Volume 2 ▪ Issue 1 ▪ March 2011

Editor-in-Chief
Professor Walid Aref

INTERNATIONAL JOURNAL OF DATA ENGINEERING (IJDE)

ISSN : 2180-1274
Publication Frequency: 6 Issues / Year

CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF DATA ENGINEERING (IJDE)

VOLUME 2, ISSUE 1, 2011

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 2180-1274

International Journal of Computer Science and Security is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJDE Journal is a part of CSC Publishers

Computer Science Journals

http://www.cscjournals.org

INTERNATIONAL JOURNAL OF DATA ENGINEERING (IJDE)

Book: Volume 2, Issue 1, March 2011
Publishing Date: 04-04-2011
ISSN (Online): 2180-1274

IJDE Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2011

EDITORIAL PREFACE

This is first issue of volume two of the International Journal of Data Engineering (IJDE). IJDE is an International refereed journal for publication of current research in Data Engineering technologies. IJDE publishes research papers dealing primarily with the technological aspects of Data Engineering in new and emerging technologies. Publications of IJDE are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics cover by IJDE is Annotation and Data Curation, Data Engineering, Data Mining and Knowledge Discovery, Query Processing in Databases and Semantic Web etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 5, 2011, IJDE appears in more focused issues. Besides normal publications, IJDE intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of IJDE is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position IJDE as one of the top International journal in Data Engineering, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to Data Engineering fields.

IJDE editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJDE. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJDE provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts..

Editorial Board Members
International Journal of Data Engineering (IJDE)

EDITORIAL BOARD

TABLE OF CONTENTS

Volume 2, Issue 1, April 2011

Pages

Ontology based Approach for Classifying Biomedical Text Abstracts

Rozilawati Binti Dollah rozeela@kde.cs.tut.ac.jp
Dept. of Electronic and Information Engineering
Toyohashi University of Technology
Hibarigaoka, Tempaku-cho,
Toyohashi-shi, Aichi, 441-8580 Japan

Masaki Aono aono@kde.cs.tut.ac.jp
Dept. of Computer Science and Engineering
Toyohashi University of Technology
Hibarigaoka, Tempaku-cho,
Toyohashi-shi, Aichi, 441-8580 Japan

Abstract

Classifying biomedical literature is a difficult and challenging task, especially when a large number of biomedical articles should be organized into a hierarchical structure. Due to this problem, various classification methods were proposed by many researchers for classifying biomedical literature in order to help users finding relevant articles on the web. In this paper, we propose a new approach to classify a collection of biomedical text abstracts by using ontology alignment algorithm that we have developed. To accomplish our goal, we construct the OHSUMED disease hierarchy as the initial training hierarchy and the Medline abstract disease hierarchy as our testing hierarchy. For enriching our training hierarchy, we use the relevant features that extracted from selected categories in the OHSUMED dataset as feature vectors. These feature vectors then are mapped to each node or concept in the OHSUMED disease hierarchy according to their specific category. Afterward, we align and match the concepts in both hierarchies using our ontology alignment algorithm for finding probable concepts or categories. Subsequently, we compute the cosine similarity score between the feature vectors of probable concepts, in the "enriched" OHSUMED disease hierarchy and the Medline abstract disease hierarchy. Finally, we predict a category to the new Medline abstracts based on the highest cosine similarity score. The results obtained from the experiments demonstrate that our proposed approach for hierarchical classification performs slightly better than the multi-class flat classification.

Keywords: Biomedical Literature, Feature Selection, Hierarchical Text Classification, Ontology Alignment

Corresponding author. Address: Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, UTM Skudai, 81310 Johor Bahru, Johor, Malaysia.

1. INTRODUCTION

Text classification is the process of using automated techniques to assign text samples into one or more set of predefined classes [1], [2]. Nonetheless, text classification system on biomedical literature aims to select relevant articles to a specific issue from large corpora [3]. Recently, classifying biomedical literature becomes one of the best challenging tasks due to the fact that a large number of biomedical articles are divided into quite a few subgroups in a hierarchy. Many researchers have attempted to find more applicable ways for classifying biomedical literature in order to help users to find relevant articles on the web. However, most approaches used in text classification task have applied flat classifiers that ignore the hierarchical structure and treat each concept separately. In flat classification, the classifier assigns a new documents to a category based on training examples of predefined documents.

Generally, text classification can be considered as a flat classification technique, where the documents are classified into a predefined set of flat categories and no relationship specified between the categories. Singh and Nakata [4] stated that the flat classification approach was suitable when a small number of categories were defined. Nevertheless, due to the increasing number of published biomedical articles on the web, the task of finding the most relevant category for a document becomes much more difficult. Consequently, flat classification turns out to be inefficient, while hierarchical classification is more preferable. Contrary to flat classification, hierarchical classification can be defined as a process of classifying documents into a hierarchical organization of classes or categories based on the relationship between terms or categories. In hierarchical classification, a new document would be assigned to a specific category based on the concepts and relationships within the hierarchy of predefined classes. Many large text databases, such as Yahoo and Google are organized into hierarchical structure, which would help the users searching and finding relevant articles or information easier.

Lately, the use of hierarchies for text classification has been widely investigated and applied by many researchers. For example, Pulijala and Gauch [5] and Gauch et al. [6] have reported that they classified the documents during indexing which can be retrieved by using a combination of keyword and conceptual match. Ruiz and Srinivasan [7] have proposed a text categorization method based on the Hierarchical Mixture of Expert (HME) model using neural networks. Li et al. [8] have proposed another approach of hierarchical document classification using linear discriminant projection to generate topic hierarchies. In addition, Deschacht and Moens [9] have proposed an automatic hierarchical entity classifier for tagging noun phrases in a text with their WordNet synset using conditional random fields (CRF). Meanwhile, Xue et al. [10] have developed a deep-classification algorithm to classify web documents into categories in large-scale text hierarchy.

Additionally, several statistical classification methods and machine learning techniques have been applied to text and web pages classification including techniques based on decision tree, neural network [7] and support vector machine (SVM) [2], [11], [12], [13]. SVM has been prominently and widely used for different classification task, in particular for document classification. For instance, in [2], Sun and Lim have developed a top-down level-based hierarchical classification method for category tree using SVM classifiers. Afterward, they have evaluated the performance of their hierarchical classification method by calculating the category-similarity measures and distance-based measures. Nenadic et al. [11] have used SVM for classifying the gene names from the molecular biology literature. Meanwhile, Wang and Gong [9] have used SVM to distinguish between any two sub-categories under the same concept or category for web page hierarchical classification. They have used the voting score from all category-to-category classifier for assigning a web document to a sub-category. And they have reported that their method can improve the performance of imbalanced data. Dumais and Chen [13] have reported that they used the hierarchical structures for classifying web content. In their research, they have employed SVM to train second-level category models using different contrast sets. Then, they have classified a web content based on the scores that were combined from the top-level and second-level model. In [14], Liu et al. have analyzed the scalability and the effectiveness of SVM for classifying a very large-scale taxonomy using distributed classifier.

Although many classification methods have been proposed in various domains in recent years such as in [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] and [15], only little attention has been paid to the hierarchical classification of biomedical literature. Towards this effort, in this paper, we propose a hierarchical classification method where we employ the hierarchical ‘concept’ structure for classifying biomedical text abstracts by using ontology alignment algorithm that we have developed. Our proposed method is different compared to the previous works because we construct two types of hierarchies, which are the OHSUMED disease hierarchy (ODH) as our training hierarchy and the Medline abstract disease hierarchy (MADH) as testing hierarchy. Afterward, we enrich our training hierarchy by mapping the relevant features that extracted from selected categories in the OHSUMED dataset to each node or category in the ODH based on

their specific category. Next, we perform ontology alignment by employing our ontology alignment algorithm, namely "Anchor-Flood" algorithm (AFA) [16]. During ontology alignment phase, AFA matches the concepts and relations between the "enriched" OHSUMED disease hierarchy (EODH) and the MADH for exploring and searching the aligned pairs. In our research, we consider the aligned pairs as a set of probable relevant categories for classification purpose. Then, we evaluate the more specific concepts by calculating the cosine similarity score between the new Medline abstract and each probable category. Finally, we classify the new Medline abstract based on the highest cosine similarity score. In order to evaluate our approach, we conducted the experiments of the multi-class flat classification and the hierarchical classification (using the ODH and EODH). We use the results of hierarchical classification using the ODH as our baseline. The experimental evaluation indicates that our proposed approach performs slightly better than the performance of the baseline.

We organize the paper as follows: In Section 2, we describe our proposed approach to hierarchical classification. Afterward, the text preprocessing process is explained in Section 3. In Section 4 and 5, we discuss on how to construct the "OHSUMED disease hierarchy and the "enriched" OHSUMED disease hierarchy, respectively. Section 6 describes the Medline abstract disease hierarchy. The ontology alignment process is explained in Section 7. In Section 8, we discuss the ontology based hierarchical classification. Section 9 contains the experiments and in Section 10, the discussions of the classification results are stated. Finally, we conclude this paper with a summary and suggestions for future work in section 11.

2. PROPOSED APPROACH TO HIERARCHICAL CLASSIFICATION

The large number of biomedical literature that published on web makes the process of classification become challenging and arduous. The reason is that there are many categories of biomedical literature available in the web such as gene, protein, human disease, etc. and each category have many different classes. For instance, human disease category contains many different classes including heart disease, cancer, diabetes and hepatitis. Moreover, each disease class consists of many subclasses. In heart disease category contains arrhythmia, heart block, myocardial diseases, etc. Due to this situation, various classification methods were proposed by many researchers for classifying biomedical literature. For example, Nenadic et al. in [8] reported that they had employed SVM in order to classify the gene names that extracted in the molecular biology literature.

However, our approach is different from the previous works, where we explore the use of hierarchical 'concept' structure with the help of ontology alignment algorithm for searching and identifying the probable categories in order to classify biomedical text abstracts. To realize our propose method, we use the features that extracted from the datasets to index the biomedical text abstracts. These features will be used to represent our documents in order to improve the accuracy of classification performance and also the result of searching relevant documents. Due to this reason, we construct two types of hierarchies, which are the ODH as our training hierarchy and the MADH as testing hierarchy. For this purpose, initially, we construct the ODH by referring to the OHSUMED disease directory. Subsequently, we perform text preprocessing (including part-of-speech tagging, phrase chunking, etc.) for extracting and selecting the relevant features from the OHSUMED dataset and the Medline abstracts respectively. Then, we enrich the ODH by assigning the relevant features that extracted from OHSUMED dataset to each node of the hierarchy. While, the MADH were constructed using a collection of biomedical text abstracts that downloaded from the Medline database.

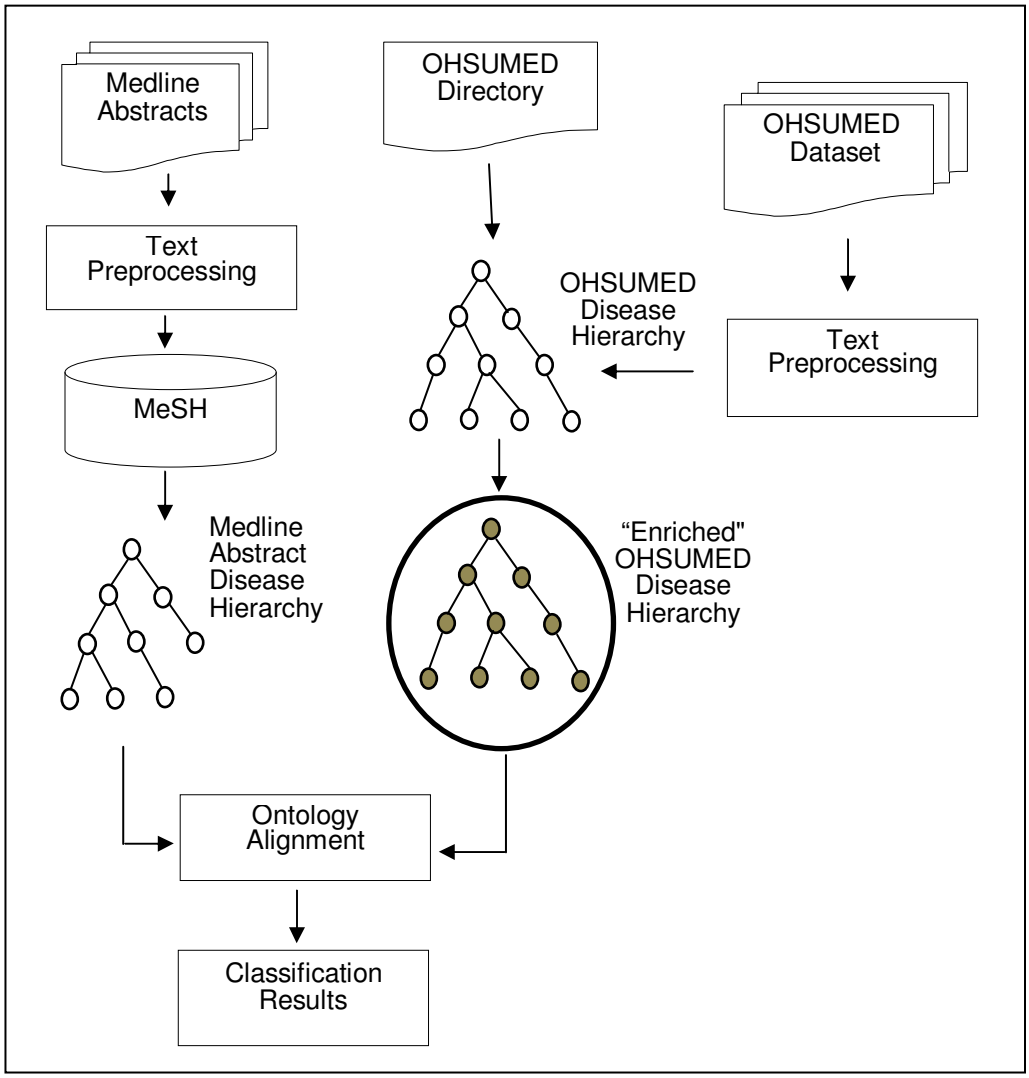

FIGURE 1: A Method for Hierarchical Classification of Biomedical Text Abstracts

Next, we perform the ontology alignment process for matching and aligning the EODH and the MADH using the "Anchor-Flood" algorithm (AFA). During ontology alignment process, AFA would match and search the concepts in both hierarchies in order to compute the similarity among the concepts and relations in the EODH and the MADH for identifying and producing the aligned pairs. We consider the aligned pairs as a set of probable categories for classifying biomedical text abstracts. Afterward, we evaluate the more specific concepts based on the cosine similarity score between the vectors of unknown new abstract in each MADH and the vectors of each probable category in the EODH for predicting more specific category. Eventually, we classify the new Medline abstracts into the first rank of cosine similarity score. Figure 1 illustrates our proposed hierarchical classification method that is implemented in our research.

3. TEXT PREPROCESSING

In our experiments, we use two different datasets, which are a subset of the OHSUMED dataset as training documents and the Medline abstracts as test documents. The OHSUMED dataset [12] is a subset of clinical paper abstracts from the Medline database, from year 1987 to 1991. This dataset contains more than 350,000 documents. However, we select 400 documents from 43 categories of the subset of OHSUMED dataset for enriching the ODH. For each category, we

retrieve on average 10 documents. For classification purpose, we have selected randomly a subset of the Medline abstracts from the Medline database. Using a PubMed website, we retrieve this dataset with the query terms, such as “arrhythmia”, "heart block", etc. to retrieve the related Medline abstracts which containing the query terms. Then, we index this dataset belonging to 15 categories of disease as shown in Table 1. A total number of Medline abstracts are 150.

In our research, each document must be represented by a set of feature vectors. Accordingly, we use noun phrases as our features. For this purpose, we perform text preprocessing to extract and select a list of unique and relevant features from our training and testing datasets. Our text preprocessing process consists of feature extraction and feature selection phase.

TABLE 1: The Number and Categories of Medline Abstracts

Category No.	Category Name	No. of Documents
1	Arrhythmia	10
2	Heart Block	10
3	Coronary Disease	10
4	Angina Pectoris	10
5	Heart Neoplasms	10
6	Heart Valve Diseases	10
7	Aortic Valve Stenosis	10
8	Myocardial Diseases	10
9	Myocarditis	10
10	Pericarditis	10
11	Tachycardia	10
12	Endocarditis	10
13	Mitral Valve Stenosis	10
14	Pulmonary Heart Disease	10
15	Rheumatic Heart Disease	10
	Total	**150**

3.1 Feature Extraction

In text preprocessing process, initially we extract the noun phrases as our features from the OHSUMED dataset and the Medline abstracts respectively. The purpose of feature extraction is to generate a list of unique features from the datasets. In our research, this process is done by performing part-of-speech (POS) tagging and phrase chunking. POS tagging can be defined as a task of assigning POS categories to terms from a predefined set of categories. Meanwhile, phrase chunking is the process of identifying and recognizing the noun phrases constructed by the POS tags. In POS tagging phase, we have automatically assigned POS tags to each term using the rule based POS tagger. Then, we extract features from the tagged text based on the chunks that consists of noun phrases. Finally, we create a list of unique features that extracted from the training and testing datasets, respectively.

3.2 Feature Selection

Feature selection phase is one of the most important tasks in text preprocessing. This is due to the fact that some features are uninformative and they do not influence the classification performance. Furthermore, as the number of unique features which extracted from our training and testing dataset is big, feature selection phase can help us to reduce the original features to a small number of features by removing the rare and irrelevant features. During feature selection phase, we attempt to find relevant features for constructing the MADH and also for enriching the ODH. Therefore, we employ the document frequency and the chi-square (χ^2) techniques to distinguish between relevant and irrelevant features, before we eliminate some percentage of the extracted features according to their document frequency and dependency between categories and features.

In order to select the relevant features for representing our datasets, we use the document frequency as a feature reduction technique for eliminating rare features. Therefore, we compute the document frequency for each unique feature in both datasets. Then, we eliminate the features with the highest and lowest frequencies. By performing this process, we could reduce the feature space into a small number of important features. Subsequently, a χ^2 test is used to measure the independence between feature (*t)* and category (*c*) in order to distinguish between relevant and irrelevant features. We attempt to identify and search the most discriminating features for each category. Then, we select the relevant features by assigning features to specific categories. We measure the relationship between features (*t)* and categories (*c*) using the following equation.

$$\chi^2 = \sum_{i,j} \frac{(o_{i,j} - e_{i,j})^2}{e_{i,j}}; \quad (1)$$

where $o_{i,j}$ is the observed frequency for each cell, while $e_{i,j}$ is the expected frequency for each cell.

In our χ^2 test, we use the 2 x 2 contingency table to compare the χ^2 distribution with one degree of freedom. Then, we rank the features according to the χ^2 score. In our experiments, we select features set by choosing features with the χ^2 score greater than 3.841 (our threshold) as the relevant features. Thereafter, we use all of selected features (as concepts) for constructing the MADH and enriching the ODH. Table 2 shows the statistic of the features used in our experiments.

TABLE 2: The Statistic of Selected Features

Experiments	Features	Number of Features
Flat Classification	Features (Before employing feature selection)	3,145
	Feature (After employing feature selection)	1,081
Hierarchical Classification	Features (Before enriching ODH)	43
	Features (After enriching ODH & before employing feature selection)	3,145
	Feature (After enriching ODH & employing feature selection)	1,081

For classification purpose, each document in our datasets is represented by a set of relevant feature vectors, whereby each feature in a vector of a document representing either a single word or multi-words in the document. Accordingly, we assign the term frequency as the feature weighting scheme to each feature for representing our documents. Term frequency $tf_{i,f}$ is the frequency of term *i* occurs in document *j* and *f* = 1, …, *m*. After text preprocessing, we assume that the document *d* is represented as follow;

$d_1 = \{ tf_{11} , tf_{12} , \dots , tf_{1m} \}$

.

.

$d_n = \{ tf_{n1} , tf_{n2} , \dots , tf_{nm} \}$

4. THE OHSUMED DISEASE HIERARCHY (ODH)

In this research, we construct the ODH by referring to the OHSUMED directory. This directory could be accessed in the OHSUMED dataset [17]. There are 101 categories in the OHSUMED directory which are divided into four levels. Level 1 contains 23 categories and level 2 consists of

56 categories. In level 3, there are 16 categories and finally, level 4 contains only 6 categories. We then construct the ODH using Protégé. Figure 2 shows the part of the OHSUMED directory, while Figure 3 illustrates the part of the ODH.

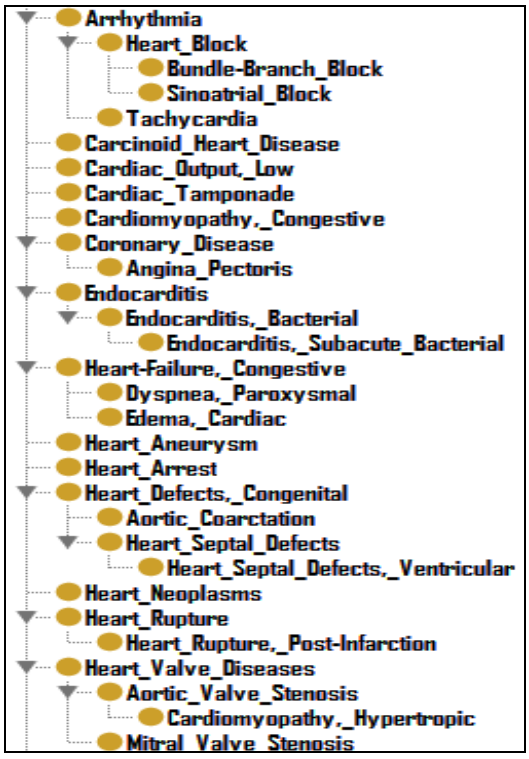

FIGURE 2: The Part of OHSUMED Directory

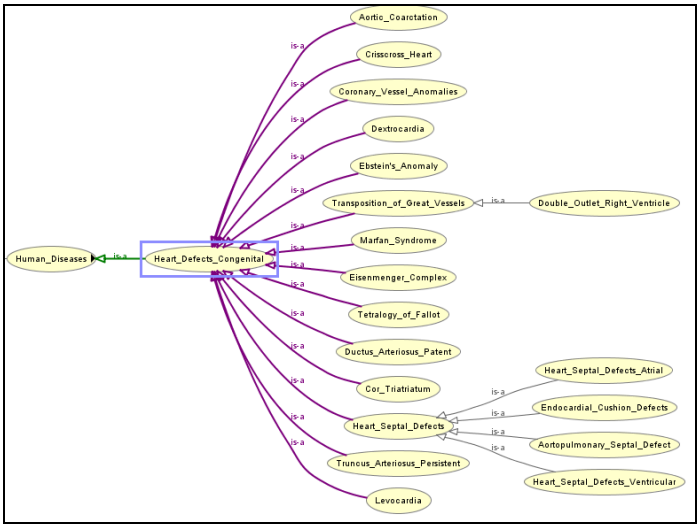

FIGURE 3: The Part of OHSUMED Disease Hierarchy

5. THE "ENRICHED" OHSUMED DISEASE HIERARCHY (EODH)

The important task in enriching the ODH is to select meaningful and relevant features from the OHSUMED dataset. In order to enrich the ODH, we select 43 categories from the OHSUMED

dataset and for each category, we retrieve about 10 documents. Afterward, we perform text preprocessing. The description of text preprocessing has been explained in Section 3.

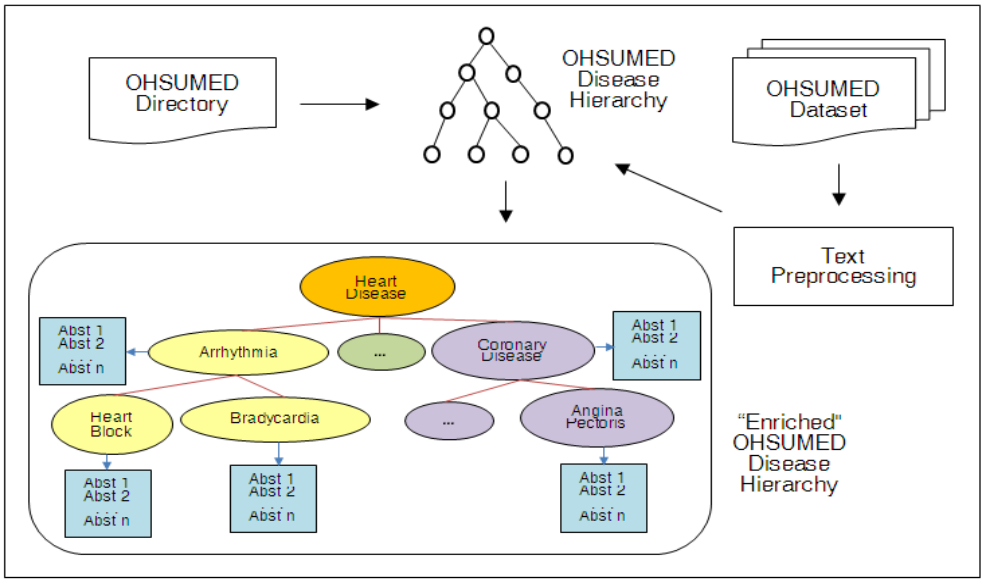

FIGURE 4: An Approach for Constructing and Enriching the ODH

During text preprocessing process, we attempt to identify and select a set of relevant features for each node. Next, we use the relevant features that extracted from 43 selected categories in the OHSUMED dataset as feature vectors. For enriching the ODH, these feature vectors are mapped to each node or concept of the ODH according to their specific category as shown in Figure 4. Meanwhile, Figure 5 describes the example of the EODH.

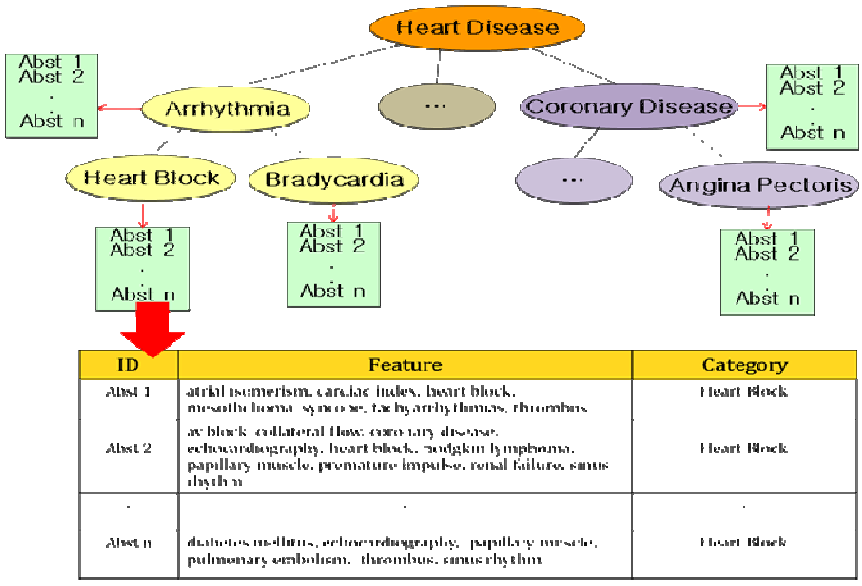

FIGURE 5: An Example of the "Enriched" OHSUMED Disease Hierarchy

6. THE MEDLINE ABSTRACT DISEASE HIERARCHY (MADH)

We construct the Medline abstract disease hierarchy using the selected features that are extracted from a collection of biomedical text abstracts that downloaded from the Medline database. Section 3 contains the description of text preprocessing process.

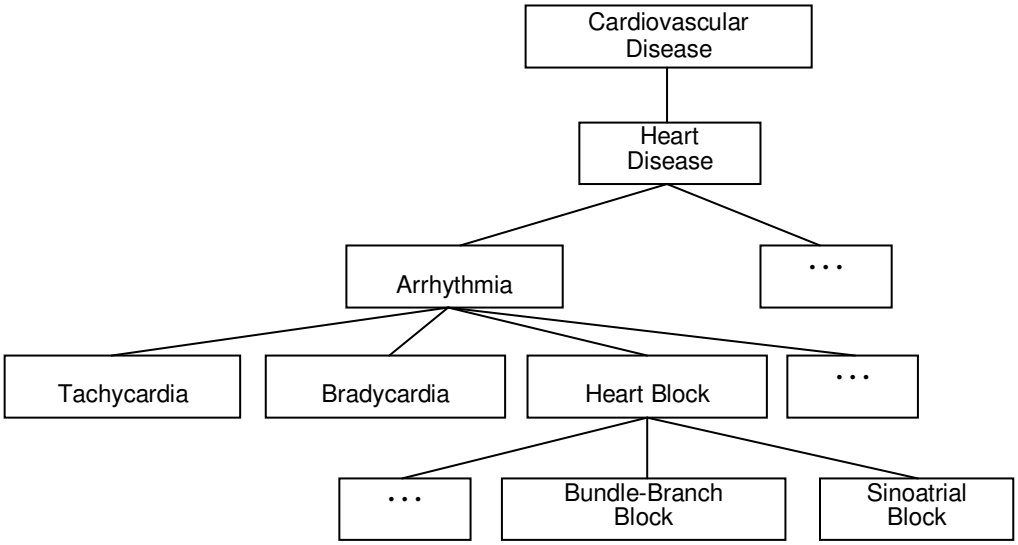

FIGURE 6: An Example of a Part of the MADH

In our research, we use Protégé for constructing the MADH. For this purpose, we have to assign the best-matching concept identifier to each selected feature. Therefore, we refer to the Medical Subject Headings (MeSH) for indexing and assigning the relevant feature in the Medline abstracts into a hierarchical structure. The reason is that the MeSH terms are arranged hierarchically and by referring to the concepts in the MeSH tree structure [18], we could identify heading and subheading of hierarchical grouping before indexing these selected features for representing our testing dataset. Finally, we construct the MADH by using the heading and subheading of hierarchical grouping that are suggested by the MeSH tree structure. Figure 6 depicts a part of the MADH.

7. ONTOLOGY ALIGNMENT

The purpose of ontology alignment in our research is to match the concepts and relations between the EODH and the MADH. Therefore, we perform ontology alignment using the "Anchor-Flood" algorithm (AFA). During ontology alignment process, AFA would explore and search for the similarity among the neighboring concepts in both hierarchies based on terminological alignment and structural alignment. Then, AFA would narrow down the EODH and the MADH for producing the aligned pairs. These aligned pairs are obtained by measuring similarity values, which consider textual contents, structure and semantics (available in the hierarchies) between pairs of entities. We consider all of the aligned pairs as a set of probable categories for classification purpose.

8. ONTOLOGY BASED HIERARCHICAL CLASSIFICATION

In our proposed approach, we construct two types of hierarchies, which are the ODH, as our training hierarchy and the MADH as testing hierarchy. Then, we perform ontology alignment in order to match both hierarchies for producing the aligned pairs, which we consider as a set of probable categories for predicting and classifying a new Medline abstract.

Afterward, we evaluate the more specific concepts based on the similarity between the new Medline abstract and each probable relevant category. Consequently, we compute the cosine

similarity score between the vector of unknown new abstract in each MADH and the vector of each probable category in the EODH for identifying and predicting more specific category. The cosine similarity score is calculated using the following equation.

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}} \quad (2)$$

where the vector of $d_j = (w_{11}, w_{12}, \ldots, w_{1n})$ and the vector of $d_k = (w_{21}, w_{22}, \ldots, w_{2n})$.

Then, we sort all the probable categories according to the assigned cosine similarity score. Eventually, we classify the new Medline abstracts based on the highest cosine similarity score.

9. EXPERIMENT

For our experiments, we have used 400 records (documents) from 43 different categories of the subset of the OHSUMED dataset for enriching the ODH. On the other hand, for classification purpose, we have randomly downloaded 150 biomedical text abstracts that related to human diseases, such as “arrhythmia”, "coronary disease”, “angina pectoris”, etc. from Medline database. The description of text preprocessing has been discussed in Section 3.

In order to evaluate the performance of our proposed method, we conduct two different experiments, which are multi-class flat classification and hierarchical classification. In both flat and hierarchical classification, we performed a few experiments using the features (for enriching our initial training hierarchy or ODH) that produced before and after feature selection process. Moreover, we also conducted the experiments using very few features, whereby only consists of keyword for each node in our training hierarchy (ODH) as a baseline for hierarchical classification. Then, we compare the performance of our method for classifying biomedical text abstracts with the performance of the multi-class flat classification using LIBSVM [19].

8.1 Flat Classification

In the flat classification, we have conducted the multi-class classification experiments using LIBSVM. In these experiments, we ignore hierarchical structure and treat each category separately. Then, we compare the results that produced from these experiments with the performance of our proposed method for hierarchical classification of biomedical text abstracts. We use the results of flat classification (without feature selection process) as our baseline for evaluating the performance of our proposed approach for hierarchical classification.

8.2 Hierarchical Classification

For the hierarchical classification experiments, we attempt to assign a category for a given new Medline abstract. To achieve our goal, we consider the aligned pairs were produced during ontology alignment process as a set of our probable categories. Then, we evaluate the more specific category based on the similarity between the new Medline abstract and each probable category. For this purpose, we compute the cosine similarity score between the vector of unknown new abstract in each MADH and the vector of each probable category in the EODH. After that, we sort all the probable categories according to the assigned cosine similarity score. In our research, we consider the highest cosine similarity score as the relevant category. Finally, we classify the new Medline abstracts into a category that has the highest cosine similarity score.

In order to evaluate our approach, we conduct three types of the experiments for hierarchy classification of biomedical text abstracts. Firstly, we perform the hierarchical classification using the initial training hierarchy or ODH (without enriching ODH). Then, we repeat the experiments of hierarchical classification using the “Enriched” ODH (without feature selection process). Furthermore, we also conduct the experiments of hierarchical classification using the “Enriched” ODH (with feature selection process).

10. DISCUSSION

The results of the flat and hierarchical classification are shown in Table 3, Table 4 and Figure 7. From the experiments, the results show the different performance for each category in the flat and hierarchical classification. Generally, the experimental results indicate that our proposed approach performs slightly better than the baseline. We observe that our proposed approach to hierarchical classification achieve the average accuracies of 14% (for hierarchical classification using the features in the ODH only), 30.67% (for hierarchical classification using the features in the EODH and without feature selection process) and 32.67% (for hierarchical classification using the features in the EODH and with feature selection process), respectively. Nevertheless, the accuracies of the flat classification are on the average 6.67% (for flat classification without feature selection process) and 18% (for flat classification with feature selection process), respectively.

TABLE 3: The Results of the Flat Classification

Category No.	Category Name	Flat Classification (% Accuracy)	Flat Classification + Feature Selection (% Accuracy)
1	Arrhythmia	0	20
2	Heart Block	0	0
3	Coronary Disease	0	0
4	Angina Pectoris	0	20
5	Heart Neoplasms	20	50
6	Heart Valve Diseases	10	10
7	Aortic Valve Stenosis	0	0
8	Myocardial Diseases	0	20
9	Myocarditis	10	20
10	Pericarditis	20	0
11	Tachycardia	0	30
12	Endocarditis	0	20
13	Mitral Valve Stenosis	10	30
14	Pulmonary Heart Disease	30	30
15	Rheumatic Heart Disease	0	20
Average		**6.67**	**18**

Table 3 shows the performance of the flat classification for each category of biomedical text abstracts. In general, the classification performances for category 1, 4, 5, 8, 9, 11, 12, 13 and 15 using the flat classification approach (with feature selection process) are better than the flat classification approach (without feature selection process). For instance, the performance of flat classification approach (with feature selection process) reach the highest accuracy (50%) in category 5, while the classification accuracy of the flat classification approach (without feature selection process) only achieve 20% in the same category, as shown in Table 3. These results might demonstrate that if the relevant features are selected carefully for representing a document, the accuracy of biomedical text abstracts classification would be increased.

In addition, we have compared the performances of hierarchical classification using different approaches. Table 4 illustrates the effect of enriching the ODH and employing the feature selection process for classifying 15 categories of biomedical text abstracts. For the hierarchical classification approach (using EODH and with feature selection process), the results of the category 1, 5, 7, 8, 10, 12, 13, 14 and 15 show the best performance compared to the hierarchical classification approach (using EODH and without feature selection process) and the hierarchical classification approach (using ODH only) as shown in Table 4.

TABLE 4: The Results of the Hierarchical Classification

Category No.	Category Name	Hierarchical Classification + ODH (% Accuracy)	Hierarchical Classification + "Enriched" ODH (% Accuracy)	Hierarchical Classification + "Enriched" ODH + Feature Selection (% Accuracy)
1	Arrhythmia	0	50	50
2	Heart Block	0	0	0
3	Coronary Disease	0	0	0
4	Angina Pectoris	0	0	0
5	Heart Neoplasms	0	30	30
6	Heart Valve Diseases	0	0	0
7	Aortic Valve Stenosis	10	20	20
8	Myocardial Diseases	0	40	50
9	Myocarditis	0	0	0
10	Pericarditis	40	60	70
11	Tachycardia	0	0	0
12	Endocarditis	40	70	70
13	Mitral Valve Stenosis	10	60	70
14	Pulmonary Heart Disease	20	40	40
15	Rheumatic Heart Disease	90	90	90
Average		**14**	**30.67**	**32.67**

Furthermore, the classification accuracies for the hierarchical classification approach (using EODH and without feature selection process) produce quite similar results to the performance of hierarchical approach (using EODH and with feature selection process) for all categories except for the category 8 (40%), category 10 (60%) and category 13 (60%). Nonetheless, for the hierarchical approach (using ODH), the classification accuracies of the some categories such as in category 10 and 15 achieve quite good results. Overall, the results for the category 15 using the hierarchical classification approaches are better compared to other categories, which achieve 90% of accuracy. These results indicate that the hierarchical classification approaches has been able to classify the biomedical text abstracts correctly although the number of training documents representing each category is small.

By comparing Table 3 and Table 4, we observe that overall average accuracies of the hierarchical classification show better performance than the average accuracies of the flat classification. For example, the classification accuracies for the category 12 and 15 show better results when employing the hierarchical classification approaches compared with the flat classification approaches as shown in Table 3 and Table 4. According to the results that are obtained from the experiments, we can say that the hierarchical classification approaches can increase the classification accuracy because our proposed approach can identify and propose more relevant category for classifying the biomedical text abstracts by using our ontology alignment algorithm.

Additionally, we also noticed that the feature selection process has little influence on the classification performance in both flat and hierarchical classification. The results of the flat classification approach (with feature selection process) show quite good performance than the flat classification approach (without feature selection process), especially for category 5 and 11, as shown in Table 3. On the other hand, the classification performance of hierarchical classification approach (with feature selection process) is slightly better than the classification performance of hierarchical classification approach (without feature selection process) as shown in the Table 4 and Figure 7. The average accuracies of the flat and hierarchical classification approaches (with feature selection process) are 18% and 32.67%, respectively. These results indicate that the flat and hierarchical classification approaches (with feature selection process) has been able to classify a new unknown biomedical text abstract correctly even though we use a small number of relevant features for representing each category.

In addition, the classification accuracies of a few categories such as in category 12 and 13 produce better results in both flat and hierarchical classification (with feature selection process) experiments compared to other diseases categories. This might be caused by the number of content-bearing keywords that are extracted and selected in these categories are higher than other categories. Moreover, the results of the flat and hierarchical classification for some categories such as category 2 and 3 show 0% accuracies. The main reason of the performances of flat and hierarchical classification for these categories being poor is that the selected features for representing each document are sparse or common features.

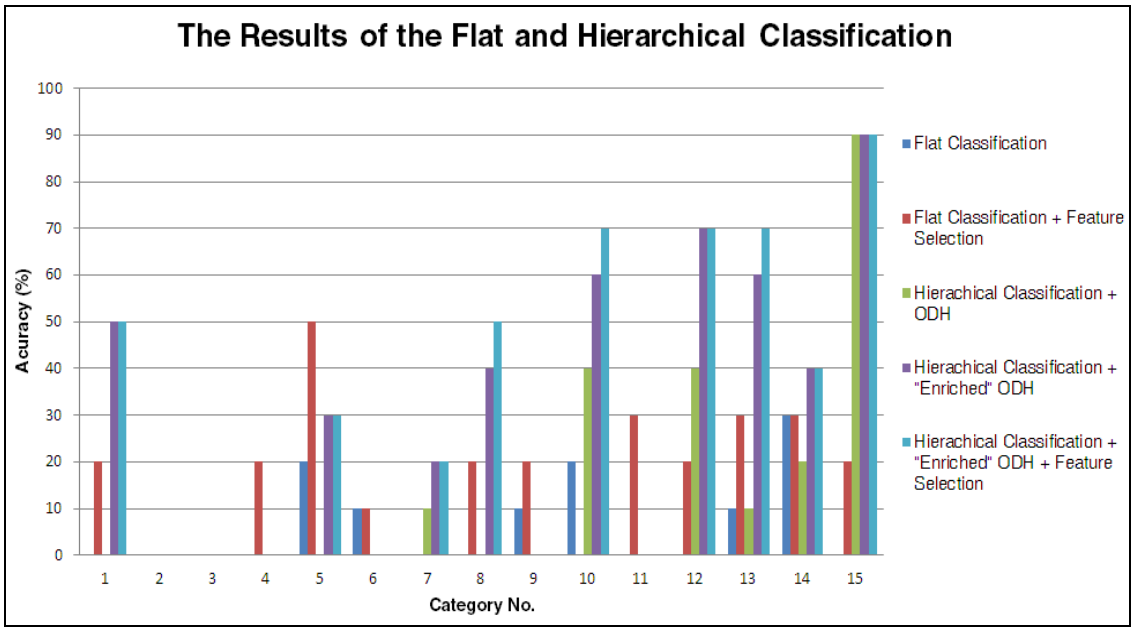

FIGURE 7: The Performance of Classification Accuracies

Although the performance of hierarchical classification experiments produced better results than the flat classification, our proposed approach is still naïve in achieving the good classification accuracies. The low performance in the hierarchical classification might be caused by the shortness of the Medline abstracts or extraction of very few relevant features. Consequently, we construct a small hierarchy for ontology alignment purpose, which may produce a small number of aligned pairs (as our probable category). Furthermore, the small number of documents that are represented in a particular category in the dataset may also affect the decrease of the classification accuracy. Even though the performance of our proposed approach for hierarchical classification is still below than our target, we believe that we can improve the classification accuracies. We are confident that, by enriching the ODH and also the MADH with the relevant features, we can identify more probable categories for classifying biomedical text abstracts with the help of our ontology alignment algorithm.

11. CONCLUSION

The main purpose of our research is to improve the performance of hierarchical classification by increasing the accuracies of classes in the datasets that are represented with a small number of biomedical text abstracts. Therefore, in this paper, we propose the hierarchical classification approach that utilizes the ontology alignment algorithm for classification purpose. Our approach is different from the previous works, where we explore the use of hierarchical 'concept' structure with the help of our ontology alignment algorithm, namely 'Anchor-Flood' algorithm (AFA) for searching and identifying the probable categories in order to classify the given biomedical text abstracts.

Then, we evaluate the performance of our approach by conducting the hierarchical classification experiments using the features that are extracted from the OHSUMED dataset and Medline abstracts. We also conduct the multi-class flat classification experiments using LIBSVM. Moreover, we perform feature selection by employing the document frequency and chi-square techniques for selecting relevant features. Then, we perform a few experiments for the flat and hierarchical classification using the features that are selected from feature selection process. Generally, the experimental results indicate that our propose approach of hierarchical 'concept' structure using ontology alignment algorithm can improve the classification performance. Although our proposed approach is still naïve in achieving the good classification accuracies, we believe that we could modify our proposed approach to produce more relevant probable categories and predict more specific category for classifying biomedical text abstracts.

Our future target is to seek and propose more accurate approaches for selecting relevant and meaningful features in order to enrich or expand the ODH and MADH. These features would be used to index the biomedical text abstracts for increasing the accuracy of classification performance and also the result of searching relevant documents. Furthermore, the performance of our proposed approach for hierarchical text classification also can be improved by increasing the total number of documents that are represented in each category in the dataset.

ACKNOLEDGEMENT

This work was supported in part by Global COE Program "Frontiers of Intelligent Sensing" from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

12. REFERENCES

1. A. M. Cohen. *"An effective general purpose approach for automated biomedical document classification"*. AMIA Annual Symposium Proceeding, 2006:161-165, 2006

2. A. Sun and E. Lim. *"Hierarchical text classification and evaluation"*. In Proceeding of the IEEE International Conference on Data Mining. Washington DC, USA, 2001

3. F. M. Couto, B. Martins and M. J. Silva. *"Classifying biological articles using web sources"*. In Proceedings of the ACM Symposium on Applied Computing. Nicosia, Cyprus, 2004

4. A. Singh and K. Nakata. *"Hierarchical classification of web search results using personalized ontologies"*. In Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction. Las Vegas, NV, 2005

5. A. Pulijala and S. Gauch. *"Hierarchical text classification"*. In Proceedings of the International Conference on Cybernetics and Information Technologies (CITSA). Orlando, FL, 2004

6. S. Gauch, A. Chandramouli and S. Ranganathan. *"Training a hierarchical classifier using inter-document relationships"*. Technical Report, ITTC-FY2007-TR-31020-01, August 2006

7. M. E. Ruiz and P. Srinivasan. *"Hierarchical text categorization using neural networks"*. Information Retrieval, 5(1):87-118, 2002

8. T. Li, S. Zhu and M. Ogihara. *"Hierarchical document classification using automatically generated hierarchy"*. Journal of Intelligent Information Systems, 29(2):211-230, 2007

9. K. Deschacht and M. F. Moens. *"Efficient hierarchical entity classifier using conditional random fields"*. In Proceedings of the 2nd Workshop on Ontology Learning and Population. Sydney, Australia, 2006

10. G. R. Xue, D. Xing, Q. Yang and Y. Yu. *"Deep classification in large-scale text hierarchies"*. In Proceeding of the 31st Annual International ACM SIGIR Conference. Singapore, 2008

11. G. Nenadic, S. Rice, I. Spasic, S. Ananiadou and B. Stapley. *"Selecting text features for gene name classification: from documents to terms"*. In Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine, PA, USA, 2003

12. Y. Wang and Z. Gong. *"Hierarchical classification of web pages using support vector machine"*. Lecture Notes in Computer Science, Springer, 5362/2008:12-21, 2008

13. S. Dumais and H. Chen. *"Hierarchical classification of web content"*. In Proceedings of 23rd ACM International Conference on Research and Development in Information Retrieval. Athens, Greece, 2000

14. T. Y. Liu, Y. Yang, H. Wan, H. J. Zeng, Z. Chen and W. Y. Ma. *"Support vector machines classification with a very large-scale taxonomy"*. ACM SIGKDD Explorations Newsletter – Natural language processing and text mining, 7(1):36-43, 2005

15. G. Nenadic and S. Ananiadou. *"Mining semantically related terms from biomedical literature"*. Journal of ACM Transactions on Asian Language Information Processing, 5(1):22-43, 2006

16. M.H. Seddiqui and M. Aono. *"An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size"*. Web Semantics: Science, Services and Agents on the World Wide Web, 7:344-356, 2009

17. OHSUMED dataset. Dataset available at http://davis.wpi.edu/xmdv/datasets/ohsumed.html, 2005

18. Medical Subject Heading (MeSH) tree structures. Available at http://www.nlm.nih.gov/mesh/trees.html, 2010

19. C.-C. Chang and C.-J. Lin. *"LIBSVM: a library for support vector machines"*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2007

On Tracking Behavior of Streaming Data: An Unsupervised Approach

Niloofar Mozafari mozafari@cse.shirazu.ac.ir
Department of Computer Science and Engineering
and Information Technology
Shiraz University
Shiraz, Iran

Sattar Hashemi s_hashemi@shirazu.ac.ir
Department of Computer Science and Engineering
and Information Technology
Shiraz University
Shiraz, Iran

Ali Hamzeh ali@cse.shirazu.ac.ir
Department of Computer Science and Engineering
and Information Technology
Shiraz University
Shiraz, Iran

Abstract

In the recent years, data streams have been in the gravity of focus of quite a lot number of researchers in different domains. All these researchers share the same difficulty when discovering unknown pattern within data streams that is concept change. The notion of concept change refers to the places where underlying distribution of data changes from time to time. There have been proposed different methods to detect changes in the data stream but most of them are based on an unrealistic assumption of having data labels available to the learning algorithms. Nonetheless, in the real world problems labels of streaming data are rarely available. This is the main reason why data stream communities have recently focused on unsupervised domain. This study is based on the observation that unsupervised approaches for learning data stream are not yet matured; namely, they merely provide mediocre performance specially when applied on multi-dimensional data streams.

In this paper, we propose a method for **Track**ing **Ch**anges in the behavior of instances using **C**umulative **D**ensity **F**unction; abbreviated as *TrackChCDF*. Our method is able to detect change points along unlabeled data stream accurately and also is able to determine the trend of data called *closing* or opening. The advantages of our approach are three folds. First, it is able to detect change points accurately. Second, it works well in multi-dimensional data stream, and the last but not the least, it can determine the type of change, namely *closing* or *opening* of instances over the time which has vast applications in different fields such as economy, stock market, and medical diagnosis. We compare our algorithm to the state-of-the-art method for concept change detection in data streams and the obtained results are very promising.

Keywords: Data Stream, Trend, Concept Change, Precision, Recall, F1 Measure, Mean Delay Time.

1. INTRODUCTION

Recently, data streams have been extensively investigated due to the large amount of applications such as sensor networks, web click streams and network flows [1], [2], [3], [4], [5]. Data stream is an ordered sequence of data with huge volumes arriving at a high throughput that must be analyzed in a single pass [6], [7]. One of the most important challenges in data streams

and generally in many real world applications is detecting the concept change [6]. In general, the process of transition from one state to another is known as the concept change [8]. In data streams where data is generated form a data generating process, concept change occurs when the distribution of the generated data changes [9], [10].

The problem of concept change detection in time-evolving data has been explored in many previous researches [15], [16], [17], [24]. They are mainly focused on labeled data streams, but nowadays, data streams consist of unlabeled instances and rarely the assumption of having data label is realistic. However, there is some respectable works to detect concept changes in unlabeled data streams [8], [10], [24] and the existing approaches merely offer a mediocre performance on data stream having high dimension. So, in this paper, we are trying to propose a new method for Tracking Changes in the behavior of instances using Cumulative Density Function; abbreviated as *TrackChCDF*. It is able to detect change points in unlabeled data streams accurately and also it can track the behavior of instances over the time. To briefly the advantages of our approach are three folds. First, it is able to detect change points accurately. Second, it works well in multi-dimensional data stream, and the last but not the least, it can determine the type of change, namely *closing* or opening of instances over the time which has vast applications in different fields such as economy, stock market, and medical diagnosis. We compare our algorithm to the state-of-the-art method for concept change detection in data streams and the obtained results are very promising.

The reminder of this paper is organized as follows: we outline the previous works in Section 2. Section 3 presents the proposed algorithm. In Section 4, we report the experimental results and the paper concludes with Section 5.

2. RELATED WORK

In general, change is defined as moving from one state to another state [8]. There are some important works to detect changes where some of them detect changes with statistical hypothesis testing and multiple testing problems [11]. In the statistical literature, there are some works for change point detection [12]. However, most of the statistical tests are parametric and also needs the whole data to run [13], [14]. These methods are not applicable in the data stream area, because they require storing all data in memory to run their employed tests [14].

Popular approaches for the concept change detection uses three techniques including (1) sliding window which is adopted to select data points for building a model [15], [16]. (2) Instance weighting which assumes that recent data points in window are more important than the other [17], [18]. (3) Ensemble learning which is created with multiple models with different window sizes or parameter values or weighting functions. Then, the prediction is based on the majority vote of the different models [19], [20], [21], [22]. Both sliding window and instance weighting families suffer from some issues: First, they are parametric methods; the sliding window techniques require determining window size and instance weighting methods need to determine a proper weighting function. Second, when there is no concept change in the data stream for a long period of time, both of sliding window and instance weighting methods would not work well because they do not take into account or give low weights to the ancient instances [10]. The ensemble methods try to overcome the problems that sliding window and instance weighting are faced with by deciding according to the reaction of multiple models with different window sizes or parameter values or weighting functions. However, these techniques need to determine the number of models in the ensemble technique.

Another family of concept change detection methods is based on density estimation. For example, Aggarwal's method [23] uses velocity density estimation which is based on some heuristics instead of classic statistical changes detectors to find changes. As another major works in this family, we could mention Kifer's [24] and Dasu's works [25] which try to determine the changes based on comparing two probability distributions from two different windows [24], [25]. For example, in [24] the change detection method based on KS test determines whether the two probability density estimations obtained from two consequent different windows are similar or not.

However, this method is impractical for high dimensional data streams and also needs to determine the proper window size. Dasu et al. propose a method for change detection which is related to Kulldorff's test. This method is practical for multi-dimensional data streams [25]. However, this method relies on a discretization of the data space, thus it suffers from the curse of dimensionality.

Another major work is proposed by Ho et al. [8], [10]. In Ho's method, upon arrival of new data point, a hypothesis test takes place to determine whether a concept change has been occurred or not. This hypothesis test is driven by a family of martingales [8] which is based on Doob's Maximal Inequality [8]. Although Ho's method detects changes points accurately, it can only detect some types of changes to be detailed in Section 4. Moreover, it is not able to determine the type of changes.

3. PROPOSED METHOD

The problem of concept change detection in time-evolving data is formulated as follows: we are given a series of unlabeled data points $D = \{z_1, z_2, \ldots, z_n\}$. D can be divided into s segments D_i where $\{i=1, 2, .., s\}$ that follow different distribution. The objective of a detection approach is basically to pinpoint when the distribution of data changes along unlabeled data stream.
In our proposed method, in the first step, we rank instances according to their differences to a mean of instances. It determines how much a data point is different from the others. Namely, the Euclidean distance of each instance with mean of instances is calculated using the Equation 1:

$$D_i(Z, z_n) = \sum_{i=1}^{m} \left| z_i - C(Z \cup \{z_n\}) \right| \quad (1)$$

Where m is the number of dimension and $C(Z \cup \{Z_n\})$ indicates the mean of the union of previous instances and new received instance z_n. The value of D is high when the new instance is farther from the representative of previous ones. Then, we calculate the number of instances that are farther than the last received instance from the mean of instances.

$$G = \frac{\#\{i : D_i > D_n\}}{n} \quad (2)$$

In this formula, D is obtained using Equation 1 and n indicates the number of instances. The changes of G toward higher values can be deemed as data points are running away from their representative. In contrast, having data close to their representative conveys that the sequences of G are approaching smaller values. In other words, the sequences of G approach to the smaller values upon widening of data distribution. Conversely, these sequences approach to the higher value when the data distribution is going to be contracted. To be illustrative, suppose that instances are produced from a distribution function whose standard deviation gets high near 2000. We calculated values of G for these instances. As Figure 1 shows, the sequences of G approach to the smaller values near 2000. We repeat this experiment for the case of *closing* instances. Figure 3 illustrates the values of G when the instances get closed near 2000. According to these figures if the sequences of G get high value in an interval of time, it means a change is occurred and the type of change is *closing* and it vice versa for the case of opening. Thus we need a temporary to store sequence of G to track the behavior of instances over the time. To do that, in the second step, we calculate Cumulative Density Function (CDF) of G. Figures 2 and 4 respectively illustrate CDF of G when the instances either get opened or get closed around instance 2000. As these figures illustrate if the sequences of G approach to the smaller value, the Cumulative Density Function (CDF) of G gets a small value and vice versa. Thus we can use this property to detect change points and also determine the type of change, namely *closing* or opening. In other words, this property can help tracking the changes in the behavior of instances over the time.
In the next step, we calculate CH using Equation 3. In this formula G is obtained using Equation 2 and t_w is the smoothing parameter.

$$CH_n = CH_{n-1} + \left(\sum_{t}^{t_c} G_t + \sum_{t}^{t_c} G_{t-t_w}\right)/t_w \quad (3)$$

If the difference of two consecutive *CH* is greater than a threshold, it means the distribution of data is going to be closed over the time, namely, there is a change in the data stream and the type of change is *closing*. If the difference of two consecutive *CH* is smaller than a threshold, it means the distribution of data would be opened, namely, there is a change in the data stream and the type of change is *opening*.

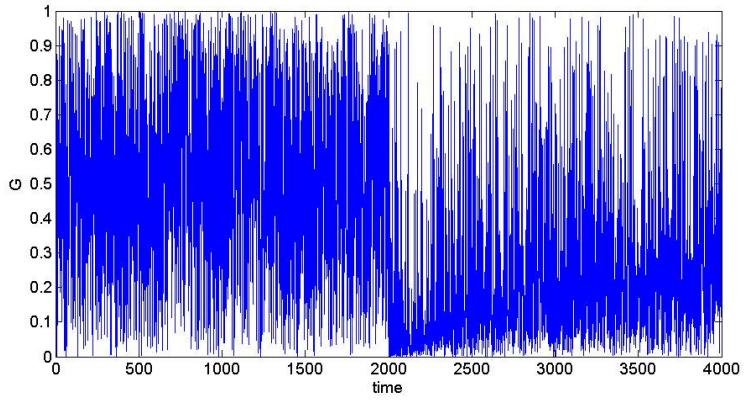

FIGURE 1: the values of *G* when the instances get open near 2000. The horizontal and vertical axes respectively show time and *G* that is calculated by Formula 2. The sequences of *G* approach to the smaller values when the data distribution would be opened.

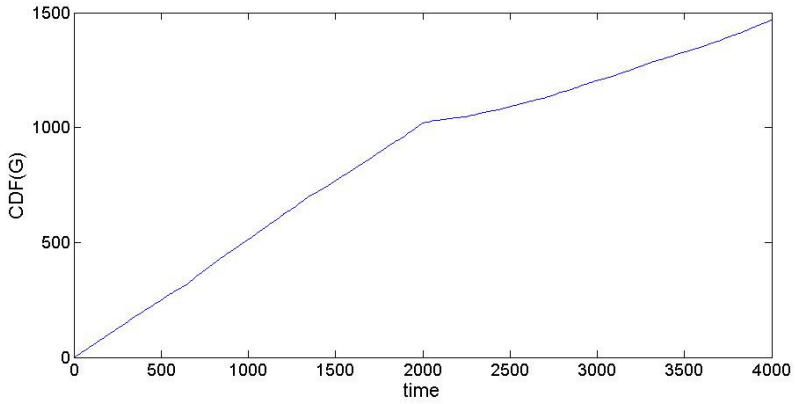

FIGURE 2: the values of CDF of *G* when the instances get open near 2000. The horizontal and vertical axes respectively show time and CDF of *G*. If the two successive slopes of the sequences of CDF are smaller than a threshold, it means the distribution of data would be opened, namely, there is a change in the data stream and the type of change is *opening*.

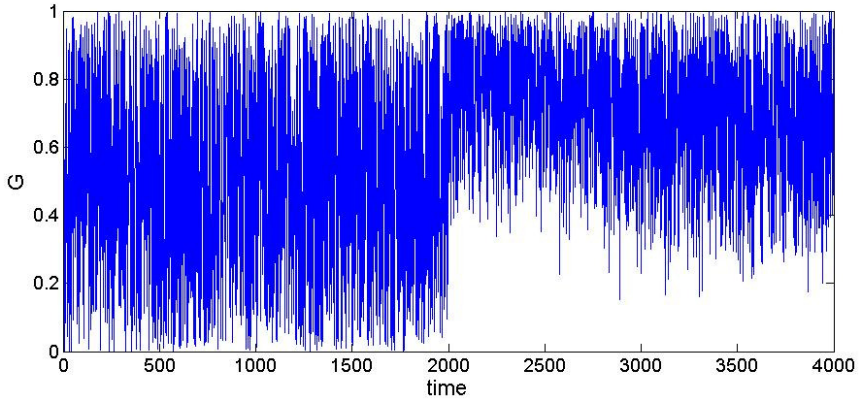

FIGURE 3: the values of *G* when the instances get close near 2000. The horizontal and vertical axes respectively show time and *G* that is calculated by Formula 2. The sequences of *G* approach to the higher values when the data distribution would be closed.

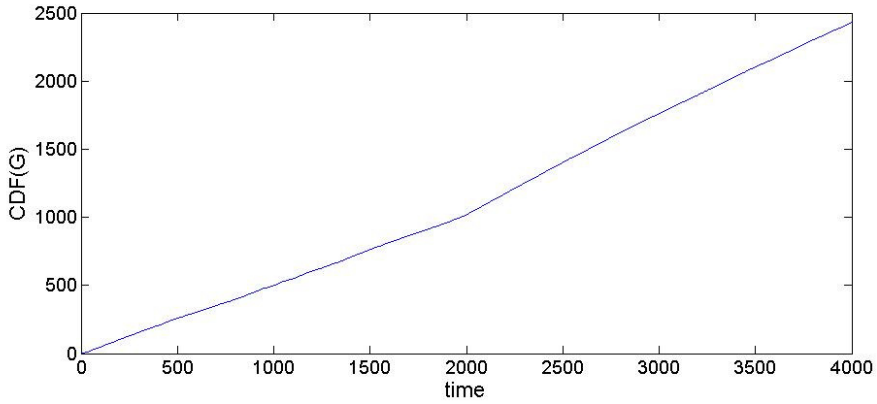

FIGURE 4: the values of CDF of *G* when the instances get close near 2000. The horizontal and vertical axes respectively show time and CDF of *G*. If the two successive slopes of the sequences of CDF are greater than a threshold, it means the distribution of data would be closed, namely, there is a change in the data stream and the type of change is *closing*.

To have a general overview of our algorithm for better understanding, the main steps of our algorithm can be capsulated in the following steps:
The Euclidean distance of each instance to the mean of pervious seen instances is calculated using Equation (1).
G; the number of instances that their distance to the mean is greater than the distance of the last received instance to the mean of instances; is counted using Equation (2).
CH is calculated using Equation (3).
If the difference of two successive *CH* is greater than a threshold, there exists a change and the type of change is *closing*. If the difference of two consecutive *CH* is smaller than a threshold, there exists a change and the type of change is opening.

4. EXPERIMENTAL RESULTS AND DISCUSSION

This section composes of two subsections, precisely covering our observation and analysis. The first subsection presents the experimental setup and description of used evaluation measures. The latter one presents and analyses the obtained results.

4.1 Experimental Setup

This section introduces the examined data set and the used evaluation measures respectively.

Data Set. To explore the advantages of *TrackChCDF*, we conduct our experiments on a data set which was used previously in Ho's work [8], [10]. In this data set, change is defined as the change in the generating model. This change is simulated by varying the parameters of the function generates the data stream. According to this definition, we construct a data set with 20000 instances that changes occur after generating each 2000 instances. Thus, this data set includes ten segments. In each segment, instances are sampled from a Gaussian distribution. We change standard deviation after each 2000 instances. Therefore, this data set has nine change points in instances 2000, 4000, 6000, 8000, 10000, 120000, 140000, 160000, and 180000. Figure 5 illustrates the behavior of data streams over the time. As this figure shows the type of changes are {*opening, opening, opening, closing, closing, opening, closing, closing, closing*}.

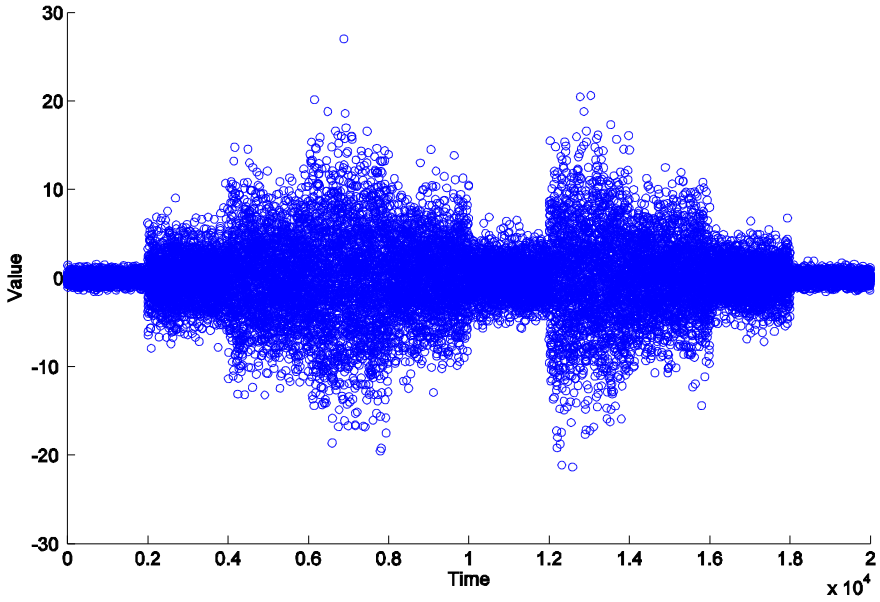

FIGURE 5: the behaviour of instances in the stream data. The X axis shows time and the Y axis indicates the value of instance in each time.

Evaluation Measures. We assess our method with three measurements criterion which is well known in the context [8], [10]: 1) the precision, that is the number of corrected detections divided by the number of all detections. 2) Recall that is the number of corrected detections divided by the number of true changes. 3) F1, that represents a harmonic mean between recall and precision. Following is the definitions of these measurements.

$$\text{Precision} = \frac{\text{Number of Corrected Detections}}{\text{Number of Detections}} \quad (3)$$

$$\text{Recall} = \frac{\text{Number of Corrected Detections}}{\text{Number of True Changes}} \quad (4)$$

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (5)$$

As the precision and recall measures are related by F1, if F1 measure gets higher value, we can ensure that precision and recall are reasonably high.

4.2 Results and Discussion

In this section, the results of applying our method on the studied data set are analyzed. As mentioned previously, this data set is created using ten overlapping Gaussian distributions. To apply concept change in this data set, we change the standard deviation of data after generating each 2000 instances. We compare our method with Ho's method [8], [10]. Table I shows the result of applying Ho's method and our method on this data set. To be more accurate, we ran this experiment 50 times and evaluated our method with three measurements; precision, recall, and F1.

TABLE I: comparison between Ho's method and the proposed method on num-ds data set.

Ho's Approach			**TrackChCDF**		
Precision	Recall	F1-measure	Precision	Recall	F1-measure
0.9850	0.3357	0.5006	0.9728	0.8556	0.9104

The Precision measure of *TrackChCDF* is slightly less than the precision of Ho's method and the Recall measure is significantly higher than Ho's method because our method detects all the change points whereas Ho's method detects smaller number of them. Also, as F1 is the balance between Recall and Precision; we can ensure that Precision and Recall are reasonably high if F1 gets high value. As *TrackChCDF* has the higher F1 in comparison to Ho's method, we can conclude that the proposed method certainly detects the true change points in addition to a few number of false change points. But, according to the value of precision, these false change points are not extortionary. Ho's method only detects those change points where data get away from the mean of data distribution. In Ho's method, changes can be detected when p_values [10] get small. The p_values will be small when the number of strangeness data [8] increases over coming numerical data, this increasing occurs when data gets away from the centre of data, i.e. the mean of data distribution. Therefore, when data is close to the centre of data, the number of strangeness data decrease, the p_value increases and the martingale value [8] would not be large enough to detect these kinds of changes. In contrast, such changes can be detected in *TrackChCDF* because it analyzes the behavior of *G* sequences by obtaining its CDF. If the sequences of G approach to the smaller value, the CDF of G sequences gets a small value and vice versa.

To explore the ability of *TrackChCDF*, when the dimension of instances increases, we increase the number of dimensions in the studied data set and investigate the behavior of our method against Ho's method. Figure 6 illustrates the Precision of *TrackChCDF* and Ho's method in different dimensions. The horizontal axis shows the dimension of data set and the vertical one represents the mean of Precision measurements in 50 independent runs respectively. Both methods have high Precision. It means that when they alarm the existence of a change point, it is a true change point and they have low false alarm in average.

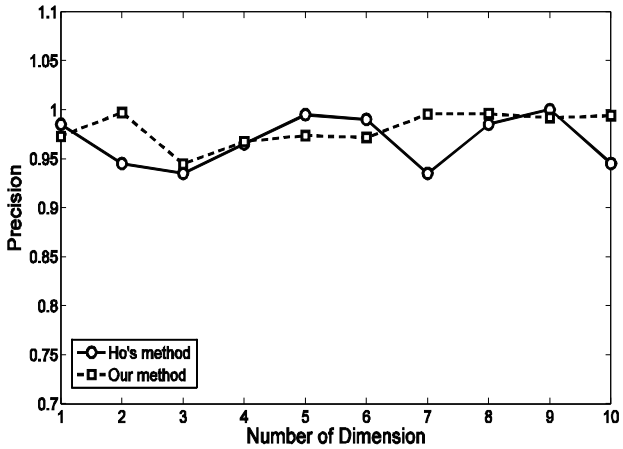

FIGURE 6: the accuracy of *TrackChCDF* and Ho's method in different number of dimensions. The horizontal axis shows dimension of data set and the vertical one represents the mean of precision measurements in 50 run respectively.

Figure 7 illustrates the Recall of *TrackChCDF* in comparison to Ho's method in different dimensions. The horizontal axis shows the dimensions of data set and the vertical one represents the mean of Recall measurements in 50 independent runs respectively. Ho's method can only detect those change points where the data get away from the mean of data distribution. In other words, it just detects the change points when the underlying data distribution will be open along the time. Thus, it detects the first three change points and it cannot detect the change points

when the distribution of data gets near the mean of data, namely closing change types. So, Ho's method cannot detect the fourth and fifth change points. Also, that method cannot detect the sixth change point in the instances of 12000 in spite of its type, *opening*, because this change occurs after two closing change points taking place in instances 8000 and 10000 respectively. It should be mentioned that in Ho's method, changes can be detected when p_values [10] get small. The p_values will be small when the number of strangeness data [8] increases through coming numerical data, this increasing occurs when data gets away from the center of data, i.e. the mean of data distribution. Therefore, when two closing type changes occur sequentially and after that an *opening* change happens, in Ho's method the two closing change causes the p_value sequences to get high value and the next *opening* change is slow down the p_value sequences but this reduction is not enough to be able to show this type of change in this manner. In contrast, *TrackChCDF* can detect such change precisely because it analyzes the behavior of G sequences by obtaining its CDF. Consequently, we can easily monitor the behavior of data distribution, including *opening* and closing changes happens in any combination, along the time. If the sequences of G approach to the smaller value, the CDF of G gets a small value and vice versa. Therefore, *TrackChCDF* has a higher Recall in comparison to Ho's method.

As Precision and Recall are related by F1, if F1 measure gets high value, we can ensure that Precision and Recall are reasonably high. Figure 8 illustrates the mean of F1 in 50 time experiments. Our method has higher F1 in comparison to Ho's method because Recall of our method is significantly higher that Ho's method.

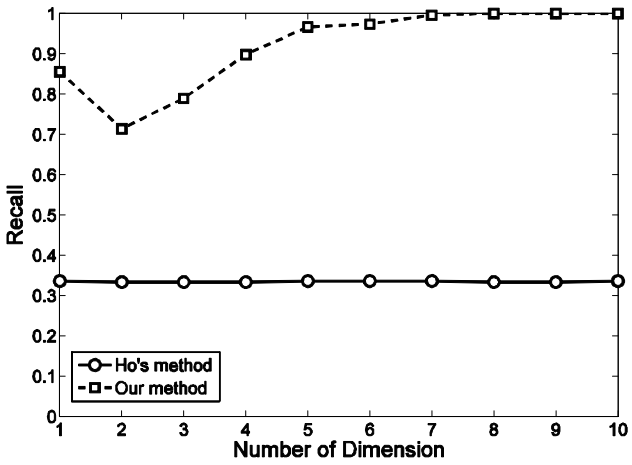

FIGURE 7: the accuracy of *TrackChCDF* and Ho's method in different number of dimensions. The horizontal axis shows dimension of data set and the vertical one represents the mean of recall measurements in 50 run respectively.

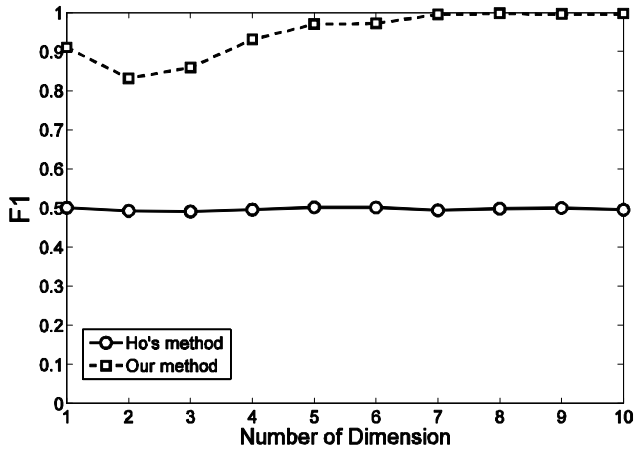

FIGURE 8: the accuracy of *TrackChCDF* and Ho's method in different number of dimensions. The horizontal axis shows dimension of data set and the vertical one represents the mean of F1 measurements in 50 run respectively.

To show the effectiveness of our method in the other point of view, we compare our method with Ho's method by calculating the mean delay time. This measure, i.e. delay time, shows the difference between the true time of occurrence of a change and the time in which ours can alarm an existence of that change. In other words, it shows how much delay time the underlying method has. Figure 9 shows the mean delay time of *TrackChCDF* in comparison to Ho's method. It should be mentioned that, for all ten experiments in each method, we perform 50 trials for each of these ten experiments. In this figure, we observe the mean delay time decreases with increasing the number of dimensions. With increasing the number of dimensions, the amount of changes increases because change applies in each dimension.

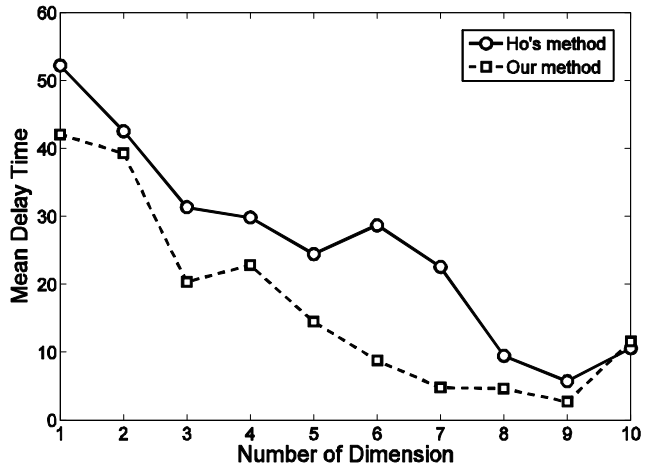

FIGURE 9: the accuracy of *TrackChCDF* and Ho's method in different number of dimensions. The horizontal axis shows dimension of data set and the vertical one represents the mean of Mean delay time in 50 run respectively.

Finally, it could be said that our method is able to determine the type of change in data stream. If the difference of two consecutive slopes of CDF is greater than a threshold, it means that there is a change and also this indicates that instances are going to be closed along the time. Also, if the difference of two successive slopes of CDF is smaller than a threshold, there exists a concept change in the distribution of data generating model and the instances are going to be opened along the time. According to this property, unlike Ho's method, *TrackChCDF* is able to determine not only the existence of change but also the type of such change, being closing or *opening* type.

5. CONSLUSION & FUTURE WORK

Recently data streams have been attractive research due to appearance of many applications in different domains. One of the most important challenges in data streams is concept change detection. There have been many researchers to detect concept change along data stream, however, majority of these researches devote to supervised domain where labels of instances are known a priori. Although data stream communities have recently focused on unsupervised domain, the proposed approaches are not yet matured to the point to be relied on. In other words, most of them provide merely a mediocre performance specially when applied on multi-dimensional data streams. In this paper, we propose a method for detecting change points along unlabeled data stream that is able to determine trend of changes in data streams as well. The abilities of our model can be enumerated as: (1) it is able to detect change points accurately. (2) It is able to report the behavior of instances along the time. (3) It works well in multi-dimensional data stream. We compare our algorithm to the state-of-the-art method for concept change detection in data streams and the obtained results are very promising. As a future work, we will incorporate our method into data clustering schemes.

6. REFERENCES

1. B. Babcock, S. Babu, R. Datar, R. Motwani and J. Widom. “Models and Issues in Data Stream Systems”, *in proceedings of ACM Symp, Principles of Databases Systems (PODS)*, pp. 1-16, 2002.

2. W. Fan. “Systematic Data Selection to Mine Concept Drifting Data Streams”, *in Proceedings of ACM SIGKDD*, pp. 128-137, 2004.

3. X. Liu, J.Guan, P. Hu. “Mining Frequent Closed Item Sets from a Landmark Window Over Online Data Streams”, *in journal of computers and mathematics with applications*, vol. 57, pp. 927-936, 2009.

4. C.C.Aggarwal, J. Han, J. Wang, P.S. Yu. “On Demand Classification of Data Streams”, *in proceedings of ACM SIGKDD*, pp. 503-508, 2004.

5. T. Jiang, Y. Feng, B. Zhang. “Online Detecting and Predicting Special Patterns over Financial Data Streams”, *in Journal of Universal Computer Science*, vol. 15, pp. 2566-2585, 2009.

6. J. Han, M. Kamber. “Data Mining: Concepts and Techniques”, *Morgan Kaufmann*, 2001.

7. O. Nasraoui, C. Rojas. “Robust Clustering for Tracking Noisy Evolving Data Streams”, *in Proceedings of Sixth SIAM International Conference of Data Mining (SDM)*, 2006.

8. S. S. Ho, H. Wechsler. “A Martingale Framework for Detecting Changes in Data Streams by Testing Exchangeability”, *in IEEE transactions on pattern analysis and machine intelligence*, 2010.

9. S. S. Ho. “A Martingale Framework for Concept Change Detection in Time Varying Data Streams”, *in Proceeding of 22th International Conference on Machine Learning*, L. D. Raedt and S. Wrobel, Eds., ACM, pp. 321–327, 2005.

10. S.-S. Ho, H. Wechsler. “Detecting Changes in Unlabeled Data Streams Using Martingale”, *in Proceeding 20th International Joint Conference on Artificial Intelligence*, M. Veloso, pp. 1912–1917, 2007.

11. P.J. Bickel, K. Doksum, “Mathematical Statistics: Basic Ideas and Selected Topics”, *Holden-Day*, Inc., 1977.

12. E. Carlstein, H. G. Muller, D. Siegmund editors. “Change point problems”, *Institute of Mathematical Statistics*, Hayward, California, 1994.

13. J. Glaz, N. Balakrishnan Editors. “Scan Statistics and Applications”, Boston, 1999.

14. J. Glaz, J. Naus, S. Wallenstein. “Scan Statistics”, Springer, New York, 2001.

15. R. Klinkenberg, T. Joachims. “Detecting Concept Drift with Support Vector Machines”, *in Proceedings of 17th International Conference on Machine Learning*, P. Langley, Ed. Morgan Kaufmann, pp. 487–494, 2000.

16. G. Widmer, M. Kubat. “Learning in the Presence of Concept Drift and Hidden Contexts”, *in Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996.

17. R. Klinkenberg. “Learning Drifting Concepts: Examples Selection VS Example Weighting”, *in Intelligent Data Analysis, Special Issue on Incremental Learning Systems capable of dealing with concept drift*, vol. 8, no. 3, pp. 281–300, 2004.

18. F. Chu, Y. Wang, C. Zaniolo. “*An Adaptive Learning Approach for Noisy Data Streams”*, *in proceedings of 4th IEEE international conference on Data Mining. IEEE Computer Society*, pp. 351–354, 2004

19. J. Z. Kolter, M. A. Maloof. “Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift*”, in proceedings of 3th IEEE international conference on Data Mining, IEEE Computer Society*, pp. 123–130, 2003.

20. H. Wang, W. Fan, P. S. Yu, J. Han. “Mining Concept Drifting Data streams Using Ensemble Classifiers”, *in proceedings of 9th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, L. Getoor, T. E. Senator, P. Domingos, and C. Faloutsos, Eds. ACM, pp. 226–235, 2003.

21. M. Scholz, R. Klinkenberg. “Boosting Classifiers for Drifting Concepts”, *in Intelligent Data Analysis*, vol. 11, no. 1, pp. 3-28, 2007.

22. O. Bousquet, M. Warmuth. “Tracking a Small Set of Experts by Mixing Past Posteriors”, *in Journal of Machine Learning Research*, vol. 3, pp. 363-396, 2002.

23. C. C. Aggarwal. “A framework for Change Diagnosis of Data Streams”, *in proceedings of ACM SIGMOD international conference on Management of Data*, pp. 575–586, 2003.

24. D. Kifer, S. Ben-David, J. Gehrke. “Detecting Change in Data Streams”, *in proceedings of 13th international conference on Very Large Data Bases*, M. A. Nascimento, M. T. O¨ zsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, Eds. Morgan Kaufmann, pp. 180–191, 2004.

25. T. Dasu, S. Krishnan, S. Venkatasubramanian, K. Yi. “An Information Theoretic Approach to Detecting Changes in Multi Dimensional Data Streams”, in Interface, 2006.

INSTRUCTIONS TO CONTRIBUTORS

Data Engineering refers to the use of data engineering techniques and methodologies in the design, development and assessment of computer systems for different computing platforms and application environments. With the proliferation of the different forms of data and its rich semantics, the need for sophisticated techniques has resulted an in-depth content processing, engineering analysis, indexing, learning, mining, searching, management, and retrieval of data.

International Journal of Data Engineering (IJDE) is a peer reviewed scientific journal for sharing and exchanging research and results to problems encountered in today's data engineering societies. IJDE especially encourage submissions that make efforts (1) to expose practitioners to the most recent research results, tools, and practices in data engineering topics; (2) to raise awareness in the research community of the data engineering problems that arise in practice; (3) to promote the exchange of data & information engineering technologies and experiences among researchers and practitioners; and (4) to identify new issues and directions for future research and development in the data & information engineering fields. IJDE is a peer review journal that targets researchers and practitioners working on data engineering and data management.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJDE.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 2, 2011, IJDE appears in more focused issues. Besides normal publications, IJDE intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJDE LIST OF TOPICS

The realm of International Journal of Data Engineering (IJDE) extends, but not limited, to the following:

- Approximation and Uncertainty in Databases and Pro
- Autonomic Databases
- Data Engineering
- Data Engineering Algorithms
- Data Engineering for Ubiquitous Mobile Distributed
- Data Engineering Models
- Data Integration
- Data Mining and Knowledge Discovery
- Data Ontologies
- Data Privacy and Security
- Data Query Optimization in Databases
- Data Streams and Sensor Networks
- Data Warehousing
- Database Tuning
- Database User Interfaces and Information Visualiza
- Knowledge Technologies
- Metadata Management and Semantic Interoperability
- OLAP and Data Grids
- Personalized Databases
- Query Processing in Databases
- Scientific Biomedical and Other Advanced Database
- Semantic Web
- Social Information Management
- Spatial Temporal

CALL FOR PAPERS

Volume: 2 - **Issue:** 3 - May 2011

i. Paper Submission: May 31, 2011 **ii. Author Notification:** July 01, 2011

iii. Issue Publication: July /August 2011

CONTACT INFORMATION

Computer Science Journals Sdn BhD
M-3-19, Plaza Damas Sri Hartamas
50480, Kuala Lumpur MALAYSIA

Phone: 006 03 6207 1607
006 03 2782 6991

Fax: 006 03 6207 1697

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2011
COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6207 1607
006 03 2782 6991

FAX: 006 03 6207 1697
EMAIL: cscpress@cscjournals.org