighting vector is calculated with the following equation,

$$s_i = d(x, w_i) \tag{4}$$

Then probability $p_i$ is calculated with the equation (5),

$$p_i = f(s_i) = 2(1 - f(s_i)) \tag{5}$$

where

$$f(x) = \frac{1}{1 + \exp(x)} \tag{6}$$

The probability is increased in accordance with input data is getting close to weighting vector. Therefore, it is probable that the distance between input data and the neuron of which the output is 1. Then the input data belongs the cluster representing the neuron of which the output is 1.

The process flow of RGCL is as follows,

*1)* a data is selected from the samples randomly
*2)* determine a winner neuron $i^*$
*3)* reward $r_i$ of input data $x_i$ is updated as follows,

$$r_i = \begin{cases} 1 & if\ i = i^* and\ y_i = 1 \\ -1 & if\ i = i^* and\ y_i = 0 \\ 0 & if\ i \neq i^* \end{cases} \tag{7}$$

Then weight vector is updated as follows,

$$\Delta w_{ij}^{t+1} = \alpha r_i (y_i - p_i)(x_j - w_{ij}^t) \tag{8}$$

SRGCL (Sustained Reinforcement Guided Competitive Learning)

SRGCL is the method which allows control the convergence speed with the parameter $\eta$ which is added to the RGCL as follows,

$$\Delta w_{ij}^{t+1} = \alpha r_i (y_i - p_i)(x_j - w_{ij}^t) - \eta w_{ij}^t \tag{9}$$

It is not always true that SRGCL is superior to RGCL. The convergence performance depends on the relation between input data and the control parameter. Therefore, the most appropriate control parameter has to be determined.

Tracking Algorithm

N arm bandit problem is defined as the machine learning problem which allows analyze a most appropriate strategy for getting the maximum prize from a slot machine with at least one lever. Learning automata is one of the N arm bandit problem solving methods in an efficient manner.

The action of "draw one the specific lever" is represented as *a*, while play is defined with *t*, together with the probability of the prize is expressed with $\pi_t(a)$, $(n + 1)$-th play of total prize depends on the accumulated prize at n-th play and the current prize. In case of the total prize is increased, the probability is expressed as follows,

$$\pi_{t+1}(a) = \pi_t(a) + \beta(1 - \pi_t(a)) \quad (10)$$

Also, the probability is represented as follows, in case of the total prize is decreased,

$$\pi_{t+1}(a) = \pi_t(a)(1 - \beta) \quad (11)$$

where β is the convergence speed control parameter. If the appreciable actions are always selected, then the total prize is getting close to the maximum prize. This method is one of the learning automata. Namely, reward is provided when it is predicted to win while punishment is given when it is predicted to loose. Through these processes with actions, the total prize is getting closer to the maximum prize.

### *E. Proposed Clutering Method*

In the convergence process of RGCL, it is sometime happened that the convergence speed is decreased and or unstable due to the weight is too large or too small. The method proposed here uses learning automata for adjustment of the weight. Namely, most appropriate prediction of win/loose probability can be done with learning automata. Thus the most appropriate reward and punishment can be given.

Online clustering method based on competitive and reinforcement learning as well as learning automata is proposed here. Namely, winner of the neuron is determined with WTA at first based on competitive neural network of basic learning method, a reward is calculated with the result of the winner neuron based on learning automata. Then the final winner neuron is determined through agent action which has the maximum reward based on reinforcement learning method. Therefore, the proposed method is called PRCL: Pursuit Reinforcement Guided Competitive Learning.

The procedure of the proposed PRCL is as follows,

*1)* Initializing the reward r for each data as follows,

$$r(x, u_i^0) = \frac{1}{n} \quad (12)$$

where *n* denotes the desirable number of cluster, while $u_i^0$ denotes initial cluster center.

*2)* data is selected from the samples randomly

*3)* winner neuron i* を is determined with equation (13)

$$c_i = \arg\min_i \|x - w_i\| \quad (13)$$

*4)* the reward of each output neuron corresponding to input data is updated based on equation (14)

$$r(x, u_i^{t+1}) = \begin{cases} r(x, u_i^t) + \alpha(1 - r(x, u_i^t)) & \text{if } i = i^* \\ r(x, u_i^t)(1 - \alpha) & \text{if } i \neq i^* \end{cases} \quad (14)$$

where $r(x,u^t)$ denote the current reward while $r(x,u^{t+1})$ denotes that for the next learning number, respectively.

*5)* The neuron i* which has the maximum reward is selected by equation (15)

$$u_i = \arg\min_i \|x - w_i\| \quad (15)$$

*6)* weight is updated with the followed equation,

$$\Delta w_{ij} = \begin{cases} \alpha(x - w_{ij}) & \text{if } i = i^* \\ 0 & \text{if } i \neq i^* \end{cases} \quad (16)$$

### *F. Proposed Image Retieval Method*

A huge computation resource is required for image retrieval when template matching is applied to a huge image database, in general. It is possible to reduce the required computer resource by shrinking search areas in concern with an online clustering. It is also possible to shrink the search areas of template matching by division of the image with an appropriate size together with clustering the divided images by using the proposed PRCL of online clustering.

Decimated template image with 1/2 sampling rate is used for template matching to the original large sized image in concern. It can be ensuring 56.25% of matching ratio in maximum. In the second level of the decimation, it is also ensuring 78.4% of matching ratio. Feature extraction is then applied to the divided image regions with feature vectors based on color information and gray scale. After that, clustering is made based on the feature vector space. Thus the search areas can be shrinking.

The actual procedure is as follows,

*1)* Down sampling (1/2 decimation) is applied to the template image

*2)* Vector representation is made for the decimated image

*3)* The proposed PRCL of online clustering is applied to the vectors in the feature space

*4)* Other template images are vectorized and input to the feature space as additional data for the proposed PRCL of online clustering

*5)* Down sampling is applied to the decimated images then the same procedures, 2) to 4) are applied to the down sampled images

*6)* Then the proposed PRCL of online clustering is applied

## III. EXPERIMENTS

### *A. Preliminary Experiments*

(a)Iris (b)Wine

Fig. 1. Examples of the UCI repository

Comparative study on online clustering performance is conducted with Iris, Wine, New thyroid, Ruspini, Chernoff and Fossil datasets form the well-known UCI repository. The examples of the UCI repository of Iris and Wine are shown in Fig.1.

Table 1 shows averaged computation time required for convergence of the proposed and the other conventional methods for Iris data in the UCI repository data. In this case, the parameters for each clustering methods are as follows,

VQ $\alpha = 0.1$
RGCL $\alpha = 0.1$
SRGCL $\alpha = 0.1$, $\eta = 0.0001$
PRCL $\alpha = 0.1$, $\beta = 0.1$

TABLE I. AVERAGED COMPUTATION TIME REQUIRED FOR CONVERGENCE OF THE PROPOSED AND THE OTHER CONVENTIONAL METHODS FOR IRIS DATA IN THE UCI REPOSITORY DATA

| Method | Iris | Wine | Fossil | Ruspini | Thyroid |
|---|---|---|---|---|---|
| PRCL | 100 | 220 | 50 | 20 | 90 |
| RGCL | 250 | 490 | 120 | 45 | 240 |
| VQ | 370 | 730 | 190 | 60 | 350 |

In this case, the maximum learning number is set at 15000 while the parameters for each method is set as follows,

VQ $\alpha = 0.1$
RGCL1 $\alpha = 0.1$
RGCL2 $\alpha = 0.5(t <= 500)$, $\alpha = 0.1(t > 500)$
SRGCL1 $\alpha = 0.1$, $\eta = 0.0001$
SRGCL2 $\alpha = 0.5(t <= 500)$, $\alpha = 0.1(t > 500)$, $\eta = 0.0001$
PRCL $\alpha = 0.1$, $\beta = 0.1$

It is found that the convergence performance of RGCL and SRGCL has influenced by the parameter $\alpha$. The averaged processing time over 100 times of 4000 of learning number (which is defined as 1 set) is evaluated. Table 2 shows just one of the examples of evaluation results for Iris dataset.

TABLE II. AVERAGED COMPUTATION TIME REQUIRED FOR CONVERGENCE OF THE PROPOSED AND THE OTHER CONVENTIONAL METHODS FOR IRIS DATA IN THE UCI REPOSITORY DATA

| | PRCL | VQ | RGCL | SRGCL |
|---|---|---|---|---|
| Time (s) | 0.057 | 0.043 | 0.129 | 0.144 |

From Table 1, it is found that the proposed method is second fastest method. The proposed method, however, shows the highest convergence performance in terms of convergence speed and stability.

Table 3 shows clustering errors of the proposed and the other conventional methods for each UCI repository data. All the parameters are set as follows,

VQ $\alpha = 0.1$
RGCL $\alpha = 0.1$
SRGCL $\alpha = 0.1$, $\eta = 0.0001$
PRCL $\alpha = 0.1$, $\beta = 0.1$

TABLE III. CLUSTERING ERRORS OF THE PROPOSED AND THE OTHER CONVENTIONAL METHODS FOR EACH UCI REPOSITORY DATA

| Error (%) | PRCL | VQ | RGCL | SRGCL |
|---|---|---|---|---|
| Iris | 19.09 | 14.99 | 17.01 | 14.09 |
| Wine | 32.51 | 31.72 | 29.41 | 28.94 |
| Ruspini | 8.77 | 9.15 | 7.39 | 5.87 |
| Fossil | 21.45 | 27.24 | 26.44 | 26.44 |
| New thyroid | 22.69 | 26.15 | 31.33 | 40.82 |

From Table 3, it is found that all of online clustering methods show almost same (within 5%) clustering performance for relatively simple dataset of Iris, Ruspini, while the clustering performance are different for comparatively complicated dataset, Fossil, New thyroid. In such case, the proposed PRCL shows the highest performance. In particular, clustering performance of PRCL for New thyroid is 3.54% better than VQ, and 8.64% better than RGCL as well as 18.13% better than SRGL. It is because that the PRCL is functioning for adjustment of the complexity of the input data by the learning automata.

### *B. Image Retrievals*

Fig.2 shows the original image of Saga, Japan which is acquired on April 17 2007 with ASTER/VNIR (Airborne Sensor for Thermal Emission and Reflection/Visible and Near Infrared Radiometer) onboard Terra satellite used for the experiment. The image consists 4980 by 4200 pixels with three bands. From the original image, 120 by 120 pixels' sub-image is extracted for image retrieval experiment.

VQ and the proposed PRCL is applied to the image retrieval. The parameter used is as follows,

VQ $\alpha = 0.1$
PRCL $\alpha = 0.1$, $\beta = 0.1$

Fig. 2. Original ASTER/VNIR image of Saga acquired on 17 April 2007

The actual procedure is as follows,

*1)* 120 by 120 of template image is extracted from the original ASTER/VNIR image

*2)* Clustering is applied to the 15 dimensional features extracted through decimation with decimation factor of 60 pixels created from the original image

*3)* Add the features derived from the template image and make the online clustering with VQ and the proposed PRCL (the first clustering)

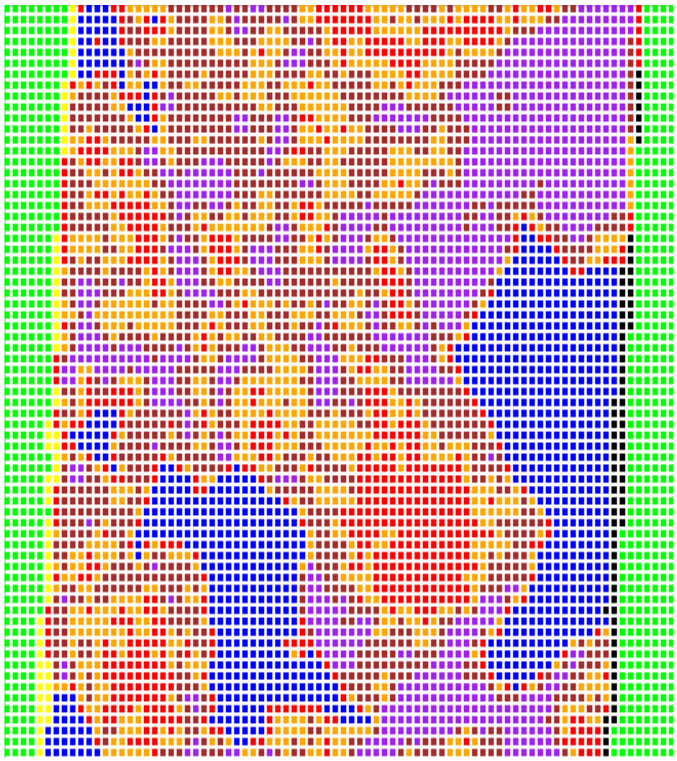
*4)* Selected cluster region in the original image is expanded to one block further,

*5)* Then the decimation with the decimation factor of 30 pixels is applied to the selected cluster region in the original image

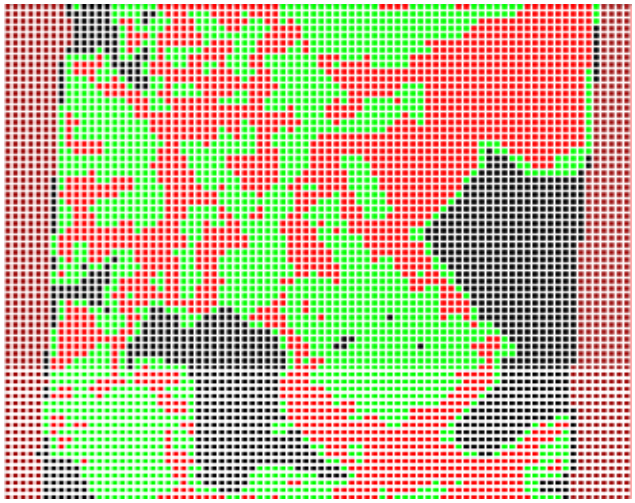
*6)* The online clustering of VQ or the proposed PRCL is applied to the decimated image (the second clustering)

*7)* Thus the best match image portion is retrieved with referring to the clustering result through the matching between template and the original image.

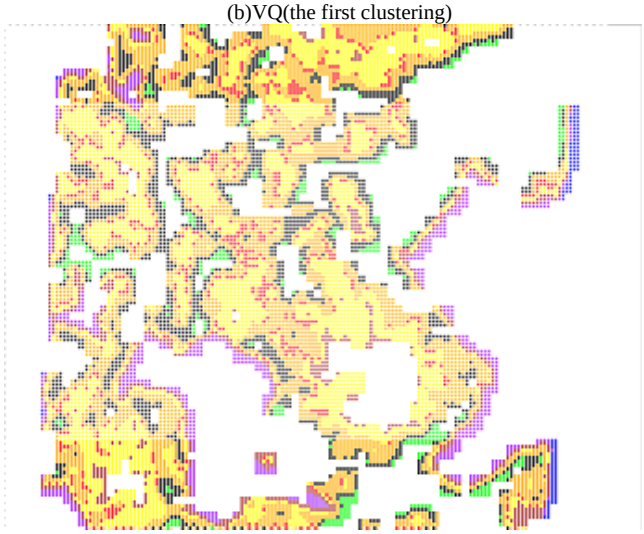
Fig.3 (a) and (b) shows the clustered image of the first clustering while Fig.3 (c) and (d) shows those for the second clustering. From these images, it is found that the number of clusters of the proposed PRCL is greater than that of VQ. Meanwhile, Fig.4 shows the first clustering results of residual error (the cost function of J) through the individual 10 times trials.



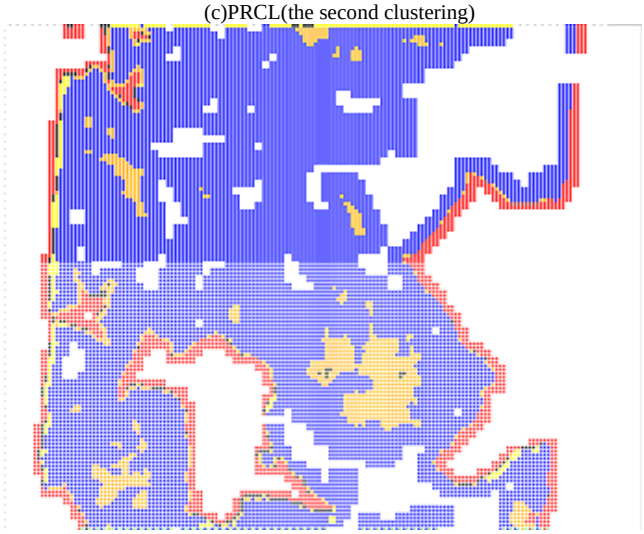
(a)PRCL(the first clustering)



(b)VQ(the first clustering)



(c)PRCL(the second clustering)



(d)VQ(the second clustering)

Fig. 3. Clustered result images

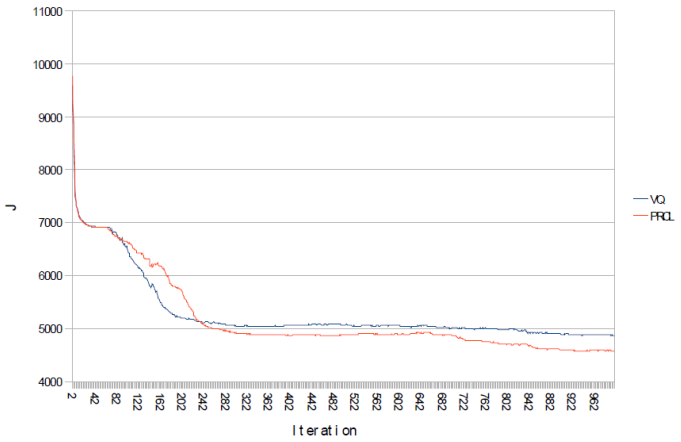



Fig. 4. Convergence processes of the image portion retrievals (120x120) from 4980x4200 of ASTER/VNIR image for the proposed PRCL method and the existing conventional VQ

From the figure, it is confirmed that convergence performance of the proposed PRCL is superior to that of VQ. Table 4 shows elapsed time for the proposed and existing template matching, pyramid search as well as conventional VQ methods.

TABLE IV. ELAPSED TIME FOR THE PROPOSED AND EXISTING TEMPLATE MATCHING, PYRAMID SEARCH AS WELL AS CONVENTIONAL VQ METHODS

| _ | PRCL | VQ | Pyramid Search | Template Matching |
|---|---|---|---|---|
| Elapsed time(s) | 9.85 | 10.16 | 10.05 | Approx.6 hours |

From Table 4, the elapsed time of the proposed PRCL shows shortest followed by the conventional Pyramid Search, VQ and the conventional Template Matching. Although VQ is the most appropriate image retrieval method, traditionally, the proposed PRCL can reduce the process time with 3.05%.

Furthermore, the proposed PRCL can use previously reduced cluster results. Therefore, much faster image retrievals can be expected referring to the database of the cluster results for the proposed PRCL. Table 5 shows the processing time for the conventional Pyramid Search, VQ and the proposed PRCL with referring to the database.

TABLE V. ELAPSED TIME FOR THE PROPOSED PRCL AND EXISTING PYRAMID SEARCH AS WELL AS THE CONVENTIONAL VQ METHODS

| _ | PRCL | VQ | Pyramid Search |
|---|---|---|---|
| Elapsed Time_(s) | 7.54 | 7.68 | 10.05 |

From Table 5, it is found that the proposed PRCL can reduce the process time by 1.82% in comparison to VQ while by 24.98% comparing to the conventional Pyramid Search, respectively.

It is also possible to retrieve the image portion in concern with online clustering only for all the required process. Namely, decimation with the decimation factor of 1/2 is applied to the original image recursively until the pixel interval becomes one pixel. The process time for this image retrieval method is evaluated for VQ and the proposed PRCL. Table 6 shows the evaluation result with the original image size of 128 by 128 pixels. From Table 6, it is found that the proposed PRCL achieves 9.79% shortened process time in comparison to the VQ.

TABLE VI. ELAPSED TIME FOR THE PROPOSED AND CONVENTIONAL VQ METHODS

| _ | PRCL | VQ |
|---|---|---|
| Elapsed time(s) | 87.94 | 96.55 |

It is suspected that image retrievals require much longer time for the distance between the template image and the portion of original image is too long. The time required for image portion retrievals of the proposed PRCL is examined with the function of the distance. As the results of examination, it is confirmed that the process time for long distance is much longer than that for short distance. It, however, only 1% longer time is required when the distance is 10 times long.

It is also suspected that process time is varied by the complexity of the image portion. Therefore, another examination is conducted for a relation between process time and variance of the image portion. As the results of the examination, it is found that the process time depends on the variance of the image portion. Therefore, the process time for the areas of sea, forest, etc. is much shorter than those for urban, river, road network, etc. as shown in Fig.5

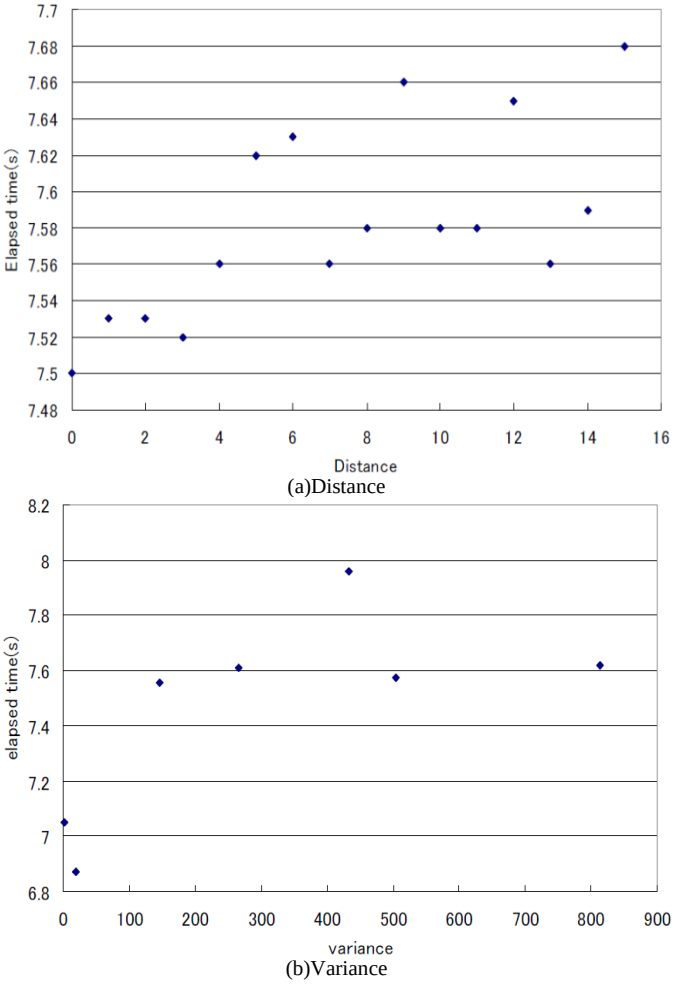



Fig. 5. Process time of the proposed PRCL as functions of the distance between template and image portion and variance of image portion

### C. Case Study for Image Retrievals when Scales do not Match between Template and the Original Images

Image retrieval might not be worked when the scales of the template and the original images do not match without a prior information of the scale. Even for the case with a prior information of the scale, it is hard to get a fine matching between both images when the scales are different. The procedure for the aforementioned case is as follows,

*1)* 120 by 120 pixels of image portion is extracted from the n times of the original image

*2)* The original image is sampled with 60 pixels of interval. Then clustering is applied to the sampled image in the 15 dimensional feature space of which the feature vectors are derived from the sampled images

*3)* Feature vector of the template image is added to the feature space. Then online clustering is applied.

*4)* Cluster region is expanded by one block at the selected cluster region in the original image

*5)* The secondary clustering applied for the sampled image in the search regions with 30 pixels' interval

*6)* By referring to the clustered result, nearest vector is selected from the original image to the template image vector. Then the image scale ratio is calculated with the norm of the template vector and the norm of the original image vector

*7)* Scale conversion is applied to the template image. Then online clustering is applied again

*8)* The image scale ratio is calculated again. If the image scale ratio is not changed largely, then the iteration process is reckoned to be converged. If not, the aforementioned processes are repeated

*9)* After that, template image matching can be done with the calculated image scale ratio and the nearest image vectors of the template and the original image portion

Image retrieval results of the case are shown in Table 7 with the image scale ratio ranges from 0.5 to 2.0. Although the matching accuracy is quite good (less than one pixel) for the case that image scale ratio is one, the matching accuracy is decreased in accordance with the image scale ratio. On the other hand, the matching accuracy is getting poor when the image scale ratio is decreased less than one sharply.

TABLE VII. ELAPSED TIME, ESTIMATED IMAGE SCALE, ERROR IN UNIT OF PIXEL AND PERCENT FOR THE CASE THAT IMAGE SCALE DOES NOT MUCH EACH OTHER BETWEEN TEMPLATE AND SATELLITE IMAGES

| - | Estimated scale | Error(%) | Error(pixel) | Elapsed time(s) |
|---|---|---|---|---|
| 0.5 | 0.78 | 56% | 33.6 | 192.8 |
| 1.5 | 1.41 | 6% | 7.2 | 78.3 |
| 2 | 1.69 | 15.5% | 18.6 | 153.9 |

## IV. CONCLUSION

Pursuit Reinforcement guided Competitive Learning: PRCL based on relatively fast online clustering that allows grouping the data in concern into several clusters when the number of data and distribution of data are varied of reinforcement guided competitive learning is proposed. One of applications of the proposed method is image portion retrievals from the relatively large scale of the images such as Earth observation satellite images. It is found that the proposed method shows relatively fast on the retrievals in comparison to the other existing conventional online clustering such as Vector Quatization: VQ. Moreover, the proposed method shows much faster than the others for the multi-stage retrievals of image portion as well as scale estimation.

Also, it is found that the matching accuracy is quite good (less than one pixel) for the case that image scale ratio is one. Meanwhile, the matching accuracy is decreased in accordance with the image scale ratio. On the other hand, the matching accuracy is getting poor when the image scale ratio is decreased less than one sharply.

Further investigation is required for another applications of the proposed online clustering.

## ACKNOWLEDGEMENTS

Author would like to thank Dr. Ali Ridoh Barakbah of EEPIS, Indonesia and Dr. Bu Quang Quong of former student of Saga University for their efforts for conducting the experiments.

## REFERENCES

[1] G. Karypis, E.H. Han, V. Kumar, Chameleon: a hierarchical clustering algorithm using dynamic modeling, IEEE Computer: Special Issue on Data Analysis and Mining, 32(8):68W5, 1999.

[2] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Clustering algorithms and validity measures, Proceedings of the 13th International Conference on Scientific and Statistical Database Management, July 18–20, IEEE Computer Society, George Mason University, Fairfax, Virginia, USA, 2001.

[3] W.H. Ming and C.J. Hou, Cluster analysis and visualization, Workshop on Statistics and Machine Learning, Institute of Statistical Science, Academia Sinica, 2004.

[4] R.S. Sutton, A.G. Barto, Reinforcement learning: an introduction, The MIT Press, 1998.

[5] Genevieve B. Orr, Simple competitive learning, Neural networks course, http://www.willamette.edu/~gorr/classes/cs449/Unsu pervised/competitive.html.

[6] A. Likas, A reinforcement learning approach to on-line clustering, Neural Computation, 11:1915- 32, 1999.

[7] Ali Ridho Barakbah and Kohei Arai, Pursuit reinforcement competitive learning: An approach for on-line clustering, Proceedings of the IEEE Indonesian Chapter of the 2nd Information and Communication Technique Seminar, ISSN1858-1633, 45-48, 2006.

[8] Akira Yoshizawa, Kohei Arai, Clustering Method Based on Genetic Algorithm Using Spectral and Spatial Context Information, Journal of Image Electronics Society of Japan, Vol.31, No.2, 202-209, 2003

[9] Akira Yoshizawa, Kohei Arai, Clustering Method Based on Genetic Algorithm Using Spectral and Spatial Context Information, Journal of Image Electronics Society of Japan, Vol.31, No.2, 202-209, 2003.

[10] Kohei Arai, Akira Yoshizawa, Koichi Tateno, Application of Messy Genetic Algorithm for Satellite Image Clustering, Journal of Japan Society of Photogrammetry and Remote Sensing, Vol.41, No.5, 34－41, 2003.

[11] Kohei Arai, Learning processes of image clustering method with density maps derived from Self-Organizing Mapping(SOM), Journal of Japan Society of Photogrammetry and Remote Sensing, 43,5,62-67 (2004)

[12] Kohei Arai, Non-linear merge and split method for image clustering, Journal of Japan Society of Photogrammetry and Remote Sensing, 43, 5, 68-73 (2004)

[13] Ali Ridho Barakbah and Kohei Arai, Revised pattern of moving variance for acceleration of automatic clustering, Electric and Electronics Polytechnics in Surabaya EEPIS Journal, 9, 7, 15-21 (2004)

[14] Kohei Arai, Bu Quang Quong, ISODATA clustering with parameter estimation based on genetic algorithm, taking into account concave characteristics of probability density function, Journal of Japan Society of Photogrammetry and Remote Sensing, 47, 1, 17-25, 2008

[15] Kohei Arai, Bu Quang Quong, Image portion retrievals in large scale imagery data by using online clustering taking into account pusuite algorithm based Reinforcement learning and competitive learning, Journal of Image Electronics Society of Japan, 39, 3, 301-309, 2010

[16] Kohei Arai, Bu Quang Quong, Covergence performance improvement of online clustering with reinforcement and competitive learning with learning automaton, Journal of Image Electronics Society of Japan, 40, 2, 361-168, 2011.

[17] Kohei.Arai, Bu Quang Quong, Comparative study between the proposed GA based ISODATA clustering and the conventional clustering methods, International Journal of Advanced Computer Science and Applications, 3, 7, 125-131, 2012.

[18] Kohei.Arai, Image clustering method based on density maps derived from Self Organizing Mapping: SOM, International Journal of Advanced Computer Science and Applications, 3, 7, 102-107, 2012.

[19] Kohei.Arai, Clustering method based on Messy Genetic Algorithm: GA for remote sensing satellite image clustering, International Journal of Advanced Research in Artificial Intelligence, 1, 8, 6-11, 2012.

[20] Kohei Arai, Visualization of learning process for back propagation Neural Network clustering, International Journal of Advanced Computer Science and Applications, 4, 2, 234-238, 2013.

[21] Kohei Arai, Image clustering method based on Self-Organization Mapping: SOM derived density maps and its application for Landsat Thematic Mapper image clustering, International Journal of Advanced Research in Artificial Intelligence, 2, 5, 22-31, 2013.

[22] Kohei Arai, Cahya Rahmad, Comparative study between the proposed shape independent clustering method and the conventional method (k-means and the others), International Journal of Advanced Research in Artificial Intelligence, 2, 7, 1-5, 2013.

[23] Kohei Arai, Genetic Algorithm utilizing image clustering with merge and split processes which allows minimizing Fisher distance between clusters, International Journal of Advanced Research in Artificial Intelligence, 2, 9, 7-13, 2013.

[24] Kohei Arai, Fisher distance based GA clustering taking into account overlapped space among probability density functions of clusters in feature space, International Journal of Advanced Research in Artificial Intelligence, 2, 11, 32-37, 2013.

[25] Ali Ridho Barakbah, K.Arai, Centronit: Initial Centroid Designation Algorithm for K-Means Clustering, EMITTER: International Journal of Engineering Technology, 2, 1, 50-62, 2014.

## AUTHORS PROFILE

**Kohei Arai,** He received BS, MS and PhD degrees in 1972, 1974 and 1982, respectively. He was with The Institute for Industrial Science and Technology of the University of Tokyo from April 1974 to December 1978 and also was with National Space Development Agency of Japan from January, 1979 to March, 1990. During from 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He moved to Saga University as a Professor in Department of Information Science on April 1990. He was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology during from 1998 to 2000. He was a councilor of Saga University for 2002 and 2003. He also was an executive councilor for the Remote Sensing Society of Japan for 2003 to 2005. He is an Adjunct Professor of University of Arizona, USA since 1998. He also is Vice Chairman of the Commission-A of ICSU/COSPAR since 2008. He received Science and Engineering Award of the year 2014 from the minister of the ministry of Science Education of Japan and also received the Bset Paper Award of the year 2012 of IJACSA from Science and Information Organization: SAI. In 2016, he also received Vikram Sarabhai Medal of ICSU/COSPAR and also received 20 awards. He wrote 34 books and published 520 journal papers. He is Editor-in-Chief of International Journal of Advanced Computer Science and Applications as well as International Journal of Intelligent Systsems and Applications. http://teagis.ip.is.saga-u.ac.jp/

# Direction for Artificial Intelligence to Achieve Sapiency Inspired by Homo Sapiens

Mahmud Arif Pavel
Dept. of Biological Sciences
St. John's University
NY, USA

***Abstract*—Artificial intelligence technology has developed significantly in the past decades. Although many computational programs are able to approximate many cognitive abilities of Homo sapiens, the intelligence and sapience level of these programs are not even close to Homo sapiens. Rather than developing a computational system with the intelligent or sapient attribute, I propose to develop a system capable of performing functions that could deem as intelligent or sapient by Homo sapiens or others. I advocate converting current computational systems to educable systems that have built-in capabilities to learn and be taught with a universal programming language. The idea is that this attempt would help to attain computational actions in artificial means, which could be viewed as similar to human intelligent and sapient acts. Although this paper is seemingly speculative, some feasible elements are proposed to advance the field of Artificial Intelligence.**

***Keywords—artificial sapience; sapient agent; artificial intelligence; bio-inspired AI***

## I. INTRODUCTION

Homo sapiens ('wise man' in Latin, coined by Carl Linnaeus in 1758) is unique and foremost among the rest of the creatures because of the breadth of Homo Sapiens' wisdom [1], [2]. Wisdom or sapience is the ability to reflect the act of using knowledge and experience [3], [4]. Beside a philosophical explanation of wisdom [2], [5], [6], a formal academic concept of computational or artificial sapience (wisdom) has been developed considering the aspect of the learning, adaptation, and judgment capabilities [4], [7], [8]. However, in this paper, I will define a sapient system simply as a computational entity that can generate functions or actions that are deemed smart and wise by Homo sapiens or others. Many believe that the peak of Artificial Intelligence (AI) will be achieved when AI can learn and perform every intelligence or sapience based act of Homo Sapiens (HS) — though winning this achievement seems very distant [9], [10].

Hitherto, most AI researchers place substantial effort to attain the automation of jobs that humans can perform [11]. However, the current paradigm of AI research is shifting to the development of AI that can be teachable or trainable to perform work as similar to what humans learn to perform [11]–[13]. Indeed, when AI is trained, it could display exceptional ability such as generating a sensible explanation of image regions in natural language statements [14]. Therefore, it is fair to assume that building a simple educable AI is the critical step to achieve HS level intelligence [8], [11]. Yet the development of such simple AI is expected to be difficult and take many years of study and investigation. For simplicity, to describe an AI agent that can be taught and trained, henceforth I will name it as Artificial Sapiens (AS) as the ultimate goal of this agent is to become Homo sapiens.

One of the approaches to achieve AS is to mimic how HS thinks and works [10]. HS provides the necessary hints on how to advance the current AI to AS level [7], [15]. Artificial neural networks and behavior dependent robots, personal digital assistants are few examples of AI inspired by HS [16]. However, the functionalities of these examples are only slightly equivalent to HS. To propose a step towards the development of an AS, I have taken the inspiration from the early development of HS. HS learns and develops from its childhood to adulthood [17], [18]. Implementing the very fundamental methodology of the HS learning process is now a well-suggested strategy to achieve an educable AI i.e. the Artificial Sapiens [11], [12]. Though achieving AS will be very difficult, here I will describe a plan to bring this endeavor one step forward.

I first examined the attempt to obtain an AS by mimicking HS child learning capabilities. I then proposed to transform the current non-intellectual electronic device such as a desktop computer, laptop, mobile phone or other to an Artificial Sapiens — by simply giving their operating systems (OS) and applications with a learning capability.

To get an idea how to change an OS to AS, I compared the human system to software and hardware of the computer. I presented an alternative approach to the comparison rather than customarily comparing computer system to human brain system [19]. To generate a teachable and trainable OS, I proposed to make the subprograms (applications) of the OS programmable with a universal programming or teachable language by its ordinary users. Inspired from how a child learns from 'Do' or 'Do not do' statements and develops wisdom (sapience) [18], my idea is when applications are programmed to modify its own output through interactions with its users, it could display a level of intelligence or sapience. I presented a few examples of intelligent or sapient acts that could be achieved by programmable or teachable applications within non-intellectual operating systems. Certainly, I do not pretend to have expertise on this matter but here I offered some observations from the biological perspective that could have some potential to achieve HS level AS.

## II. A Child Computer

Before achieving an adult level sapient system, making a computer/AI system that simulates a child is one of the core goals of AI researcher [10], [13]. Although something like the child computer is overly optimistic, this is a necessary step to generate a HS level AS since HS continuously learns to be a sapience from its childhood to adulthood [11], [12]. HS has the knowledge, learning and skill acquisition ability — the mechanism is present at the very birth[18], [20]. HS has the ability to be taught by others or by own self through observation, imitation, assimilation and experimentation. HS employs existing knowledge and skills, and burgeons with more knowledge and skills [6], [8]. A childlike computer should have the above ability to learn like HS and to grow to a higher maturity level. Considering that the child-brain can be programmed, the current approach is to make a device or a robot that mimics a child or parts of a child's ability [21]. However, there is still no such HS level child computer or close to it. It seems too difficult to create. In this paper, I propose to convert existing computational devices (e.g. laptop or mobile) to a child-like system. At the beginning of a child's learning stage, an adult HS mostly teaches a child what to do or what not to do [17], [18], [20]. Similarly, current devices could have functions that can be done or cannot be done based on its users' instructions. In other words, these devices could learn what to do or what not to do from its users. These teachable devices, then, can grow as they continuously learn and evolve to show a level of intelligence or sapience. Thus, these devices can also be attributed as Artificial Sapiens. In the next sections, I will elaborate more about how to obtain a teachable device based on the comparison between a human and a computer.

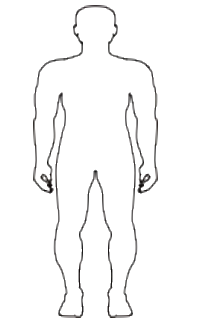
## III. New Paradigm of Comparison between Computer and Homo Sapiens

There are many AI systems that are inspired by biology [15], [16]. Observation of real life could provide valuable insights on the plausible design of Artificial Sapiens systems. Real life comparison of AI is often limited to the comparison of the computer with the brain of HS [19]. The focal point of comparison between HS and Computer/AI is that the computer inputs, stores, processes, and outputs information somewhat similar to an HS brain. A few basic differences between HS and computer information processing include central versus distributed control, sequential versus parallel input, exclusive versus overlaid output, and low versus high self-processing [19]. The current comparison between HS and computer/AI is less focused on how to achieve better human intelligent or sapient activity. Therefore, here I revisit the original comparison by drawing further inspiration from real HS systems. I explored an alternative comparison by trying to resemble the whole HS system with a computer system.

| Computer Devices | External Input | Internal Output |
|---|---|---|
| Webcam | Light | Electrical signals |
| Microphone | Sound | Electrical signals |

| Computer Devices | Internal Input | Internal Output |
|---|---|---|
| Processor | Electrical signals | Process instructions that drive a computer |
| Hard Disk | Electrical signals | Store information |

| Computer Devices | Internal Input | External Output |
|---|---|---|
| Monitor | Electrical signals | Display |
| Speaker | Electrical signals | Sound |




| Human Devices | External Input | Internal Output |
|---|---|---|
| Eyes | Light | Electrical nerve impulses |
| Ears | Sound | Electrical impulses |
| Lungs | Air | Absorb oxygen and release carbon dioxide |

| Human Devices | Internal Input | Internal Output |
|---|---|---|
| Heart | Blood | blood Circulation |
| Brain | Electrical nerve impulses | Coordination, Perception, Store memories |

| Human Devices | Internal Input | External Output |
|---|---|---|
| Larynx | Electrical nerve impulses | Sound |
| Voluntary Muscles | Electrical nerve impulses | Movement |

Fig. 1. Comparison between Computer and Homo sapiens based on the internal devices

Computers are built with devices so as the Homo sapiens are built with organs (**Fig. 1**). Some of the devices in both computer and HS, listed in **Fig. 1**, are completely internal (both input and output). Some are internal but have external output, and some have external input but produce internal output. These comparisons reveal that there is no such device in both

computer and HS, which takes only external input and provides only external output. However, the computer or HS as a whole might fall into this category (external input and external output). Interestingly, in both case of HS or computer, all of the input has to be processed internally (**Fig. 1**). This comparison also indicates the missing parts that are necessary to generate an AS — the brain and the memory. Brain processes and coordinates what HS learns through instructions and memory stores those instructions [22].

Building an HS equivalent brain is impossible and unnecessary as well. But a device (for example, a computer), could have a simple 'brain software'. If the existing software of the device is modifiable or programmable by every user to generate new functions, the 'brain software' is necessary to coordinate and process those programming instructions. A memory could also be introduced into the device to store the user's given instructions. Such an instructive device could provide a platform to the development of an AS. But at this point, the most critical aspect of the device is to develop a language by which a device can be instructed or taught to modify or generate its output by its every user.

## IV. A Language to Teach Artificial Sapiens

The computer (or any other electronic device) is run by software instructed by the language of the computer [23]. The language here is any type of programming language that is used to write any programs for any computational devices [23], [24]. Although programming languages have evolved to fifth-generation programming languages (5GL), these languages are converting the given instructions into the machine language – ultimately to bits (0 and 1) [23], [24]. In contrast to the computer's only one distinct type of language (bits), Homo sapiens can have two languages: extrinsic and intrinsic. Extrinsic natural language, which the HS speaks or signs is a means of expressing and communicating ideas, emotions or desires externally with other beings. The other is the intrinsic language inside the human body that all the internal body elements use to communicate with each other for the survival as an entire HS. This intrinsic language of HS, an abstract language idea, is the instructions provided by deoxyribonucleic acids (DNA) and other cellular elements [22]. Intrinsic language accounts for the HS innate biological motivations for gaining knowledge and improving its own intelligence and sapience. Extrinsic language is the external medium for HS to acquire the knowledge, wit and wisdom. The ability of HS to generate salient meanings (intelligent action or knowledge), which is useful to others, comes from what HS learns from its surroundings through the extrinsic language.

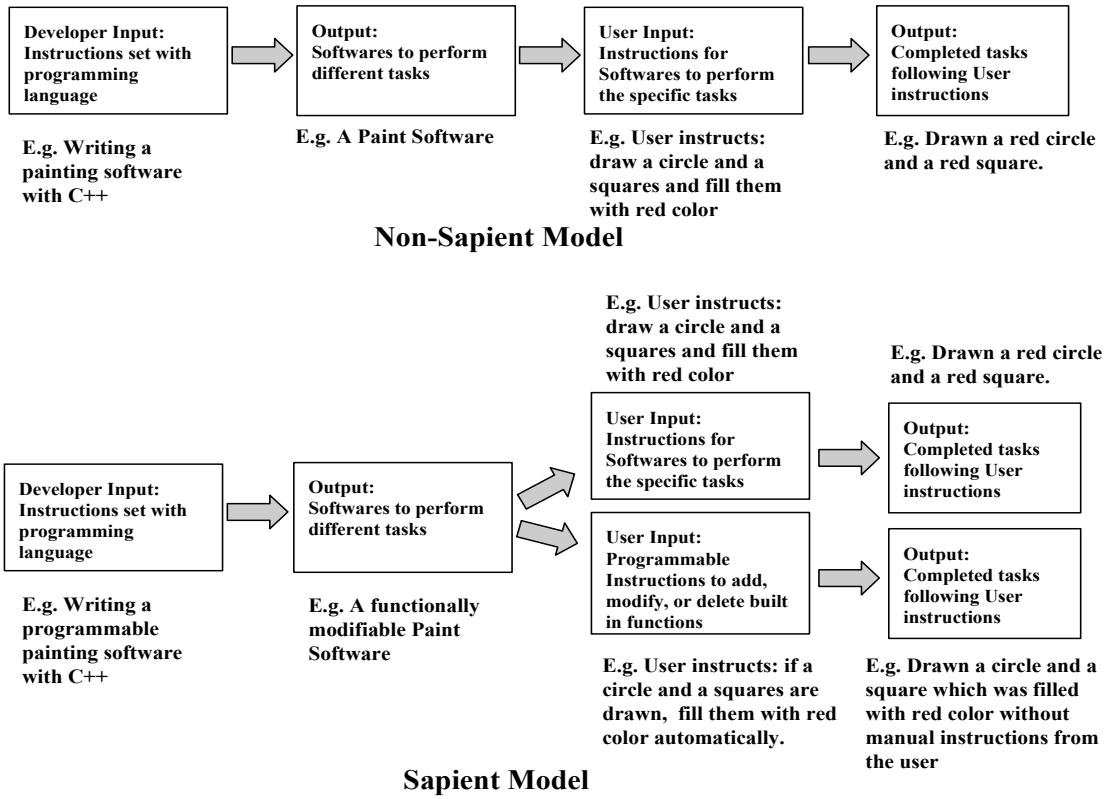



Fig. 2. Scheme to convert a non-sapient model to sapient model

If sapience is attributed to a computer system by its useful knowledge or action, a computer can be made teachable by introducing a second type of extrinsic language — similar to what HS have to learn and communicate. Currently, programmers and developers write software that can take pre-fixed input from the user and generate specific results (**Fig. 2**). I propose, to make a teachable computer, programmers and developers should make software (e.g. applications in OS) that could be further programmable/modifiable by all of its users (**Fig. 2**). In this case, a simple universal programming language

(extrinsic language for the computer) could be developed. This language should be very close to the natural language of the users and accessible to all of the device users through textual or natural language user interfaces.

The theory to render an AS is that when software or applications are programmable such that the users can modify their functional outputs, they could display a level of intelligence or sapience. Example of how this theory could render intelligent and sapient act: current operating systems for mobile devices such as Android or iOS usually contain a Clock application (app) that shows the time and provides time-related utilities (e.g. setting up an alarm). These devices also contain an Stock application that displays real-time stock updates. If the Clock and Stock apps are programmable at a certain level such that its user could teach the stock app to display the closing time stock prices (NY 4:30pm) of what the user bought — this function could be considered as an intelligent act. At the same time, the stock app could also suggest whether the user should sell or keep his or her stock if the app was taught the trend of stocks when its prices go high or low — and this function then could be considered as a sapient act.

A device that is programmable or learnable is the Artificial Sapiens. AS can be made to learn from all of its users and perform the learned functions to every user regardless of whom it learned from (updated through sharing/clouding). If a new user experiences the stock app that just learned to be smart, the user would surely admire such intelligence and sapience level. Following the above example, AS could thus start to exhibit, if not the highest, a low-level wit and wisdom.

Here is another example: nowadays, grouping the emails by priority in a Mail application is very common. Consider an email group named 'VIP'. In an AS system, a user could program or teach it to remind him or her about an unread VIP email every five minutes. The user could also teach that if the email is from his family member and the content has an identifying phrase such as 'very urgent', then the message would constantly pop up with a beep until the user confirms that it has been read. In this case of a newly learned mail app, one could easily deem the first function as an intelligent act and the second function as a sapient act.

While judging an AS, one might ask why not provide all the future possible functions of AS as built-in? This question is similar to the question, why a child is not born as an adult? Big companies like Apple and Google have made their application development platform open to the developers. That's why thousands of unique apps have been and are being created. When all of the apps will become more functional and interconnected by a universal simple programming language, we could expect the evolution of Artificial Sapiens from current non-intellectual operating systems.

## V. Conclusion

I have proposed a plan to achieve human-level artificial intelligence or sapience underlying the fact that humans are the judge to attribute intelligence or sapience based on actions. The proposed way of making devices that are programmable/teachable by each of its users, in principle, is easy to build with the currently available technology. But an Artificial Sapiens that mimics Homo Sapiens may be many years away.

I proposed to make the current computational devices such that they can learn and be taught by a computational language. The plan would be to convert a computational system to a teachable system, much as a human child gains knowledge and wisdom as the child grows. This teachable system should have built-in facilities for learning (through a language) that is similar to what human infants have. Initially, these teachable systems could be less educable. But at some point, in the future, they could significantly be educated.

My proposal may appeal to researchers with distant interests, as it overlaps the understanding of biological and computer science. Ideally, artificial sapient systems would be an appropriate artifact of biological sapient systems. In the long run, AS are expected to be self-reliable, adaptive, socially-interactive and to be competent in the jobs which require collective actions (such as: conforming or co-ordinating a team, arbitration or negotiation).

More understanding on Artificial Sapiens will emerge after they start to learn and interact with the Homo Sapiens. If the proposal I have outlined is followed successfully, one day AS could have the performance of humans and could replace their difficult jobs. The conceptual task of developing Artificial Sapiens may seem formidable, but perhaps the alternative approach that I proposed might prove effectual.

## Acknowledgment

I thank Zinat Sharmin for comments and valuable discussions.

## References

[1] C. von Linné, Systema naturae per regna tria naturae secundum classes, ordines, genera, species cum characteribus, differentiis, synonymis, locis; editio duodecima.-. 1766.

[2] P. Baltes, U. Staudinger, and A. Maercker, "People nominated as wise: a comparative study of wisdom-related knowledge.," Psychol., 1995.

[3] S. Ryan, "Wisdom," 2007.

[4] R. V. Mayorga, "A Paradigm for Sapient (Wise) Systems: Implementations, Design&Operation," in Toward Artificial Sapience, London: Springer London, 2008, pp. 143–173.

[5] T. Karelitz and L. Jarvin, "The meaning of wisdom and its development throughout life," life-span Dev., 2010.

[6] L. I. Perlovsky, "Sapience, Consciousness, and the Knowledge Instinct (Prolegomena to a Physical Theory)," in Toward Artificial Sapience, London: Springer London, 2008, pp. 33–60.

[7] M. van Otterlo, M. Dastani, M. Wiering, and J.-J. Meyer, "A Characterization of Sapient Agents," in Toward Artificial Sapience, London: Springer London, 2008, pp. 129–141.

[8] P. Noriega, "Sapients in a Sandbox," in Toward Artificial Sapience, London: Springer London, 2008, pp. 105–115.

[9] E. Feigenbaum, "Some challenges and grand challenges for computational intelligence," J. ACM, 2003.

[10] NOW, "The Quest for Artificial Intelligence," Cambridge Univ Press.

[11] N. Nilsson, "Human-level artificial intelligence? Be serious!," AI Mag., 2005.

[12] J. McCarthy, "The well-designed child," Artif. Intell., vol. 172, no. 18, pp. 2003–2014, Dec. 2008.

[13] Turing, "Computing machinery and intelligence," Mind, 1950.

[14] Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," Proc. IEEE Conf., 2015.

[15] D. Floreano and C. Mattiussi, Bio-inspired artificial intelligence: theories, methods, and technologies. 2008.

[16] J. William, "Directions for Bio-Inspired Artificial Intelligence," J. Comput. Sci. Syst. Biol., 2012.

[17] P. Bloom, Descartes' baby: How the science of child development explains what makes us human. 2009.

[18] D. Boyd and H. Bee, The developing child. 2012.

[19] Whitworth and H. Ryu, "A comparison of human and computer information processing," 2008.

[20] P. Harris, Children and emotion: The development of psychological understanding. 1989.

[21] T. Minato, Y. Yoshikawa, and T. Noda, "CB2: A child robot with biomimetic body for cognitive developmental robotics," 2007 7th IEEE-RAS, 2007.

[22] J. Reece, L. Urry, M. Cain, and S. Wasserman, Campbell biology. 2011.

[23] RAJARAMAN AND N. ADABALA, FUNDAMENTALS OF COMPUTERS. 2014.

[24] R. Wexelblat, History of programming languages. 2014.

# Prediction of Employee Turnover in Organizations using Machine Learning Algorithms

## A case for Extreme Gradient Boosting

Rohit Punnoose, PhD candidate
XLRI – Xavier School of Management
Jamshedpur, India

Pankaj Ajit
BITS Pilani
Goa, India

***Abstract*—Employee turnover has been identified as a key issue for organizations because of its adverse impact on work place productivity and long term growth strategies. To solve this problem, organizations use machine learning techniques to predict employee turnover. Accurate predictions enable organizations to take action for retention or succession planning of employees. However, the data for this modeling problem comes from HR Information Systems (HRIS); these are typically under-funded compared to the Information Systems of other domains in the organization which are directly related to its priorities. This leads to the prevalence of noise in the data that renders predictive models prone to over-fitting and hence inaccurate. This is the key challenge that is the focus of this paper, and one that has not been addressed historically. The novel contribution of this paper is to explore the application of Extreme Gradient Boosting (XGBoost) technique which is more robust because of its regularization formulation. Data from the HRIS of a global retailer is used to compare XGBoost against six historically used supervised classifiers and demonstrate its significantly higher accuracy for predicting employee turnover.**

***Keywords—turnover prediction; machine learning; extreme gradient boosting; supervised classification; regularization***

## I. Introduction

The problem of employee turnover has shot to prominence in organizations because of its negative impacts on issues ranging from work place morale and productivity, to disruptions in project continuity and to long term growth strategies. One way organizations deal with this problem is by predicting the risk of attrition of employees using machine learning techniques thus giving organizations leaders and Human Resources (HR) the foresight to take pro-active action for retention or plan for succession. However, the machine learning techniques historically used to solve this problem fail to account for the noise in the data in most HR Information Systems (HRIS). Most organizations have not prioritized investments in efficient HRIS solutions that would capture an employee's data during his/her tenure. One of the major factors is the limited understanding of benefits and cost. It is still difficult to measure the return of investment in HRIS [1]. This leads to noise in the data, which in turn attenuates the generalization capability of these algorithms.

In this paper, the problem of employee turnover and the key machine learning algorithms that have been used to solve it are discussed. The novel contribution of this paper is to explore the application of extreme gradient boosting (XGBoost) as an improvement on these traditional algorithms, specifically in its ability to generalize on noise-ridden data which is prevalent in this domain. This is done by using data from the HRIS of a global retailer and treating the attrition problem as a classification task and modeling it using supervised techniques. The conclusion is reached by contrasting the superior accuracy of the XGBoost classifier against other techniques and explaining the reason for its superior performance.

This paper is structured as follows. Section II gives a brief overview of the employee turnover problem, the importance of solving it, and the historical work done in terms of application of machine learning techniques to solve this problem. Section III explores the 7 different supervised techniques, including XGBoost, that this paper compares. Section IV outlines the experimental design in terms of the characteristics of the dataset, pre-processing, cross-validation, and the choice of metrics for accuracy comparison. Section V showcases the results of the study and its subsequent discussion. Section VI concludes the paper by recommending the XGBoost classifier for predicting turnover.

## II. Literature Review on Employee Turnover

Employee turnover can be interpreted as a leak or departure of intellectual capital from the employing organization [2]. Most of the literature around turnover categorizes turnover as either voluntary or involuntary.

This analysis is centered on voluntary turnover. In a meta-analytic review of voluntary turnover studies [3], it was found that the strongest predictors for voluntary turnover were age, tenure, pay, overall job satisfaction, and employee's perceptions of fairness. Other similar research findings suggested that personal or demographic variables, specifically age, gender, ethnicity, education, and marital status, were important factors in the prediction of voluntary employee turnover [4], [5], [6], [7], [8]. Other characteristics that studies focused on are salary, working conditions, job satisfaction, supervision, advancement, recognition, growth potential, burnout etc. [9], [10], [11], [12].

High turnover has several detrimental effects on an organization. It is difficult to replace employees who have niche skill sets or are business domain experts. It affects ongoing work and productivity of existing employees. Acquiring new employees as replacement has its own costs like hiring costs, training costs etc. Also, new employees will have their learning curves towards arriving at similar levels of

technical or business expertise as a seasoned internal employee.

Organizations tackle this problem by applying machine learning techniques to predict turnover thus giving them the vision to take necessary action. Table 1 below briefly documents the literature review findings. Subsequent sections of the paper will highlight the inadequacy of the classifiers recommended here in handling noise of the scale in HRIS.

TABLE I. RELATED WORK ON TURNOVER PREDICTION

| **Research Authors** | **Problem studied** | **Data Mining Techniques studied** | **Recommend** |
|---|---|---|---|
| Jantan, Hamdan and Othman [13] | Data Mining techniques for performance prediction of employees | C4.5 decision tree, Random Forest, Multilayer Perceptron(MLP) and Radial Basic Function Network | C4.5 decision tree |
| Nagadevara, Srinivasan and Valk [14] | Relationship of withdrawal behaviors like lateness and absenteeism, job content, tenure and demographics on employee turnover | Artificial neural networks, logistic regression, classification and regression trees (CART), classification trees (C5.0), and discriminant analysis) | Classification and regression trees (CART) |
| Hong, Wei and Chen [15] | Feasibility of applying the *Logit* and *Probit* models to employee voluntary Turnover predictions. | Logistic regression model (logit), probability regression model (probit) | Logistic regression model (logit) |
| Marjorie Laura Kane-Sellers [16] | To explore various personal, as well as work variables impacting employee voluntary turnover | Binomial logit regression | Binomial logit regression |
| Alao and Adeyemo [17] | Analyzing employee attrition using multiple decision tree algorithms | C4.5, C5, REPTree, CART | C5 decision tree |
| Saradhi and Palshikar [18] | To compare data mining techniques for predicting employee churn | Naïve Bayes, Support Vector Machines, Logistic Regression, Decision Trees and Random Forests | Support Vector Machines |

## III. METHODS

In machine learning, classification has two distinct meanings. We may be given a set of observations with the aim of establishing the existence of classes or clusters in the data. Or we may know for certain that there are a certain number of classes, and the aim is to establish a rule(s) whereby we can classify a new observation into one of the existing classes. The former type is known as Unsupervised Learning, the latter as Supervised Learning [19]. This paper deals with classification as supervised learning, because the data contains 2 classes – active and terminated. This section details the theory behind various classification algorithms compared.

### A. Logistic Regression

Logistic regression/ maximum entropy classifier is one of the basic linear models for classification. Logistic regression is a specific category of regression best used to predict for binary or categorical dependent variables. It's often used with regularization in the form of penalties based on L1-norm or L2-norm to avoid over-fitting. An L2-regularized logistic regression for this paper. This technique obtains the posterior probabilities by assuming a model for the same and estimates the parameters involved in the assumed model. The form of the model is given below in (1):

$$p(\text{churn}|w) = \frac{1}{1+e^{-\left[w0+\sum_{i=1}^{N} wiXi\right]}} \quad (1)$$

The parameters w, are estimated using maximum likelihood estimation technique [20]

### B. Naïve Bayesian

Naïve Bayes is a popular classification technique that has attracted attention for its simplicity and performance [21]. Naïve Bayes performs classification based on probabilities arrived, with a base assumption that all variables are conditionally independent of each other. To estimate the parameters (means and variances of the variables) necessary for classification, the classifier requires only a small amount of training data. It also handles real and discrete data [22].

The underlying logic to using the Bayes' rule for machine learning is as follows: To train a target function fn: $X \rightarrow Y$, which is the same as, $P(Y|X)$, we use the training data to learn estimates of $P(X|Y)$ and $P(Y)$. Using these estimated probability distributions and Bayes' rule new X samples could then be classified [21].

### C. Random Forest

Random Forest algorithm is a popular tree based ensemble learning technique. The type of 'ensembling' used here is bagging. In bagging, successive trees do not depend on earlier trees — each is independently constructed using a different bootstrap sample of the data set. In the end, a simple majority vote is taken for prediction. Random forests are different from standard trees in that for the latter each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node [23]. This additional layer of randomness makes it robust against over-fitting [24].

### D. K-Nearest Neighbor (KNN)

The intuition behind Nearest Neighbor Classification is to classify data points based on the class of their nearest neighbors. It is often useful to take more than one neighbor into account so the technique is more commonly referred to as k-Nearest Neighbor (k-NN) Classification [25].

The 2 stages for classification using KNN involve determining neighboring data points and then deciding the class based on the classes of these neighbors. The neighbors can be determined using distance measures like Euclidean

distance (used in this paper), Manhattan distance etc. The class can be decided on majority vote of neighbors or weighting inversely proportional to the distance. The data was scaled to [0, 1] range before building the KNN based model.

### *E. Linear Discriminant Analysis (LDA)*

Discriminant analysis involves creating one or more discriminant functions so as to maximize the variance between the categories relative to the variance with the categories [14]. Linear Discriminant Analysis is explained as deriving a variate or z-score, which is a linear combination of two or more independent variables that will discriminate best between two (or more) different categories or groups.

The z-scores calculated using the discriminant functions is then used to estimate the probabilities that a particular member or observation belongs to a class. An important point to note with LDA is that the features used should be continuous or metric in nature.

### *F. Support Vector Machine (SVM)*

An SVM is a supervised learning algorithm that implements the principles of statistical learning theory [26] and can solve linear as well as nonlinear binary classification problems. A support vector machine constructs a hyper-plane or set of hyper-planes in higher dimensional space for achieving class separation. The intuition here is that a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class- the larger the margin the lower the generalization error of the classifier. For this reason, it is also referred to as maximum margin classifier. The data was scaled to [0, 1] range before building this model.

### *G. Extreme Gradient Boosting (XGBoost)*

Boosting refers to the general problem of producing a very accurate prediction rule by combining rough and moderately inaccurate rules-of-thumb [27]. This involves fitting a sequence of weak learners on modified data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modification at each step consists of assigning higher weights to the training examples that were misclassified in the previous iteration. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. This forces the weak learner to concentrate on the examples that are missed by its predecessor.

XGBoost is a boosted tree algorithm. It follows the principle of gradient boosting [28]. Compared to other gradient boosted machines, it uses a more regularized-model formalization to control over-fitting, which gives it better performance. What we need to learn are the functions $f_i$, with each containing the structure of the tree and the leaf scores [29]. This can be formalized as seen in (2):

$$f_t(x)=w_{q(x)},\ w\in R^T,\ q:R^d\rightarrow\{1,2,\cdots,T\} \tag{2}$$

Where 'w' is the vector of scores on leaves, 'q' is a function assigning each data point to the corresponding leaf and 'T' is the number of leaves. The model complexity is formulated as:

$$\Omega(f)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \tag{3}$$

The objective function at the $t^{th}$ iteration is as seen in (4):

$$Obj^{(t)}=\sum_{j=1}^{T}[G_j w_j+\frac{1}{2}(H_j+\lambda)\,w_j^2]+\gamma T \tag{4}$$

Solving this quadratic (4), the best $w_j$ for a given structure q(x) and the best objective reduction we can get is:

$$w_j^*=-\frac{G_j}{H_j+\lambda} \tag{5}$$

$$Obj^*=-\frac{1}{2}\sum_{j=1}^{T}\frac{sqr(G_j)}{H_j+\lambda}+\gamma T \tag{6}$$

The score gained by splitting a leaf into 2 leaves is as seen in (7):

$$Gain=\frac{1}{2}\left[\frac{sqr(G_L)}{H_L+\lambda}+\frac{sqr(G_R)}{H_R+\lambda}-\frac{sqr(G_L+G_R)}{H_L+H_R+\lambda}\right]-\gamma \tag{7}$$

Where: $G_j=\Sigma_{i\in Ij}\ g_i$ and $H_j=\Sigma_{i\in Ij}\ h_i$ ; the definitions of which are as per [29].

## IV. Exprerimental Design

The population under study was a particular level of stores leadership team of a global retailer over an 18 months period. The population chosen is distributed across various locations in the US. The data was pulled at a Quarterly level. There are 2 Class labels - Active and Terminated labeled 0 and 1 respectively. Each employee would have a record for every quarter of being active in the organization, until the quarter of turnover (if it occurs), at which time the data point changes class label from active to terminated. The dataset had 73,115 data points with each labeled active or terminated.

The features for the dataset were chosen based on the studies referenced in section II. The data was gathered from 2 sources: the HRIS database of the organization, as well as the BLS (Bureau of Labor Statistics). The HRIS database of the organization provided some key features like demographics features e.g. age etc.; compensation related features like pay etc.; team related features like peer attrition etc. The BLS data provided key features like unemployment rate, median household income etc.

Overall there were 33 features of which 27 were numeric while 6 were categorical in nature.

### *A. Data pre-processing*

For categorical variables the missing values were imputed using the mode of that field. For numerical variables, missing values were imputed on a case-to-case basis. Zero-imputation was done on fields like number of promotions to prevent inflating data around employee promotions. Domain knowledge directed the imputation of certain numeric fields. For instance time since last promotion was imputed using tenure-in-position, as was known to be a good approximation. Certain other numeric variables were median-imputed as it handles the presence of outliers unlike mean imputation. As part of the data preparation, the categorical features were One-Hot Encoded, by which each of the distinct values in the categorical fields was converted to binary fields.

### B. Model validation technique

The dataset was split 80:20 into training and hold out sets. A grid-search was performed over tuning parameters, including regularization or penalty hyper-parameters, for each algorithm. The optimal configuration of hyper-parameters for each algorithm was chosen based on a 10-fold cross validation on the training set. The models were trained using their optimal-configuration on the training dataset. The trained model from each algorithm was then used to predict and test on the 20% holdout sample.

### C. Evaluation criteria for model(s)

The Area under the receiver operating characteristic curve (ROC-AUC) is the measure chosen here to compare classification accuracies. The AUC is a general measure of 'predictiveness' and decouples classifier assessment from operating conditions i.e., class distributions and misclassification costs [30]. Furthermore, AUC is preferable over alternative indicators like, e.g., error-rate because it measures the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one, which is equivalent to the Wilcoxon test of ranks [31].

Additionally, model run time and memory utilization are also used to compare the performance of the classifiers. These 2 measures are important to report on, as they build a case from a practitioner's perspective on determining what algorithm is good to implement for real-life business problems, solving for scalability and performance.

### D. System specification

All classifiers except XGBoost are used from the scikit-learn package in Python 2.7. XGBoost classifier was used from the XGBoost package. The codes were run on a 16 GB MacBook OS 10.10.5 version.

## V. Results

TABLE II. Model Results

| Algorithm | AUC (Training) | AUC (Holdout) | Run-time (Training) | Maximum Memory Utilization (Of 16 GB) |
|---|---|---|---|---|
| XGBoost | **0.88** | **0.86** | 16 min 12 sec | **12%** |
| Logistic Regression | 0.66 | 0.50 | **52 sec** | 20% |
| Naïve Bayesian | 0.64 | 0.59 | 59 sec | 20% |
| Random Forest (Depth controlled) | 0.79 | 0.51 | 23 min 10 sec | 29% |
| SVM (RBF kernel) | 0.68 | 0.52 | 105 min 30 sec | 21% |
| LDA | 0.74 | 0.52 | 6 min 51 sec | 35% |
| KNN (Euclidean distance) | 0.52 | 0.5 | 180 min 12 sec[a] | 35% |

[a] *Since KNN is a lazy learner, we are measuring the run time till final output for this model*

### A. Lift Charts

The output obtained as the prediction is the probability of attrition, which is then converted to a risk ranking of employees. The model was further validated by checking the performance of each risk decile by means of a lift chart as depicted in Figure 1. A Lift Chart visualizes the improvement that a particular model provides when compared against a random guess.

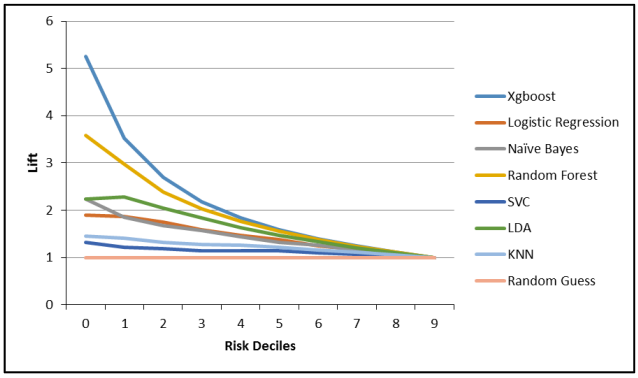



Fig. 1. Lift Chart for the Classifiers

It can be gauged from figure 1 that the XGBoost model has better decile performance than other models till the 7th decile (inclusive). It is also consistently and considerably better than a random guess.

### B. Discussion

The population in this dataset is representative of a workforce that is distributed across the United States, comprising of people at different stages of their careers, different levels of performance and pay, and from different backgrounds. Hence, it's intuitive to assume that a rule based approach or a tree-based model will most likely perform best, considering the various themes and groups naturally occurring in the data. This intuition is validated by the observations in Table 2. It is seen that the two tree-based classifiers in Random Forest and XGBoost performs better than the other classifiers during training and that XGBoost is significantly better than Random Forest during testing. The XGBoost classifier outperforms the other classifiers in terms of accuracy and memory utilization.

Algorithmically, Random Forests trusts its stages of randomization to help it achieve better generalization but as is seen from the table it's still insufficient to prevent over-fitting in this case. On the other hand the XGBoost tries to add new trees that compliments the already built ones. Boosting serves to improve training for the difficult to classify data points. Another important point is the over-fitting suffered by classifiers other than XGBoost despite regularization or introduction of randomness, as the case maybe. XGBoost overcomes this problem due to its excellent inherent regularization (as shown mathematically in Section III, G) and hence works perfectly for the noisy data from the HRIS.

The XGBoost classifier is also optimized for fast, parallel tree construction, and designed to be fault tolerant under the distributed setting [29]. XGBoost classifier takes data in the form of DMatrix. DMatrix is an internal data structure used by

XGBoost which is optimized for both memory efficiency and training speed. Here, DMatrixes were constructed from numpy arrays of the features and the classes.

## VI. Conclusions and Future Work

The importance of predicting employee turnover in organizations and the application of machine learning in building turnover models was presented in this paper. The key challenge of noise in the data from HRIS that compromises the accuracy of these predictive models was also highlighted. Data from the HRIS of a global retailer was used to compare the XGBoost classifier against six other supervised classifiers that had been historically used to build turnover models. The results of this research demonstrate that the XGBoost classifier is a superior algorithm in terms of significantly higher accuracy, relatively low runtimes and efficient memory utilization for predicting turnover. The formulation of its regularization makes it a robust technique capable of handling the noise in the data from HRIS, as compared to the other classifiers, thus overcoming the key challenge in this domain. Because of these reasons it is recommended to use XGBoost for accurately predicting employee turnover, thus enabling organizations to take actions for retention or succession of employees.

For future studies, the authors recommend the capture of data around interventions done by the organization for at-risk at employees and its outcome. This will transform the model into a prescriptive one, addressing not just the question "Who is at risk?" but also "What can we do?". It is also recommended to study the application of deep learning models for predicting turnover. A well-designed network with sufficient hidden layers might improve the accuracy, however the scalability and practical implementation aspect has to be studied as well.

## References

[1] S. Jahan, "Human Resources Information System (HRIS): A Theoretical Perspective", Journal of Human Resource and Sustainability Studies, Vol.2 No.2, Article ID:46129, 2014.

[2] M. Stoval and N. Bontis, "Voluntary turnover: Knowledge management – Friend or foe?", Journal of Intellectual Capital, 3(3), 303-322, 2002.

[3] J. L. Cotton and J. M. Tuttle, "Employee turnover: A meta-analysis and review with implications for research", Academy of management Review, 11(1), 55-70, 1986.

[4] L. M. Finkelstein, K. M. Ryanand E.B. King, "What do the young (old) people think of me? Content and accuracy of age-based metastereotypes", European Journal of Work and Organizational Psychology, 22(6), 633-657, 2013.

[5] B. Holtom, T. Mitchell, T. Lee, and M. Eberly, "Turnover and retention research: A glance at the past, a closer review of the present, and a venture into the future", Academy of Management Annals, 2: 231-274, 2008

[6] C. von Hippel, E. K. Kalokerinos and J. D. Henry, "Stereotype threat among older employees: Relationship with job attitudes and turnover intentions", Psychology and aging, 28(1), 17, 2013.

[7] S. L. Peterson, "Toward a theoretical model of employee turnover: A human resource development perspective", Human Resource Development Review, 3(3), 209-227, 2004.

[8] J. M. Sacco and N. Schmitt, "A dynamic multilevel model of demographic diversity and misfit effects", Journal of Applied Psychology, 90(2), 203-231, 2005.

[9] D. G. Allen and R. W. Griffeth, "Test of a mediated performance – Turnover relationship highlighting the moderating roles of visibility and reward contingency", Journal of Applied Psychology, 86(5), 1014-1021, 2001.

[10] D. Liu, T. R. Mitchell, T. W. Lee, B. C. Holtom, and T. R. Hinkin, "When employees are out of step with coworkers: How job satisfaction trajectory and dispersion influence individual-and unit-level voluntary turnover", Academy of Management Journal, 55(6), 1360-1380, 2012.

[11] B. W. Swider, and R. D. Zimmerman, "Born to burnout: A meta-analytic path model of personality, job burnout, and work outcomes", Journal of Vocational Behavior, 76(3), 487-506, 2010.

[12] T. M. Heckert and A. M. Farabee, "Turnover intentions of the faculty at a teaching-focused university", Psychological reports, 99(1), 39-45, 2006.

[13] H. Jantan, A. R. Hamdan, and Z. A. Othman, "Towards Applying Data Mining Techniques for Talent Managements", 2009 International Conference on Computer Engineering and Applications, IPCSIT vol.2, Singapore, IACSIT Press, 2011.

[14] V. Nagadevara, V. Srinivasan, and R. Valk, "Establishing a link between employee turnover and withdrawal behaviours: Application of data mining techniques", Research and Practice in Human Resource Management, 16(2), 81-97, 2008.

[15] W. C. Hong, S. Y. Wei, and Y. F. Chen, "A comparative test of two employee turnover prediction models", International Journal of Management, 24(4), 808, 2007.

[16] L. K. Marjorie, "Predictive Models of Employee Voluntary Turnover in a North American Professional Sales Force using Data-Mining Analysis", Texas, A&M University College of Education, 2007.

[17] D. Alao and A. B. Adeyemo, "Analyzing employee attrition using decision tree algorithms", Computing, Information Systems, Development Informatics and Allied Research Journal, 4, 2013.

[18] V. V. Saradhi and G. K. Palshikar, "Employee churn prediction", Expert Systems with Applications, 38(3), 1999-2006, 2011.

[19] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, Machine Learning, Neural and Statistical Classification. Ellis Horwood Limited, 1994.

[20] G. King and L. Zeng, "Logistic regression in rare events data", Political Analysis, 9(2), 137–163, 2001.

[21] T. Mitchell, Machine learning. 2nd ed. USA: McGraw Hill, 1997.

[22] H. A. Elsalamony (2014), "Bank direct marketing analysis of data mining techniques", International Journal of Computer Applications, 85(7).

[23] A. Liaw and M. Wiener, "Classification and regression by randomForest", R news, 2(3), 18-22, 2002.

[24] L. Breiman, Random forests. Machine Learning, 45(1), 5–32, 2001.

[25] P. Cunningham and S. J. Delany, "k-Nearest neighbour classifiers", Multiple Classifier Systems, 1-17, 2007.

[26] C. Cortes and V. Vapnik, Support-vector networks. Machine learning, 20(3), 273-297, 1995.

[27] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", Journal of computer and system sciences, 55(1), 119-139, 1997.

[28] J. H. Friedman, "Greedy function approximation: a gradient boosting machine", Annals of statistics, 1189-1232, 2001.

[29] T. Chen and C. Guestrin, "XGBoost: Reliable Large-scale Tree Boosting System, 2015", Retrieved from http://learningsys.org/papers/LearningSys_2015_paper_32.pdf. Accessed 12 December 2015.

[30] S. Lessmann and S. Voß, "A reference model for customer-centric data mining with support vector machines", European Journal of Operational Research 199, 520–530, 2009.

[31] T. Fawcett, "An introduction to ROC analysis", Pattern Recognition Letters 27 (8), 861–874, 2006.

# WSDF: Weighting of Signed Distance Function for Camera Motion Estimation in RGB-D Data

Pham Minh Hoang, Vo Hoai Viet, Ly Quoc Ngoc
Department of Computer Vision and Robotics, University Of Science, VNU-HCM, Viet Nam

***Abstract*—With the recent advent of the cost-effective Kinect, which can capture real-time high-resolution RGB and visual depth information, has opened an opportunity to significantly increase the capabilities of many automated vision based recognition including object/action classification, 3D reconstruction, etc… In this work, we address the camera motion estimation which is an important phase in 3D object reconstruction system based on RGB-D data. We segment objects by thresholding algorithm based on depth data and propose the weighting function for SDF that is called WSDF. The problem of minimizing of this function is solved by Gauss-Newton methods. We systematically evaluate our method on TUM dataset. The experimental results are measured by ATE and RPE that evaluate both global and local consistency of camera motion estimation algorithm. We demonstrate large improvements over the state-of-the-art methods on both plant and teddy3 objects and achieve the best ATE as 0.00564 and 0.0182 and the best RPE as 0.00719 and 0.00104, respectively. These experiments show that the proposed method significantly outperforms state-of-the-art techniques.**

***Keywords*—RGB-D data; 3D Reconstruction; SDF; Camera Motion Estimation**

## I. Introduction

Reconstructing 3D object is an interesting and challenging problem in computer vision. It has attracted many research efforts from the computer vision community in recent decades for its high potential applications such as game, SLAM, medical technology, virtual reality, and robotics. Due to its wide range of applications, 3D object reconstruction has attracted much attention in recent years [2]. Generally speaking, 3D object reconstruction framework contains three main steps namely object segmentation, camera motion estimation, and surface reconstruction (see in Fig. 2). Object segmentation is to identity the object region in images that can achieve by using the algorithms such as kmean, mean shift, ostu ... Camera motion estimation aims to represent the movement of object over frames. The result of this phase is point cloud that describe object in 3D space. Surface reconstruction focus on reconstructing the surface mesh… In this work, we only focus the problem of the camera motion estimation phase. We use the Ostu and thresholding algorithm for object segmentation.

The advent of affordable RGB-D sensors has opened up a whole new range of applications based on the 3d perception of the environment by computers, which includes the creation of a virtual 3d representation of real objects. Compared with conventional color data, depth maps provide several advantages, such as the ability of reflecting pure geometry and shape cues, or insensitive to changes in lighting conditions. Moreover, the range sensor provides 3D structural information of the scene and objects. These characteristics will be helpful for object segmentation and camera motion estimation.

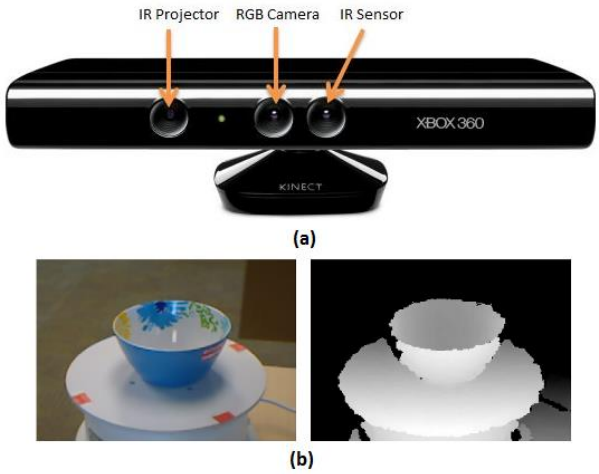



Fig. 1. Illustration of 3D camera and RGB-D data: a) Microsoft Kinect Device; b) an object example of RGB-D data is captured by Kinect

In this manuscript, we proposed the weighting parameters for SDF that was proposed at [4, 5] to improve the performance of camera motion of 3D reconstruction system based on RGB-D data. The main contributions of this paper are summarized as follows: Firstly, we apply the weighting approach for SDF for camera motion estimation based on RGB-D data. Secondly, we systematically evaluate our WSDF on four challenging datasets.

The rest of this paper is organized as follows: Section II gives a concise review of existing works on camera motion estimation for 3D reconstruction. Section III presents signed distance function for camera motion estimation. Section IV introduces our improvement for camera motion estimation. Section V presents action classification. Section V shows the experiment results on relevant benchmarks. Finally, section VI draws conclusions of our work and indicates future studies.

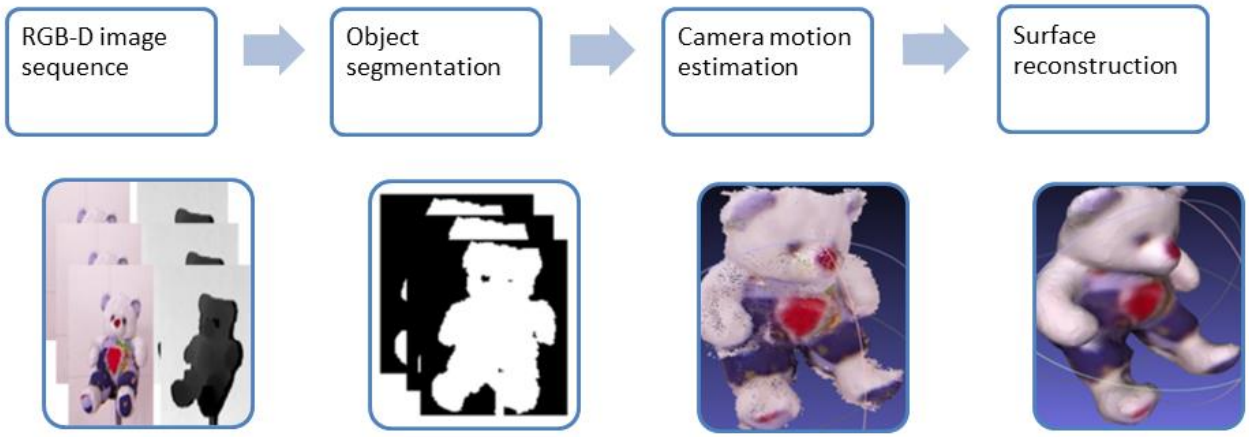



Fig. 2. Flowchart of 3D object reconstruction system in RGB-D data

## II. Literature Review

Comprehensive reviews of the previous studies can be found in [2]. Our discussion in this section is restricted to a few influential and relevant parts of literature, with a focus on camera motion estimation based on RGB-D data.

The camera motion estimation aims to find the affine transformations to convert point clouds in local frames into global coordination and integrate them into a final point cloud for object representation. These transformations represent the movement of camera from the first frame to the last frame. The earliest approaches focus on finding the affine transformation between two consecutive frames. In [13], the author use ICP algorithm to find affine between two consecutive frames based on the features are extracted from them. Another famous method are called Kinect Fusion [10, 11], the method build the Signed Distance Function (SDF) and use the function for initializing the point cloud for each frame. Then, ICP algorithm is used to find affine transform in the next frame. However, the integration of affine transformations between two consecutive frames makes the errors that accumulated to misleading in the following frame is greater. The difference from Kinect Fusion, these methods in [7, 10] estimate directly the affine transformation by minimizing the RSME of SDF, then updating SDF based on the computed transformation. In [8], the authors build SDF based on Octree to reduce memory and computational cost. These methods that use ICP algorithm focus on minimizing the point cloud, some methods [3 , 4 , 5, 6, 10] minimize the RGB-D of SDF between two consecutive frames. In [9], the method finds corresponding points between two consecutive frames and minimizes the total of the distance of these corresponding points.

In this paper, we propose the camera motion estimation based on SDF in [5, 6]. However, we improve SDF by adding the weighting function in [3] that is called WSDF. And, the problem of minimizing for this function is solved by Gauss-Newton method.

## III. Background of Camera Motion Estimation

In this session, we present the camera motion estimation over frames from RGB-D sequences. The inputs of this phase are local point clouds are extracted from RGB and depth of each frame $P_i = \{x_j\}$ with $x_j$ is 3D vertex of point cloud $P_i$. The problem is to find affine transformation to transfer the local point cloud at i-th frame from local coordinate to global coordinate. The affine transformation also describes motion of camera over frames, so this phase is called camera motion estimation. In [4, 5], Bylow et al. introduced the method of camera motion estimation based on signed distance function (SDF).

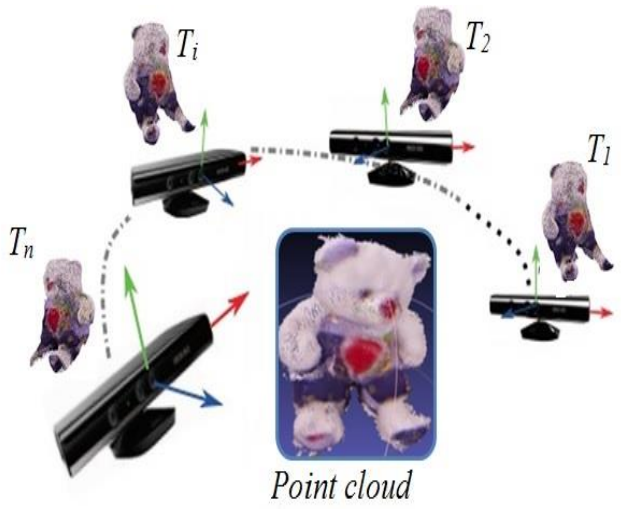



Fig. 3. An example of camera motion estimation

### A. Signed Distance Function

The SDF of given surface $\theta(x)$: $R^3 \rightarrow R$. This function returns for any point $x \in R^3$ the signed distance from x to the surface. The SDF have four properties as follows:

- If $x$ is outside the surface then $\theta(x) > 0$.
- If $x$ is inside the surface then $\theta(x) < 0$.
- If $x$ is on the surface then $\theta(x) = 0$.
- If $x$ is nearer the surface then $\theta(x)$ is smaller.

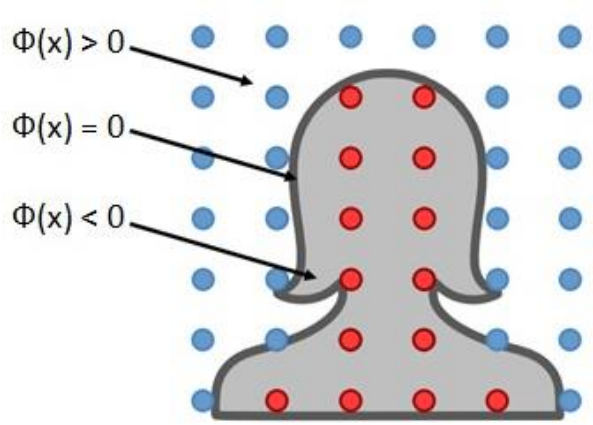



Fig. 4. Illustration of SDF for object's surface

*B. Affine Transformation*

An affine transformation consists of two components: a three-dimension square matrix $R_i$ and a three-dimension translation vector $t_i$. We assume that we already have the surface of object represented by a signed distance function (SDF). For each vertex of point cloud in local coordinate, our goal is the transformed point lies as close as possible to the object surface. It means $[\theta(R_i x_j + t_i)]^2$ is as smaller as possible. We must find $R_i$ and $t_i$ such that the function $E(R_i, t_i) = \sum_i [\theta(R_i x_j + t_i)]^2$ is minimized.

Considering the function $T_i = [R_i | t_i]$ consists of 12 parameters. However, the limitation of problem only needs the rotation and translation that can be solved by 6 parameters with three parameters for rotation $(\omega_1, \omega_2, \omega_3)$ and three parameters for translation $(t_1, t_2, t_3)$. Therefore, $T_i$ can be written as a vector of 6 dimensions $\xi_i = (\omega_1, \omega_2, \omega_3, t_1, t_2, t_3)$ and $E(R_i, t_i)$ is also written as $E(\xi_i) = \sum_i [\theta_j(\xi_i)]^2$. To minimize this function, Bylow et al. [4, 5] used Gauss-Newton algorithm.

*C. Update the SDF and the colors*

The SDF is not traditional formula function due to it is formed by dividing the space into grids in 3D. Each node in 3D grid is called voxel. If a point does not match to voxel, SDF value of x is obtained based SDF value of the nearest neighbor voxels. So, the objective in this step is to compute SDF for each voxel.

Assume that $v^G$ is global coordinate of each voxel. Based on the estimated pose $T_i$, we can transfer to local coordinate of frame i as $v^L = R^T(v^G - t)$. According to camera model, with the focal lengths $f_x$ and $f_y$ and principal point $(c_x, c_y)$,we can project 3D point $v^L = (v_x^L, v_y^L, v_z^L)$ to image plane by projection

$$\pi(x, y, z) = \left( \frac{f_x x}{z} + c_x, \frac{f_y y}{z} + c_y \right)$$

Let (i, j) be pixel coordinate of projected point $v^L$ in image and I(d) be the corresponding depth value at $(i, j)$. We can compute distance $d(v^L)$ of the depth of voxel and the depth value at $(i, j)$.

$$d(v^L) = z - I_d(i, j)$$

Since the distance $d(v^L)$ is a rough approximation which can get arbitrary wrong, we follow the standard approach to reduce the impact of bad measurements by truncating the measured distance if $|d| > \delta$ for some threshold $\delta$ as follows:

$$d = \begin{cases} -\delta & if \quad d < -\delta \\ d & if \quad |d| \le \delta \\ \delta & if \quad d > \delta \end{cases}$$

For each frame, we can compute the distance $d_i$ of each voxel at frame $i^{th}$. The SDF value of a voxel can be obtained by weighted average of these distances as follows:

$$\theta(v) = \frac{\sum_i w_i d_i}{w_i}$$

However, this is not enough to decrease the impact of bad measurements. We do also have a higher uncertainty when the voxel lies behind the surface. To handle this, we weight the measurements using the following weight function as follows:

$$w(d) = \begin{cases} 1 & if \quad d < \varepsilon \\ e^{-\sigma(d-\varepsilon)^2} & if \quad d \ge \varepsilon \ and \ d \le \delta \\ 0 & if \quad d > \delta \end{cases}$$

Therefore, we can update SDF of each voxel as follows:

$$\theta = \frac{W\theta + w_i d_i}{W + w_i}$$

$$W = W + w_i$$

From the RGB image and each voxel the color is estimated as the formula as follows:

$$R \rightarrow \frac{W^c R + w_i^c r}{W^c + w_i^c}$$

$$G \rightarrow \frac{W^c G + w_i^c g}{W^c + w_i^c}$$

$$B \rightarrow \frac{W^c B + w_i^c b}{W^c + w_i^c}$$

Where $w_i^c$ the weight of color for new measurement, $w_i^c$ is used as $w_i^c = w_i cos\alpha$ where $\alpha$ is the angle between the ray

and the principal axis to give more weight to pixels whose normal is pointing towards the key frame.

## IV. Weighting of Signed Distance Function

To increase the accuracy for the problem of minimize $E(\xi_i)$, we propose the weighting function $w(r_i)$ for SDF that is called WSDF where $r_i(\xi) = \theta_j(\xi_i)$. According to [3], the weighting function $w(r_i)$ is defined as follows:

$$w(r_i) = \frac{\nu + 1}{\nu + \left(\frac{r_i}{\sigma}\right)^2}$$

The points are near surface can more accurately describe the shape of the object than the points are far from surface. So, the w($r_i$) will increase when $r_i$ increase, this means the weights of the points are near the surface will be higher than the weight of the points are far from the surface. Meanwhile, we have to find $\xi_i$ by solving the optimization of the non-linear function $\xi_i = \sum_i \text{argmin}(w(r_i)(r_i(\xi))^2)$. We apply the Gauss-Newton method to solve the problem. The initialization for $\xi = \xi^{(0)}$, and $\xi$ at each loop is computed by the following formula: $\xi^{(k+1)} = \xi^{(k)} - (J^T W J)^{-1} J^T W r(\xi^{(k)})$ where J is Jacobian matrix $J = \left[\frac{\partial r}{\partial \omega_1} \frac{\partial r}{\partial \omega_2} \frac{\partial r}{\partial \omega_3} \frac{\partial r}{\partial t_1} \frac{\partial r}{\partial t_2} \frac{\partial r}{\partial t_3}\right]$ and $W$ is matrix that is created by main diagonal of $w(r_i)$. The loop will end when $\| \xi^{(k+1)} - \xi^{(k)} \|_\infty$ is enough small or the number of loop achieve the limitation. We adopt $\nu = 5$ based on the experiment, $\sigma^2$ at each loop is computed as follows:

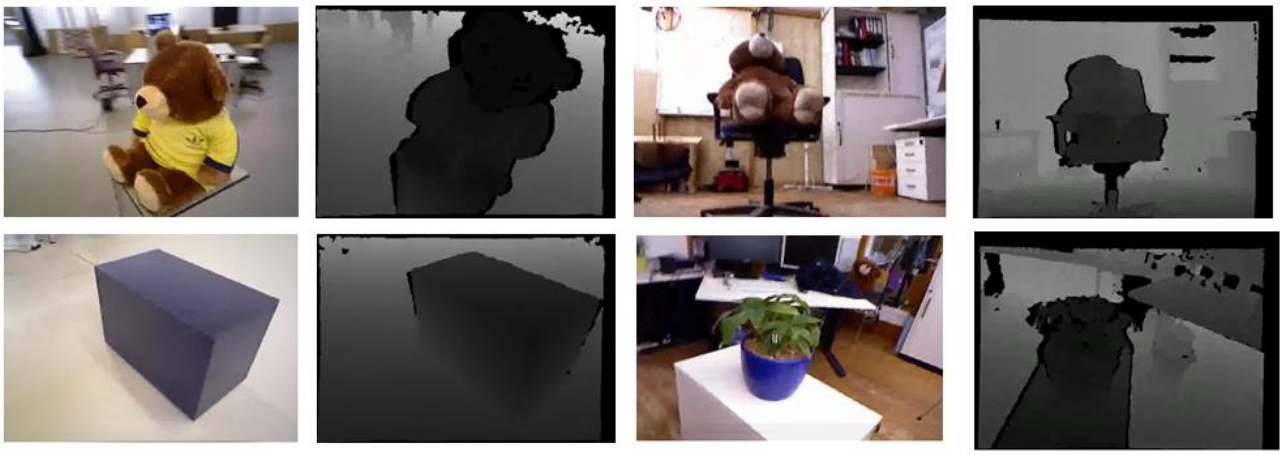


Fig. 5. Some RGB and depth frames from TUM dataset

$$\sigma^2 = \frac{1}{n}\sum_i r_i^2 \frac{\nu + 1}{\nu + \left(\frac{r_i}{\sigma}\right)^2}$$

The end of the process, we have $T_i$ is computed by a vector of 6 dimensions of $\xi_i$. Then, we update SDF to compute for the next frame.

## V. Expriment Results

### A. Dataset

We also evaluated our approach on the TUM 3D object reconstruction RGB-D benchmark dataset [12]. In this wok, we use plant and teddy 3 to measure the errors of our approach. Fig. 5 shows some examples of the TUM dataset.

### B. Measurement Evaluation

*1) Relative pose error (RPE)*

The relative pose error [8] measures the local accuracy of the trajectory over a fixed time interval Δ. Therefore, the relative pose error corresponds to the drift of the trajectory which is in particular useful for the evaluation of visual odometry systems. We define the relative pose error at time step i as follow:

$$E_i = (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta})$$

From a sequence of n camera poses, we obtain in this way m = n − Δ individual relative pose errors along the sequence. From these errors, we propose to compute the root mean squared error (RMSE) over all-time indices of the translational component as follows:

$$RMSE(E_{1:n}, \Delta) = \left(\frac{1}{m}\sum_{i=1}^{m} \|trans(E_i)\|^2\right)^{1/2}$$

where $trans(E_i)$ refers to the translational components of the relative pose error $E_i$.

*2) Absolute trajectory error (ATE)*

The absolute trajectory error [8] measures the global consistency can be evaluated by comparing the absolute distances between the estimated and the ground truth trajectory. As both trajectories can be specified in arbitrary coordinate frames, they first need to be aligned.

This can be achieved in closed form using the method of Horn [1], which finds the rigid-body transformation S corresponding to the least-squares solution that maps the estimated trajectory $P_{1:n}$ onto the ground truth trajectory $Q_{1:n}$. Given this transformation, the absolute trajectory error at time step i can be computed as follows:

$$F_i = Q_i^{-1} S P_i$$

Similar to the relative pose error, we propose to evaluate the root mean squared error over all time indices of the translational components as follows:

$$RMSE(E_{1:n}) = \left( \frac{1}{m} \sum_{i=1}^{m} \|trans(F_i)\|^2 \right)^{1/2}$$

where $trans(F_i)$ refers to the translational components of the relative pose error $F_i$.

### C. Experimental Results

We firstly evaluate our proposed approach on the benchmark objects in TUM dataset. Then we compare our experimental results to the-state-of-the-art methods to prove the effectiveness and robust of the proposed method.

In this research, we focus on camera motion estimation for 3D object reconstruction. Our approach based on object segmentation and SDF in RGB-D data. More specific, we use depth data for segmenting object and proposed the weighting function for SDF and solve the problem of minimizing for this function by using Gauss-Newton method. We evaluate our method by ATE and RPE that evaluate both global and local consistency. Moreover, we also evaluate many different time intervals to have deeper in understanding of the problem of camera motion estimation. Table I and II give our experimental results on plant and teddy3 objects. However, the same approach has the different result on the different objects. This is the different characteristics of these datasets. The plant object have the slow movement more than teddy3 object. In addition, teddy3 object have structure of surface more complexity than plant object.

Table III, IV, V and VI compare our experimental results with state-of-the-art results on TUM dataset. We achieve better than Bylow's approach on both plant and teddy3 object. Our method is more efficient on both global and local consistency (can see Fig. 6). These results show that our approach is robust for camera motion estimation. To have these promising results based on updating SDF with the weighting function to get more accuracy when estimate the motion between two consecutive frames.

TABLE I. EXPERIMENTIAL RESULTS ON PLANT OBJECT

| Frames | Measurement (m) | |
|---|---|---|
| | ***ATE*** | ***RPE*** |
| **10** | 0.00654 | 0.0182 |
| **20** | 0.00856 | 0.0209 |
| **30** | 0.00809 | 0.0294 |
| **40** | 0.01024 | 0.0504 |
| **50** | 0.01444 | 0.0673 |

TABLE II. EXPERIMENTIAL RESULTS ON TEDDY3 OBJECT

| Frames | Measurement (m) | |
|---|---|---|
| | ***ATE*** | ***RPE*** |
| **10** | 0.00719 | 0.00104 |
| **20** | 0.007 | 0.01152 |
| **30** | 0.00813 | 0.01387 |
| **40** | 0.01433 | 0.02149 |
| **50** | 0.0225 | 0.03348 |

TABLE III. COMPARISION WITH THE STATE OF THE ARE METHOD ON PLANT OBJECT USING ATE

| Frames | Methods | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.00654 | 0.00937 |
| **20** | 0.00856 | 0.01168 |
| **30** | 0.00809 | 0.01193 |
| **40** | 0.01024 | 0.01605 |
| **50** | 0.01444 | 0.02335 |

TABLE IV. COMPARISION WITH THE STATE OF THE ARE METHOD ON PLANT OBJECT USING RPE

| Frames | Methods | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.0182 | 0.0278 |
| **20** | 0.0209 | 0.0338 |
| **30** | 0.0294 | 0.0503 |
| **40** | 0.0504 | 0.0847 |
| **50** | 0.0673 | 0.119 |

TABLE V. COMPARISION WITH THE STATE OF THE ARE METHOD ON TEDDY3 OBJECT ON ATE

| Frames | Methods | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.00719 | 0.0114 |
| **20** | 0.007 | 0.0224 |
| **30** | 0.00813 | 0.027 |
| **40** | 0.01433 | 0.0476 |
| **50** | 0.0225 | 0.0654 |

TABLE VI. COMPARISION WITH THE STATE OF THE ARE METHOD ON TEDDY3 OBJECT ON RPE

| Frames | Methods | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.00104 | 0.0173 |
| **20** | 0.01152 | 0.0346 |
| **30** | 0.01387 | 0.0401 |
| **40** | 0.02149 | 0.0679 |
| **50** | 0.03348 | 0.0943 |

## VI. CONCLUSION

In this work, we present a novel approach for camera motion estimation based on SDF in 3D object reconstruction using RGB-D data. In order to segment object, we use depth data based on threshold method. To estimate camera motion, we proposed a weighting function is added to SDF function is called WSDF to improve the performance of camera motion estimation phase. And, the WSDF is minimized by Gauss-Newton method. We systematically evaluate our approach on benchmark dataset. The experiments are measured on both ATE and RPE that assess the global and local consistency of the camera motion estimation. The experimental results show that our proposed approach achieves superior performance to the state-of-the-art algorithm on TUM dataset.

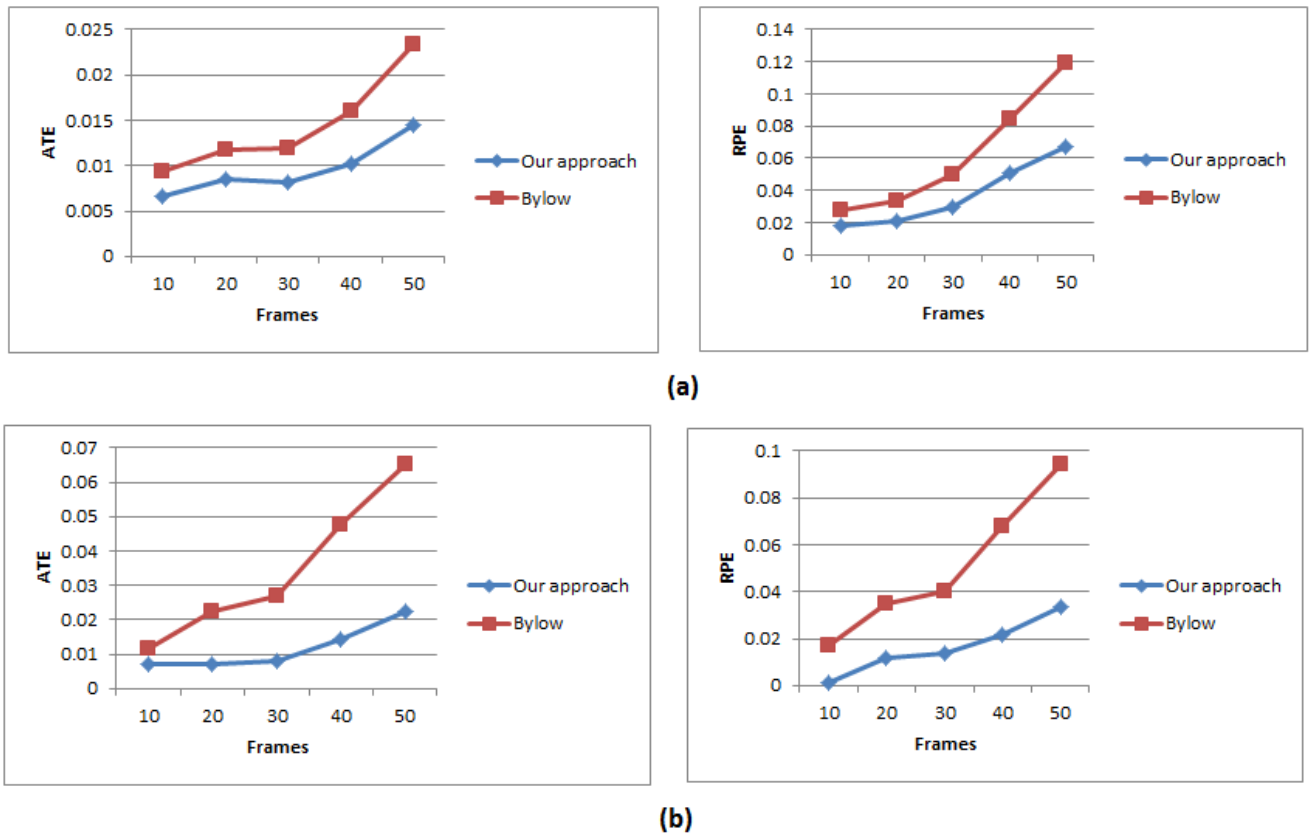



Fig. 6. Comparison with the baseline method in [4]: a) plant object; b) teddy3 object

In the future, we will consider SIFT or SIFT-flow for camera motion estimation based on RGB data to have better the performance of the system.

## ACKNOWLEDGMENT

This research is funded by Vietnam National University, Ho Chi Minh City (VNU-HCM) under grant number *B2014-18-02.*

## REFERENCES

[1] B. Horn, “Closed-form solution of absolute orientation using unit quaternions,” Journal of the Optical Society of America A, vol. 4, pp. 629–642, 1987.

[2] Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Levine, J., Sharf, A.: State of the art in surface reconstruction from point clouds. In: Proceedings of Eurographics—Eurographics star reports, vol. 1, pp. 161–185, 2014.

[3] Christian Kerl, Jurgen Sturm, and Daniel Cremers. "Robust odometry estimation for RGB-D cameras." Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.

[4] Erik Bylow, et al. "Real-time camera tracking and 3d reconstruction using signed distance functions." Robotics: Science and Systems (RSS) Conference 2013. Vol. 9. Robotics: Science and Systems, 2013.

[5] Erik Bylow, Carl Olsson, and Fredrik Kahl. "Robust Camera Tracking by Combining Color and Depth Measurements." ICPR. 2014.

[6] Fatih Calakli, and Gabriel Taubin. "SSD: Smooth signed distance surface reconstruction." Computer Graphics Forum. Vol. 30. No. 7., 2011.

[7] Fatih Calakli, and Gabriel Taubin. "SSD-C: Smooth signed distance colored surface reconstruction." Expanding the Frontiers of Visual Analytics and Visualization, pp 323-338, 2012.

[8] Jurgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers, A Benchmark for the Evaluation of RGB-D SLAM Systems, International Conference on Intelligent Robots and Systems, 2012.

[9] Qian-Yi Zhou Vladlen Koltun. "Depth camera tracking with contour cues." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[10] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim. "KinectFusion: Real-time dense surface mapping and tracking." Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. IEEE, 2011.

[11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, Andrew Fitzgibbon. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera." Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 2011.

[12] http://vision.in.tum.de/data/datasets/rgbd-dataset/download#

[13] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, Dieter Fox, “RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments”, International Journal of Robotics Research, Vol. 31, pp 647-663, 2012.