Using Latex to Write a Scientific Paper

Ran Wang
College of Mathematics and Statistics
Shenzhen University, China

- **Latex – Advantages for Typesetting**
- **Latex – Commands and Skills**

Advantages

- Authoritative format by many international academic institutions

  IEEE, Elsevier, Springer, IOS Press, IET Press, etc……

- High convenience

  math, formulas, symbols, tables, figures, references, etc……

- Small size for Tex file

  It only includes the first 128 characters in ASCII extended character set

- **How to design Latex macros and templates**

  .bst

  .cls

- **How to write a paper using an existing template**

  IEEE http://ieeeauthorcenter.ieee.org/create-your-ieee-article/use-authoring-tools-and-ieee-article-templates/ieee-article-templates/

  Elsevier http://www.latextemplates.com/template/elseviers-elsarticle-document-class

  Springer https://www.springer.com/cn/livingreviews/latex-templates

Segment Based Decision Tree Induction with Continuous Valued Attributes

Ran Wang, *Member, IEEE,* Sam Kwong, *Fellow, IEEE,* Xi-Zhao Wang, *Fellow, IEEE,* and Qingshan Jiang

***Abstract*—A key issue in decision tree (DT) induction with continuous valued attributes is to design an effective strategy for splitting nodes. Traditional approaches for solving this problem is by adopting the candidate cut point (CCP) with the highest discriminative ability, which is evaluated by some frequency based heuristic measures. However, such methods ignore the class permutation of examples in the node, and they cannot distinguish the CCPs with the same or similar frequency information, thus may fail to induce a better and smaller tree. In this paper, a new concept, i.e., *segment of examples*, is proposed to differentiate the CCPs with same frequency information. Then, a new hybrid scheme that combines the two heuristic measures, i.e., frequency and segment, is developed for splitting DT nodes. The relationship between frequency and the expected number of segments, which is regarded as a random variable, is also given. Experimental comparisons demonstrate that the proposed scheme is not only effective to improve the generalization capability, but also valid to reduce the size of the tree.**

***Index Terms*—Classification, continuous valued attributes, decision tree induction, segment.**

I. Introduction

INDUCTION of decision trees (DTs) is a technique of supervised learning, which builds up a knowledge-based expert system by inductive inference from examples. Due to a good interpretability and simple implementation, DTs have been utilized in various application domains such as fuzzy rule extraction [41], [11], [9], [19], ensemble learning [2], user authentication [37], anomaly detection [16], sample selection [40], monotonic classification [18], object ranking [20], and uncertainty analysis [38], etc. Recent developments of DTs could be found from the literature, such as multivariate DT [1], cost-sensitive DT [27], fuzzy DT [24], [25], [26], [32], geometric DT [28], and DT for handling continuous label [17].

DTs can easily produce some well-organized classification rules and have relatively low computational loads, thus are treated as powerful classification tools. As it is mentioned in [31], any effective methodology of supervised learning must have its inductive bias. The inductive bias of DT proposed by Quinlan [33] is that we prefer a smaller tree to a bigger tree when both of them are acceptable. This bias is supported by an old philosophic idea, i.e., Occam's razor [4], which clearly states that a model should be as simple as possible.

Traditional DT induction algorithms are typically designed for the data with symbolic/discrete valued attributes. For the one with continuous valued attributes, discretization must be conducted before or during the tree growth [5], [12]. Discretization before the tree growth is simple and easy to carry out, but the performance is poor since it neglects the relationship between the conditional attributes and the decision attribute. Discretization during the tree growth follows some guidelines given by the decision attribute, thus can achieve better performances. The main task in this kind of discretization is to split the currently chosen attribute into several intervals such that the discriminative ability on the training examples is high. Along this direction, one can further adopt binary splitting or multiple splitting [22], [29], which respectively divide the attribute into two or more intervals. The well-behavedness of multiple splitting have been demonstrated in many works [3], [13], [15], [6], but the inductive procedure is complex and the size of the induced tree is large. Thus, we only deal with the typical binary splitting in this paper. Obviously, the trees generated are of two branches.

The induction of DT is a recursive process that follows a top-down approach by repeated splits of the training set. Generally, there are two key issues during the tree growth:

- one is how to judge a leaf node;
- the other is how to split a non-leaf node [8].

Usually, a leaf node is determined if its class purity is higher than a given coefficient, or the number of examples in it is smaller than a given threshold. As for splitting a non-leaf node, the typical solution is to sort the examples according to each attribute, evaluate all the possible splits by a certain heuristic measure, and select the one with the highest discriminative ability. The earliest method is known as IDE3 [21], which selects the split with the highest information gain. However, IDE3 is specially designed for discrete attributes, and tends to select the one with more values, which may lead to the over-fitting problem. Thus, C4.5 [14], [33], [34] is proposed to improve IDE3, which replace the criterion of information gain by gain ratio, and is extended to handle both discrete and continuous attributes. Both IDE3 and C4.5 are based on the heuristic of information entropy [36]. Later, CART algorithm [7], [23] is proposed based on the heuristic of Gini-index by selecting the split that can maximally reduce the degree of sample disorder, and CHAID algorithm [43], [39] is proposed based on the CHi square detection. The

This work was supported by the National Natural Science Foundation of China (Grant No. 61272289, 61175123, and 61170040).

R. Wang and S. Kwong are with the Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong. (e-mail: ranwang3-c@my.cityu.edu.hk, CSSAMK@cityu.edu.hk).

X-Z. Wang is currently with the College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China. (e-mail: xizhaowang@ieee.org).

R. Wang and Q. Jiang are with the Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China. (e-mail: wangran@siat.ac.cn, qs.jiang@siat.ac.cn).

Segment Based Decision Tree Induction With Continuous Valued Attributes

Ran Wang, *Member, IEEE,* Sam Kwong, *Fellow, IEEE,* Xi-Zhao Wang, *Fellow, IEEE,* and Qingshan Jiang

***Abstract*—A key issue in decision tree (DT) induction with continuous valued attributes is to design an effective strategy for splitting nodes. The traditional approach to solving this problem is adopting the candidate cut point (CCP) with the highest discriminative ability, which is evaluated by some frequency based heuristic measures. However, such methods ignore the class permutation of examples in the node, and they cannot distinguish the CCPs with the same or similar frequency information, thus may fail to induce a better and smaller tree. In this paper, a new concept, i.e., segment of examples, is proposed to differentiate the CCPs with same frequency information. Then, a new hybrid scheme that combines the two heuristic measures, i.e., frequency and segment, is developed for splitting DT nodes. The relationship between frequency and the expected number of segments, which is regarded as a random variable, is also given. Experimental comparisons demonstrate that the proposed scheme is not only effective to improve the generalization capability, but also valid to reduce the size of the tree.**

***Index Terms*—Classification, continuous valued attributes, decision tree (DT) induction, segment.**

I. Introduction

INDUCTION of decision trees (DTs) is a technique of supervised learning, which builds up a knowledge-based expert system by inductive inference from examples. Due to a good interpretability and simple implementation, DTs have been utilized in various application domains such as fuzzy rule extraction [9], [11], [19], [41], ensemble learning [2], user authentication [37], anomaly detection [16], sample selection [40], monotonic classification [18], object ranking [20], and uncertainty analysis [38], etc. Recent developments of DTs could be found from the literature, such as multivariate DT [1], cost-sensitive DT [27], fuzzy DT [24]–[26], [32], geometric DT [28], and DT for handling continuous label [17].

DTs can easily produce some well-organized classification rules and have relatively low computational loads, thus are treated as powerful classification tools. As it is mentioned in [31], any effective methodology of supervised learning must have its inductive bias. The inductive bias of DT proposed by Quinlan [33] is that we prefer a smaller tree to a bigger tree when both of them are acceptable. This bias is supported by an old philosophic idea, i.e., Occam's razor [4], which clearly states that a model should be as simple as possible.

Traditional DT induction algorithms are typically designed for the data with symbolic/discrete valued attributes. For the one with continuous valued attributes, discretization must be conducted before or during the tree growth [5], [12]. Discretization before the tree growth is simple and easy to carry out, but the performance is poor since it neglects the relationship between the conditional attributes and the decision attribute. Discretization during the tree growth follows some guidelines given by the decision attribute, thus can achieve better performances. The main task in this kind of discretization is to split the currently chosen attribute into several intervals such that the discriminative ability on the training examples is high. Along this direction, one can further adopt binary splitting or multiple splitting [22], [29], which respectively divide the attribute into two or more intervals. The well-behavedness of multiple splitting have been demonstrated in many works [3], [6], [13], [15], but the inductive procedure is complex and the size of the induced tree is large. Thus, we only deal with the typical binary splitting in this paper. Obviously, the trees generated are of two branches.

The induction of DT is a recursive process that follows a top-down approach by repeated splits of the training set. Generally, there are two key issues during the tree growth.

1) One is how to judge a leaf node.
2) The other is how to split a nonleaf node [8].

Usually, a leaf node is determined if its class purity is higher than a given coefficient, or the number of examples in it is smaller than a given threshold. As for splitting a nonleaf node, the typical solution is to sort the examples according to each attribute, evaluate all the possible splits by a certain heuristic measure, and select the one with the highest discriminative ability. The earliest method is known as IDE3 [21], which selects the split with the highest information gain. However, IDE3 is specially designed for discrete attributes, and tends

Manuscript received December 13, 2013; revised March 2, 2014 and July 30, 2014; accepted August 6, 2014. Date of publication September 29, 2014; date of current version June 12, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61272289, Grant 61175123, and Grant 61170040. This paper was recommended by Associate Editor J. Basak.

R. Wang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: ranwang3-c@my.cityu.edu.hk; wangran@siat.ac.cn).

S. Kwong is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: cssamk@cityu.edu.hk).

X.-Z. Wang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: xizhaowang@ieee.org).

Q. Jiang is with the Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: qs.jiang@siat.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2348012

A Vector-valued Support Vector Machine Model for Multiclass Problem

Ran Wang[a], Sam Kwong[a,*], Degang Chen[b], Jingjing Cao[a]

[a]Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong.
[b]Department of Mathematics and Physics, North China Electric Power University, 102206 Beijing, PR China.

Abstract

In this paper, a new model named Multiclass Support Vector Machines with Vector-Valued Decision (M-SVMs-VVD) or VVD is proposed. The basic idea is to separate 2^a classes by a SVM hyperplanes in the feature space induced by certain kernels, where a is a finite positive integer. We start from a 2^a-class problem, and extend it to any-class problem by applying a hierarchical decomposition procedure. Compared with the existing SVM-based multiclass methods, the VVD model has two advantages. First, it reduces the computational complexity by using a small number of classifiers. Second, the feature space partition induced by the hyperplanes effectively eliminates the unclassifiable regions (URs) that may affect the classification performance of the algorithm. Experimental comparisons with several state-of-the-art multiclass methods demonstrate that VVD maintains a comparable testing accuracy, while it improves the classification efficiency with less classifiers, a smaller number of support vectors (SVs), and shorter testing time.

Keywords: Feature space, hyperplane, multiclass classification, support vector machines (SVMs), unclassifiable regions (URs)

1. Introduction

Support vector machines (SVMs) [15, 37, 40] have attracted significant interest during recent years and have exhibited outstanding performances

*Corresponding author. *E-mail address:* CSSAMK@cityu.edu.hk.

Information Sciences 235 (2013) 174–194

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A vector-valued support vector machine model for multiclass problem

Ran Wang[a], Sam Kwong[a,*], Degang Chen[b], Jingjing Cao[a]

[a]Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong
[b]Department of Mathematics and Physics, North China Electric Power University, 102206 Beijing, PR China

ARTICLE INFO

Article history:
Received 1 November 2011
Received in revised form 1 January 2013
Accepted 3 February 2013
Available online 20 February 2013

Keywords:
Feature space
Hyperplane
Multiclass classification
Support vector machine (SVM)
Unclassifiable region (UR)

ABSTRACT

In this paper, a new model named Multiclass Support Vector Machines with Vector-Valued Decision (M-SVMs-VVD) or VVD is proposed. The basic idea is to separate 2^a classes by a SVM hyperplanes in the feature space induced by certain kernels, where a is a finite positive integer. We start from a 2^a-class problem, and extend it to any-class problem by applying a hierarchical decomposition procedure. Compared with the existing SVM-based multiclass methods, the VVD model has two advantages. First, it reduces the computational complexity by using a small number of classifiers. Second, the feature space partition induced by the hyperplanes effectively eliminates the Unclassifiable regions (URs) that may affect the classification performance of the algorithm. Experimental comparisons with several state-of-the-art multiclass methods demonstrate that VVD maintains a comparable testing accuracy, while it improves the classification efficiency with less classifiers, a smaller number of support vectors (SVs), and shorter testing time.

1. Introduction

Support vector machines (SVMs) [15,37,40] have attracted significant interest during recent years and have exhibited outstanding performances on various learning domains, such as data mining [41], syndrome recognition [16], hidden Markov modeling [30], protein prediction [18], and financial time series forecasting [35]. Traditional SVMs are typically designed for binary classification problems. Extending them to multiclass classification remains a hot topic [19,22]. There are two kinds of approaches for multiclass SVMs (M-SVMs). One is to form a multiclass classifier by combining several binary SVMs that are trained separately based on some class decomposition schemes. The other is to train all SVMs simultaneously by designing a single objective function known as the "all-together" method [1,4,12]. The objective function in this "all-together" method allows us to address the importance/influence of different SVMs, however, it is time consuming and sometimes impractical. Thus, we have mainly focused on the first kind of approach in this paper.

Vapnik [37] first proposed the one-against-all (OAA) method which decomposes an $\mathcal{A}$-class problem into $\mathcal{A}$ binary problems, where $\mathcal{A}$ is a finite positive integer and $\mathcal{A} \geqslant 3$. This method has been very popular, but suffers from high training expenses and the unclassifiable region (UR) problem. Inoue et al. [14] proposed the membership function to identify the unclassifiable patterns of OAA, and Ryan et al. [28] presented the comparable performance of OAA against other approaches, but the high training expenses remain as a problem yet to be solved.

Friedman [11] later raised the one-against-one (OAO) approach which solves an $\mathcal{A}$-class problem via $\mathcal{A}(\mathcal{A}-1)/2$ binary classifiers by adopting a pairwise method. OAO has a faster training speed than OAA, but it also suffers from the UR problem

* Corresponding author.
E-mail address: CSSAMK@cityu.edu.hk (S. Kwong).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.02.001

Noname manuscript No.
(will be inserted by the editor)

Multi-Criteria Decision Making Based Architecture Selection for Single-Hidden Layer Feedforward Neural Networks

Ran Wang · Haoran Xie · Jiqiang Feng · Fu Lee Wang · Chen Xu

Received: date / Accepted: date

Abstract Architecture selection is a fundamental problem in artificial neural networks, which could be treated as a decision making process that evaluates, ranks, and makes choices from a set of network structures. Traditional methods evaluate a network structure by designing a criterion based on a validation model or an error bound model. On one hand, the time complexity of a validation model is usually high; on the other hand, different validation models or error bound models may lead to different (even conflicting) results, which post challenges to the traditional single criterion-based architecture selection methods. In the area of decision making, many problems employed multiple criteria since the performance is better than using a single criterion. In this paper, we propose a multi-criteria decision making based architecture selection algorithm for single-hidden layer feedforward neural networks trained by extreme learning machine. Two criteria are incorporated into the selection process, i.e., training accuracy and the Q-value estimated by the localized generalization error model. The training accuracy reflects the capability of the model on correctly categorizing the known samples, and the Q-value estimated by localized generalization error model reflects the size of the neighbourhood of training samples in which the model can predict unseen samples with confidence. By achieving a trade-off between these two criteria, a new architecture selection algorithm is proposed. Experimental comparisons demonstrate the feasibility and effectiveness of the proposed method.

Keywords Architecture selection · Extreme learning machine · Localized generalization error model · Multi-criteria decision making.

1 Introduction

Artificial neural network (ANN) [1,2] is a commonly used technique for solving classification problems. Single-hidden layer feedforward neural network (SLFN), as the simplest type of ANN, has been proved to have the universal approximation capability on any given data set. In recent years, a fast training method for SLFN named extreme learning machine (ELM) [3] has been proposed, which largely reduces the computational complexity of ANN and enlarges its application domains [4–7]. In order to construct a high-performance ELM, architecture selection [8–11] is an inevitably necessary step. In other words, for a given data set, it is necessary to select an appropriate network structure (i.e., the number of hidden nodes) to guarantee a satisfactory performance of the learning model. Architecture selection is important in many ANN based applications such as time series forecasting [12], manufacturing [13], quantum computer [14,15], gas metering system [16], and video object classification [17], etc.

For a classification model, training accuracy reflects its capability of correctly classifying the known samples, which is important for evaluating a network structure. However, the training set is usually a small subset of

R. Wang · J. Feng · C. Xu
College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, China.
E-mail: wangran@szu.edu.cn, fengjq@szu.edu.cn, xuchen@szu.edu.cn.

H. Xie
Department of Mathematics and Information Technology, The Education University of Hong Kong.
E-mail: hrxie2@gmail.com.

F. L. Wang
Caritas Institute of Higher Education, Hong Kong.
E-mail: pwang@cihe.edu.hk.

International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-017-0746-9

ORIGINAL ARTICLE

Multi-criteria decision making based architecture selection for single-hidden layer feedforward neural networks

Ran Wang[1] · Haoran Xie[2] · Jiqiang Feng[1] · Fu Lee Wang[3] · Chen Xu[1]

Received: 7 April 2017 / Accepted: 14 November 2017

Abstract

Architecture selection is a fundamental problem in artificial neural networks, which could be treated as a decision making process that evaluates, ranks, and makes choices from a set of network structures. Traditional methods evaluate a network structure by designing a criterion based on a validation model or an error bound model. On one hand, the time complexity of a validation model is usually high; on the other hand, different validation models or error bound models may lead to different (even conflicting) results, which post challenges to the traditional single criterion-based architecture selection methods. In the area of decision making, many problems employed multiple criteria since the performance is better than using a single criterion. In this paper, we propose a multi-criteria decision making based architecture selection algorithm for single-hidden layer feedforward neural networks trained by extreme learning machine. Two criteria are incorporated into the selection process, i.e., training accuracy and the Q-value estimated by the localized generalization error model. The training accuracy reflects the capability of the model on correctly categorizing the known samples, and the Q-value estimated by localized generalization error model reflects the size of the neighbourhood of training samples in which the model can predict unseen samples with confidence. By achieving a trade-off between these two criteria, a new architecture selection algorithm is proposed. Experimental comparisons demonstrate the feasibility and effectiveness of the proposed method.

Keywords Architecture selection · Extreme learning machine · Localized generalization error model · Multi-criteria decision making

1 Introduction

Artificial neural network (ANN) [1, 2] is a commonly used technique for solving classification problems. Single-hidden layer feedforward neural network (SLFN), as the simplest type of ANN, has been proved to have the universal approximation capability on any given data set. In recent years, a fast training method for SLFN named extreme learning machine (ELM) [3] has been proposed, which largely reduces the computational complexity of ANN and enlarges its application domains [4–7]. In order to construct a high-performance ELM, architecture selection [8–11] is an inevitably necessary step. In other words, for a given data set, it is necessary to select an appropriate network structure (i.e., the number of hidden nodes) to guarantee a satisfactory performance of the learning model. Architecture selection is important in many ANN based applications such as time series forecasting [12], manufacturing [13], quantum computer [14, 15], gas metering system [16], and video object classification [17], etc.

Ran Wang
wangran@szu.edu.cn

Haoran Xie
hrxie2@gmail.com

Jiqiang Feng
fengjq@szu.edu.cn

Fu Lee Wang
pwang@cihe.edu.hk

Chen Xu
xuchen@szu.edu.cn

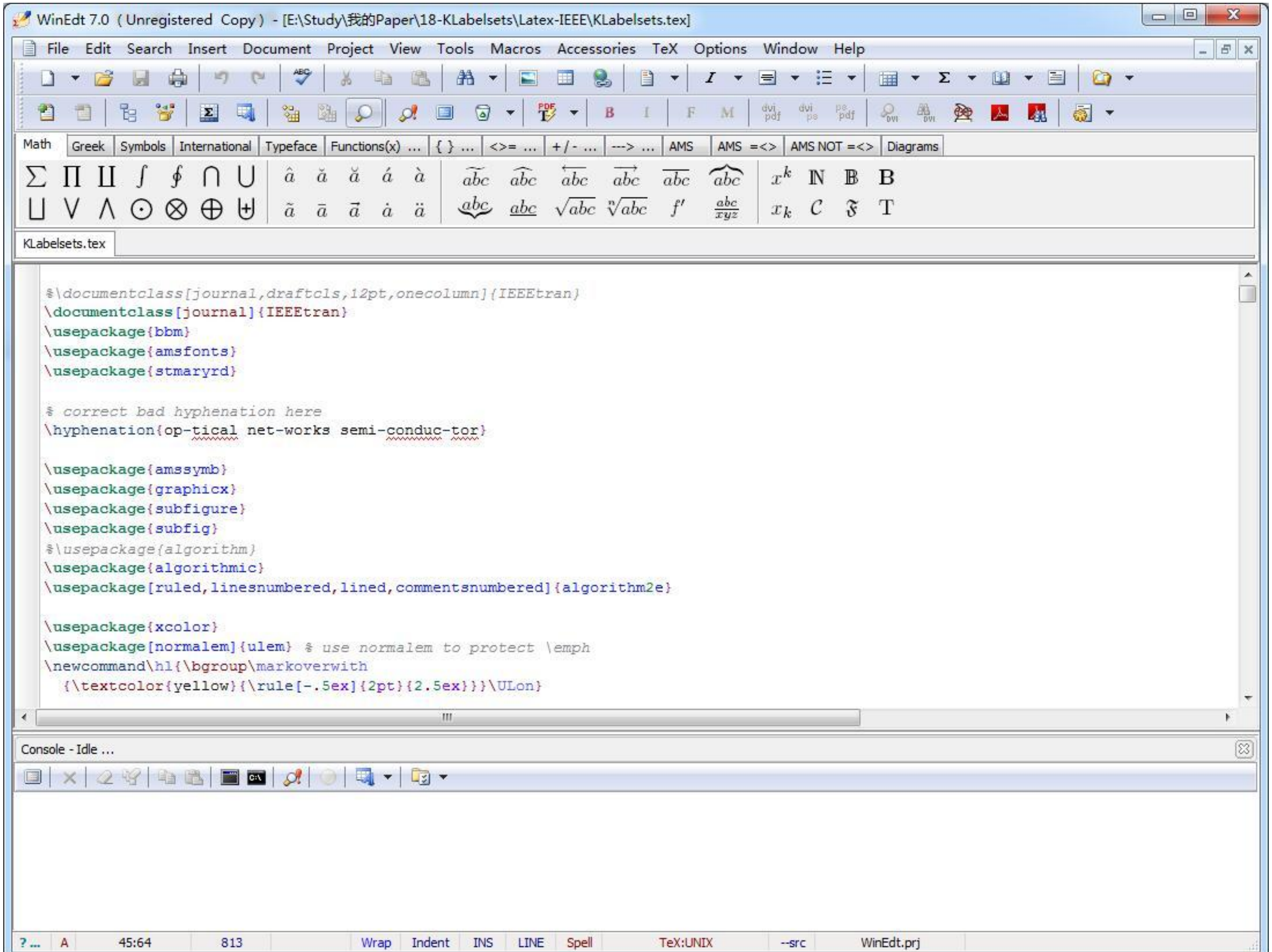
1 College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, China

2 Department of Mathematics and Information Technology, The Education University of Hong Kong, Hong Kong, Hong Kong

3 Caritas Institute of Higher Education, Hong Kong, Hong Kong

Published online: 18 November 2017

CTex, LyX, TexMaker, TeXstudio ……



Start from the title page

```latex
\begin{document}

\title{TaxiRec: Recommending Road Clusters to Taxi Drivers Using Ranking-based Extreme Learning Machines}

\author{Ran Wang,~\IEEEmembership{Member,~IEEE,}
        Chi-Yin Chow,~\IEEEmembership{Senior Member,~IEEE,}
        Yan Lyu,
        Victor C. S. Lee,~\IEEEmembership{Member,~IEEE,}\\
        Sam Kwong,~\IEEEmembership{Fellow,~IEEE,}
        Yanhua Li,~\IEEEmembership{Senior Member,~IEEE,}
        and~Jia Zeng,~\IEEEmembership{Senior Member,~IEEE}% <-this % stops a space

\IEEEcompsocitemizethanks{\IEEEcompsocthanksitem Ran Wang is with the College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, China.\protect\\
% note need leading \protect in front of \\ to get a newline within \thanks as
% \\ is fragile and will error, could use \hfil\break instead.
E-mail:~wangran@szu.edu.cn.
\IEEEcompsocthanksitem Chi-Yin Chow, Yan Lyu, Victor C. S. Lee, and Sam Kwong are with the Department of Computer Science, City University of Hong Kong, Hong Kong.\protect\\
E-mail:~chiychow@cityu.edu.hk, yanlv2-c@my.cityu.edu.hk, csvlee@cityu.edu.hk, and cssamk@cityu.edu.hk.
\IEEEcompsocthanksitem Yanhua Li is with the Department of Computer Science, Worcester Polytechnic Institute
Worcester, MA, USA.\protect\\
Email:~yli15@wpi.edu.
\IEEEcompsocthanksitem Jia Zeng is with Huawei Noah's Ark Lab, Hong Kong.\protect\\
Email:~zeng.jia@huawei.com.}
\thanks{Manuscript received XX XX, 2016; revised XX XX, 2016.}}


\IEEEtitleabstractindextext{%
\begin{abstract}
Utilizing large-scale GPS data to improve taxi services has become a popular research problem in the areas of data mining, intelligent transportation, geographical information systems, and the Internet of Things. In this paper, we utilize a large-scale GPS data set generated by over 7,000 taxis in a period of one month in Nanjing, China, and propose TaxiRec: a framework for evaluating and discovering the passenger-finding potentials of road clusters, which is incorporated into a recommender system for taxi drivers to seek passengers. In TaxiRec, the underlying road network is first segmented into a number of road clusters, a set of features for each road cluster is extracted from real-life data sets, and then a ranking-based extreme learning machine (ELM) model is proposed to evaluate the passenger-finding potential of each road cluster. In addition, TaxiRec can use this model with a training cluster selection algorithm to provide road cluster recommendations when taxi trajectory data is incomplete or unavailable. Experimental results demonstrate the feasibility and effectiveness of TaxiRec.
\end{abstract}

% Note that keywords are not normally used for peerreview papers.
\begin{IEEEkeywords}
Extreme learning machine, passenger-finding potential, recommender system, taxi trajectory data analytics.
\end{IEEEkeywords}}

\maketitle
```

TaxiRec: Recommending Road Clusters to Taxi Drivers Using Ranking-Based Extreme Learning Machines

Ran Wang, *Member, IEEE*, Chi-Yin Chow, *Senior Member, IEEE*, Yan Lyu,
Victor C. S. Lee, *Member, IEEE*, Sam Kwong, *Fellow, IEEE*,
Yanhua Li, *Senior Member, IEEE*, and Jia Zeng, *Senior Member, IEEE*

Abstract—Utilizing large-scale GPS data to improve taxi services has become a popular research problem in the areas of data mining, intelligent transportation, geographical information systems, and the Internet of Things. In this paper, we utilize a large-scale GPS data set generated by over 7,000 taxis in a period of one month in Nanjing, China, and propose TaxiRec: a framework for evaluating and discovering the passenger-finding potentials of road clusters, which is incorporated into a recommender system for taxi drivers to seek passengers. In TaxiRec, the underlying road network is first segmented into a number of road clusters, a set of features for each road cluster is extracted from real-life data sets, and then a ranking-based extreme learning machine (ELM) model is proposed to evaluate the passenger-finding potential of each road cluster. In addition, TaxiRec can use this model with a training cluster selection algorithm to provide road cluster recommendations when taxi trajectory data is incomplete or unavailable. Experimental results demonstrate the feasibility and effectiveness of TaxiRec.

Index Terms—Extreme learning machine, passenger-finding potential, recommender system, taxi trajectory data analytics

Latex Commands and Skills

- Latex Characters

- Latex Environments

- Latex Commands

1、 Character Set

Extended Characters: $, &, ~, \, %, {, }, ^, _, #

Math Characters: the characters that can be printed in math mode **only**

- math mode “$ characters $”

Special characters $, &, ~, \, %, {, }, ^, _, #...... have special meanings for control sequence, thus can not be generated by direct typing.

1) $: math mode, must be in pair like α.

2) &: delimiter for (tabular) or (array)

3) ~: space

4) **#: variable parameters in macro commands, e.g., #1, #2 in macros, we can replace #1, #2 by other characters when calling the macros**

5) **^: superscript,** "$2\wedge x$" → 2^x

6) **_: subscript,** "2_x" → 2_x

7) **\ : Latex command**

8) **{ : Latex scope initiator**

} : Latex scope terminator

9) **%: comment**

11） **** : move to the beginning of the next line

12） **\par** : paragraph separator

13） Latex size metrics

- 英寸(in),
- 厘米(cm),
- 毫米(mm),
- 点(pt)   1/72.27 in

2、 Environment Set

1. Tex environment
2. Table and figure environment
3. Math environment

```
\begin{name}[options]
        …content…
\end{name}
```

Tex Environment

- 1) center

  \begin{center} …\end{center}

  Center mode
- 2) flushleft

  \begin{flushleft} … \end{flushleft}

  Left alignment
- 3) flushright

  \begin{flushright} … \end{flushright}

  Right alignment

4) Itemize

```
\begin{itemize}
   \item~<item 1>
   \item~<item 2>
       ........
   \item~<item n>
\end{itemize}
```

- The mathematical model, which has a direct impact on both the training accuracy and testing accuracy.
- The algorithm for training the model parameters, which is sensitive to the prediction results.
- The data distribution. In supervised learning, there is a fundamental assumption that the training data has the same distribution as the testing data. The learning scheme that does not follow this fundamental assumption is referred to as transfer learning [19], which is out of the scope of this paper.

5) Enumerate

```
\begin{enumerate}
    \item~<item 1>
        \begin{enumerate}
            \item~<item 1>
        \end{enumerate}
    \item~<item 2>
    ........
    \item~<item n>
\end{enumerate}
```

Multiple factors have critical impacts on the generalization of a classifier.

1) Model selection. It is hard to select the most appropriate model for a given classification task. When the training data is fixed, the generalizations of two models might be quite different. This is due to the data distribution, i.e., a model suitable for one type of data may not be appropriate for another type of data.
2) Training algorithm. When a model is fixed, the subsequent work is to train the model parameters based on a training set. A model with a set of trained parameters has the generalization quite different from the model with another set of trained parameters.
3) Representatives of training data. Since both the objective function and its approximating function are defined on a space $\mathcal{S}$, one problem is that the training set $\mathbb{X}$ should be a reasonable sampling of the space $\mathcal{S}$, which directly relates to the fundamental assumption of machine learning that the training set has an identical distribution as the testing set has.
4) Model knowledge parameters. Different from the parameters inside the model that are acquired directly from the training process, model knowledge parameters do not explicitly appear in the model, which are usually evaluated after the training process. For example, the uncertainty of classifier's outputs is a typical model knowledge parameter. The relationship between generalization and uncertainty of a classifier is initially demonstrated in [20]. This paper will conduct further studies on this relationship through incorporating a new index, i.e., complexity of classification.

Table and Figure Environment

1) tabbing

Create a table without frame:

```
\begin {tabbing}
    <content>
\end {tabbing}
```

table

- \begin{table}[position]
  <content>
  \end{table}

Position: h(here), b(bottom), t(top), p(another page)

Title: \caption{title}

2) tabular

Create a table with various kinds of frames:

```
\begin{tabular}{alignment mode}
        …content…
\end{tabular}
```

Left alignment (*l*)
Right alignment (r)
Center (c)

```latex
\begin{table*}[t]
\caption{Data Sets for Performance Comparison}
\begin{center}
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
\textbf{Data Sets} &\textbf{\textit{Domain}} &\textbf{\textit{\# Training Samples}} &\textbf{\textit{\# Testing Samples}}
&\textbf{\textit{\# Nominal Features}} &\textbf{\textit{\# Numeric Features}} &\textbf{\textit{\# Labels}}
&\textbf{\textit{Label Cardinality}}\\
\hline
birds       &audio        &322    &323    &2      &258    &19     &1.014\\
CAL500      &music        &502    &---    &0      &68     &174    &26.044\\
emotions    &music        &391    &202    &0      &72     &6      &1.869\\
enron       &text         &1,123  &579    &1,001  &0      &53     &3.378\\
flags       &images (toy) &129    &65     &9      &10     &7      &3.392\\
genbase     &biology      &463    &199    &1,186  &0      &27     &1.252\\
medical     &text         &333    &645    &1,449  &0      &45     &1.245\\
rcv1v2      &text         &3,000  &3,000  &0      &47,236 &101    &2.880\\
scene       &image        &1,211  &1,196  &0      &294    &6      &1.074\\
yeast       &biology      &1,500  &917    &0      &103    &14     &4.237\\
\hline\noalign{\smallskip}
\end{tabular}
\begin{minipage}{17.5cm}
\small{\textbf{Note:} The original data set {\it CAL500} has not separated the training and testing sets, we randomly select
half of the samples as the training set and the remaining half samples as the testing set. {\it Label Cardinality}
represents the average number of positive labels per sample.}
\end{minipage}
\label{tab.datasets}
\end{center}
\end{table*}
```

TABLE III
DATA SETS FOR PERFORMANCE COMPARISON

Data Sets	***Domain***	***# Training Samples***	***# Testing Samples***	***# Nominal Features***	***# Numeric Features***	***# Labels***	***Label Cardinality***
birds	audio	322	323	2	258	19	1.014
CAL500	music	502	—	0	68	174	26.044
emotions	music	391	202	0	72	6	1.869
enron	text	1,123	579	1,001	0	53	3.378
flags	images (toy)	129	65	9	10	7	3.392
genbase	biology	463	199	1,186	0	27	1.252
medical	text	333	645	1,449	0	45	1.245
rcv1v2	text	3,000	3,000	0	47,236	101	2.880
scene	image	1,211	1,196	0	294	6	1.074
yeast	biology	1,500	917	0	103	14	4.237

Note: The original data set *CAL500* has not separated the training and testing sets, we randomly select half of the samples as the training set and the remaining half samples as the testing set. *Label Cardinality* represents the average number of positive labels per sample.

3) picture

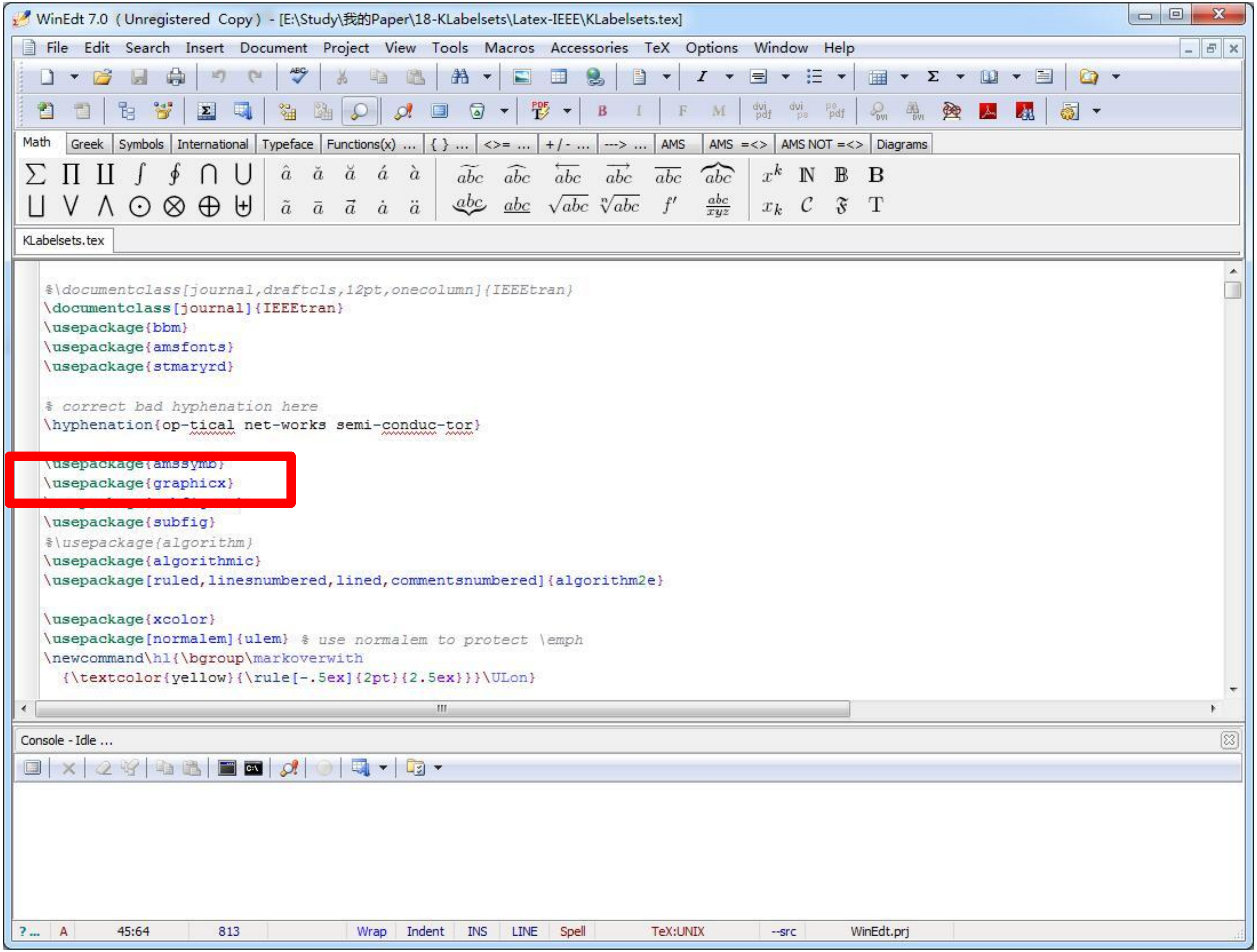
You can use Latex drawing circles, lines, vectors, and rounded rectangles to create simple lines and curves.

```
\begin{picture}(num1,num2)
        …graphics commands…
\end{picture}
```

\usepackage{graphicx}

.eps or .pdf will be the best format for figures

\usepackage{graphicx}



```
\begin{figure}[htbp]\centering
  \includegraphics[width=0.35\textwidth]{measureCurves.eps}
\caption{Relationship between measure values and frequency of positive class.}
\label{fig:frequencyMeasures}
\end{figure}
```

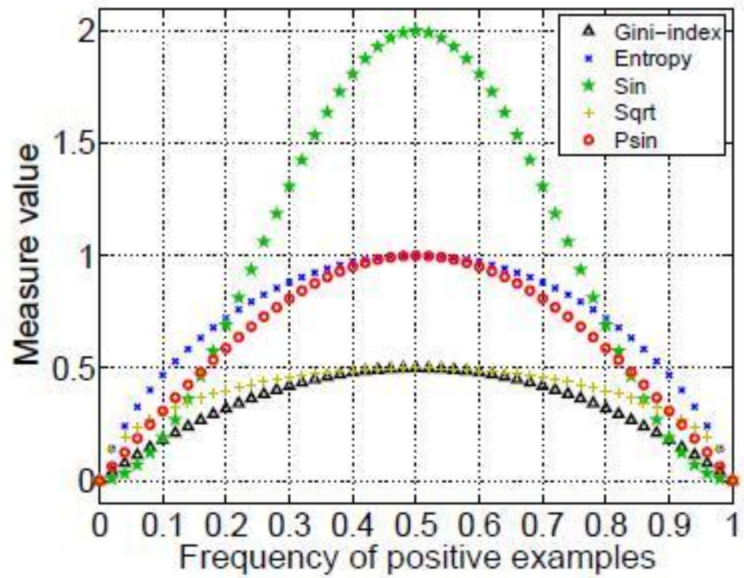
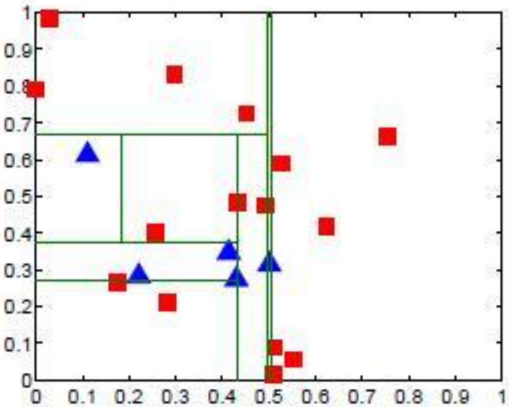
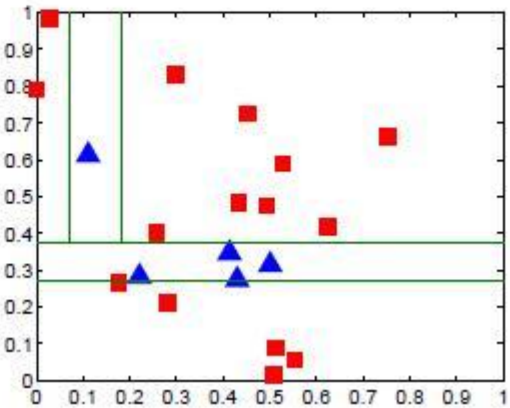


Fig. 1: Relationship between measure values and frequency of positive class.

```latex
\begin{figure}[htbp]
\centering
\subfigure[DT of C4.5]{\includegraphics[width=1.7in]{2DexampleC45.eps}}
\subfigure[DT of Segment+C4.5]{\includegraphics[width=1.7in]{2DexampleProposed.eps}}
\caption{Difference between DTs induced by C4.5 and the proposed algorithm on the 2-dimensional
dataset.}\label{fig.2DExample}
\end{figure}
```



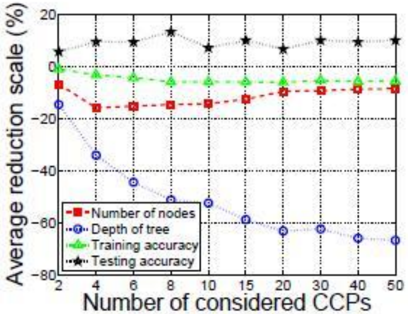
(a) DT of C4.5



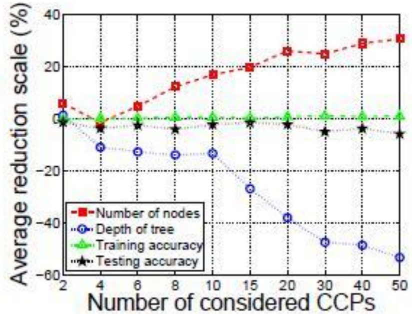
(b) DT of Segment+C4.5

Fig. 4: Difference between DTs induced by C4.5 and the proposed algorithm on the 2-dimensional dataset.

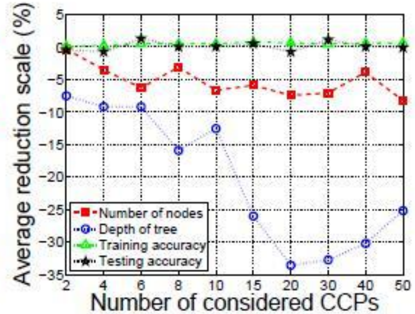
```latex
\begin{figure*}[htbp]
\centering
\subfigure[Haberman]{\includegraphics[width=1.7in]{habermanC45.eps}}
\subfigure[Ionosphere]{\includegraphics[width=1.7in]{ionosphereC45.eps}}
\subfigure[Cancer]{\includegraphics[width=1.7in]{cancerC45.eps}}
\subfigure[Australian]{\includegraphics[width=1.7in]{australianC45.eps}}
\subfigure[German]{\includegraphics[width=1.7in]{germanC45.eps}}
\subfigure[Bupa]{\includegraphics[width=1.7in]{bupaC45.eps}}
\subfigure[Heart]{\includegraphics[width=1.7in]{heartC45.eps}}
\subfigure[Transfusion]{\includegraphics[width=1.7in]{transfusionC45.eps}}
\subfigure[Wdbc]{\includegraphics[width=1.7in]{wdbcC45.eps}}
\subfigure[Wpbc]{\includegraphics[width=1.7in]{wpbcC45.eps}}
\subfigure[Pima]{\includegraphics[width=1.7in]{pimaC45.eps}}
\subfigure[Plrx]{\includegraphics[width=1.7in]{plrxC45.eps}}
\subfigure[SPECTF]{\includegraphics[width=1.7in]{SPECTFC45.eps}}
\subfigure[CT]{\includegraphics[width=1.7in]{CTC45.eps}}
\subfigure[Sonar]{\includegraphics[width=1.7in]{sonarC45.eps}}
\subfigure[Cotton]{\includegraphics[width=1.7in]{cottonC45.eps}}
\subfigure[Ecoli]{\includegraphics[width=1.7in]{ecoliC45.eps}}
\subfigure[Libras]{\includegraphics[width=1.7in]{librasC45.eps}}
\subfigure[Vowel]{\includegraphics[width=1.7in]{vowelC45.eps}}
\subfigure[Yeast]{\includegraphics[width=1.7in]{yeastC45.eps}}
\caption{Average reduction scale in accuracy and tree size of method \textit{Segment+C4.5} compared with method
\textit{C4.5}.}\label{fig.reductionEntro}
\end{figure*}
```



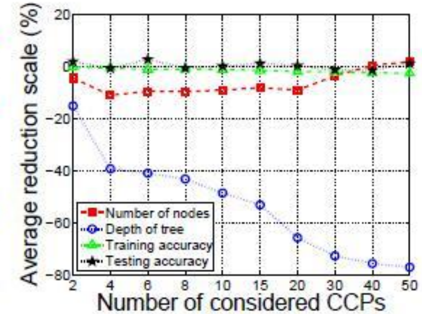
(a) Haberman



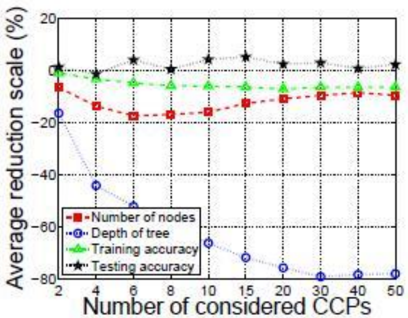
(b) Ionosphere



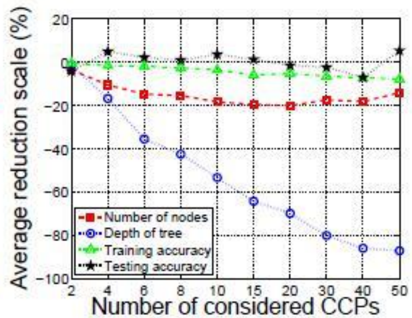
(c) Cancer



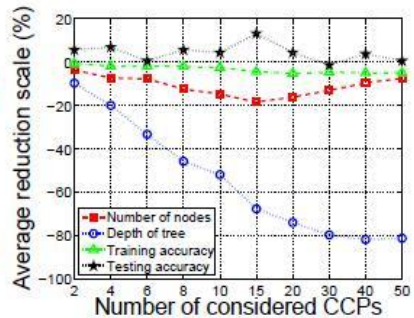
(d) Australian



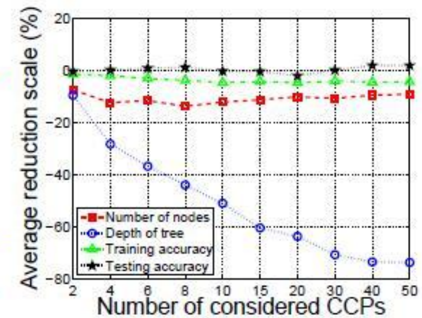
(e) German



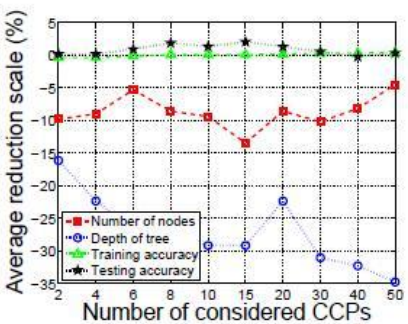
(f) Bupa



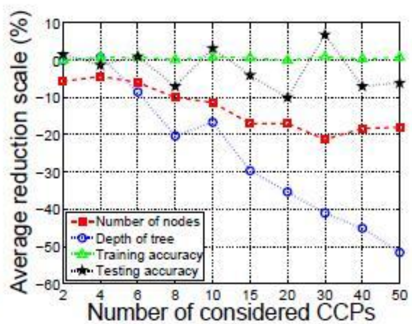
(g) Heart



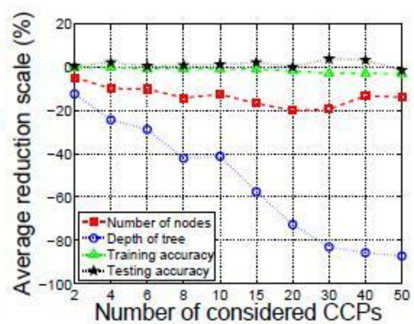
(h) Transfusion



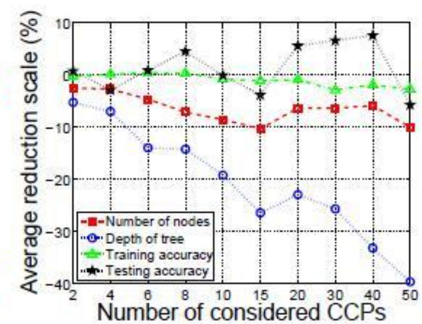
(i) Wdbc



(j) Wpbc



(k) Pima



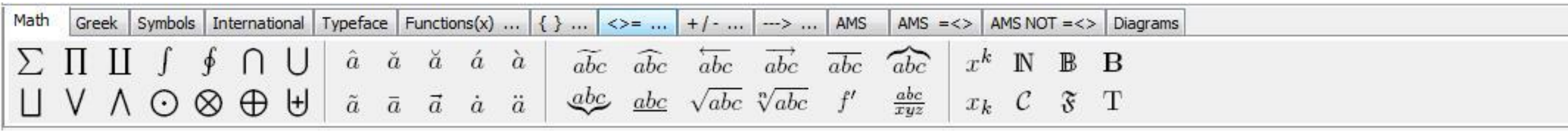
(l) Plrx

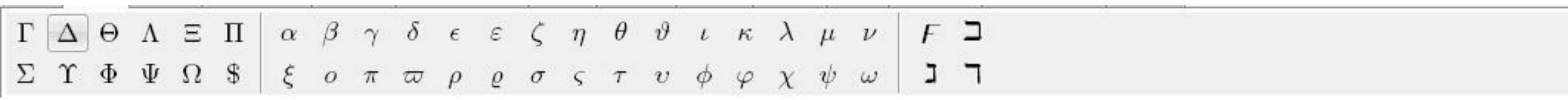
Math Environment

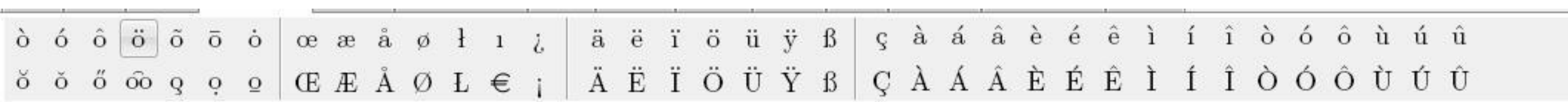
1、math

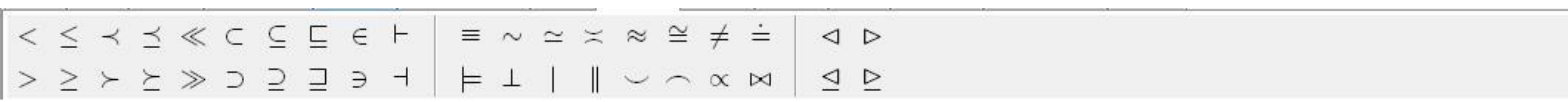
```
Noting that in section~\ref{subsec.trainingELM}, the training process of ELMs is written as $\beta^{*}=(\mathbf{H}^{\rm
T}\mathbf{H}+{\mu}\mathbf{I})^{-1}\mathbf{H}^{\rm T}\mathbf{T}$, where $\mu$ is a regularizing factor. This formula is
identical to $\beta^{*}={\mathbf{H}^{\dag}}\mathbf{T}$ if the regularizing factor takes value zero. It is proven
in~\cite{huang2012extreme} that the matrix $\mathbf{H}$ is of full rank with probability 1, and therefore, we can say that
the solution of normal equation $\mathbf{H}^{\rm T}\mathbf{H}\beta=\mathbf{H}^{\rm T}\mathbf{T}$ is available with
probability 1. In fact, the regularizing factor, which makes the solved weights as small as possible, has the effect to
become the matrix $\mathbf{H}$ full of rank.
```

Noting that in section II-A, the training process of ELMs is written as $\beta^{*}=(\mathbf{H}^{\rm T}\mathbf{H}+\mu\mathbf{I})^{-1}\mathbf{H}^{\rm T}\mathbf{T}$, where μ is a regularizing factor. This formula is identical to $\beta^{*}=\mathbf{H}^{\dag}\mathbf{T}$ if the regularizing factor takes value zero. It is proven in [24] that the matrix $\mathbf{H}$ is of full-rank with probability 1, and therefore, we can say that the solution of normal equation $\boxed{\mathbf{H}^{\rm T}\mathbf{H}\beta=\mathbf{H}^{\rm T}\mathbf{T}}$ is available with probability 1. In fact, the regularizing factor, which makes the solved weights as small as possible, has the effect to become the matrix $\mathbf{H}$ full of rank.


Math
Greek
Symbols
International
Typeface
Functions(x) ...
{ } ...
<>= ...
+ / - ...
---> ...
AMS
AMS =<>
AMS NOT =<>
Diagrams







2、displaymath
3、equation

```
\begin{equation}
\min\nolimits_{\beta}\left\{||\mathbf{T}-\mathbf{H}\beta||_2^2+\mu||\beta||_2^2\right\},\ \mu>0,
\end{equation}
```

$$\min\nolimits_{\beta}\left\{||\mathbf{T}-\mathbf{H}\beta||_2^2+\mu||\beta||_2^2\right\},\ \mu>0, \qquad (2)$$

4、Matrix and Array

```
\begin{equation}
\begin{array}{llll}
&\mathbf{H}({\mathbf{w}_1,\mathbf{w}_2,....,\mathbf{w}_{\widetilde{N}}},b_1,b_2,....,b_{\widetilde{N}},\mathbf{x}_1,\mathbf{x}_2,
\\
=&\left[
\begin{array}{cccc}
g(\mathbf{w}_1\cdot\mathbf{x}_1+b_1) &\cdots &g(\mathbf{w}_{\widetilde{N}}\cdot\mathbf{x}_1+b_1)\\
\vdots& \ddots& \vdots\\
g(\mathbf{w}_1\cdot\mathbf{x}_N+b_1) &\cdots &g(\mathbf{w}_{\widetilde{N}}\cdot\mathbf{x}_N+b_{\widetilde{N}})\\
\end{array}\right]_{N\times\widetilde{N}},
\end{array}
\end{equation}
```

$$
\begin{array}{ll}
&\mathbf{H}(\mathbf{w}_1,\mathbf{w}_2,....,\mathbf{w}_{\widetilde{N}},b_1,b_2,....,b_{\widetilde{N}},\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N)\\
=&\left[
\begin{array}{ccc}
g(\mathbf{w}_1\cdot\mathbf{x}_1+b_1) &\cdots &g(\mathbf{w}_{\widetilde{N}}\cdot\mathbf{x}_1+b_1)\\
\vdots& \ddots& \vdots\\
g(\mathbf{w}_1\cdot\mathbf{x}_N+b_1) &\cdots &g(\mathbf{w}_{\widetilde{N}}\cdot\mathbf{x}_N+b_{\widetilde{N}})
\end{array}\right]_{N\times\widetilde{N}},
\end{array}
\quad (3)
$$

5、界标排版命令

一、动态开界标命令：

\left<界标排版命令><数学公式>.....

二、动态闭界标命令：

.....<数学公式>\right<界标排版命令>

例子：

```
\[f(a,b,c)=(a+b)+\left\{\frac{a+b}{a-b}}\right\}
+\left[\frac{1}{a+b+c}}{\frac{1}{a+b-c}}\right]\]
```

输出结果：

$$f(a,b,c)=(a+b)+\left\{\frac{a+b}{a-b}\right\}+\left[\frac{\frac{1}{a+b+c}}{\frac{1}{a+b-c}}\right]$$

3、Command Set

1、\documentclass[options]{style}

Layout Options

\twoside (双面输出),
\twocolumn (左右两列排印),
\titlepage (标题内容单独占一页),
\proc (专用于排印符和ACM,IEEE要求的会议录文献),公式及其编号对齐方式选择
\leqno (在左边并左对齐),
\fleqn (左对齐)。

2、Font

- \rm
- \bf
- \it
- \sc
- \sl
- \tt
- \em

3、Font size, 10pt in default

- \tiny =5pt
- \scriptsize =7pt
- \footnotesize =8pt
- \small =9pt
- \normalsize =10pt
- \large =12pt
- \Large =14pt
- \LARGE =17pt
- \huge =20pt
- \Huge =25pt

4、Title commands

- **\title{}**
- **\author{}**
- **\date{ }**
- **\maketitle**
- **\thanks{ } – address, funding, emails, ……**

5、chapters/sections/paragraphs

- \chapter{…} -- for report and book
- \section{…}
- \subsection{…}
- \subsubsection{…}
- \subsubsubsection{…}
- \paragraph{…}
- \subparagraph{…},\sub…subparagraph{…}.
- \appendix

6、Page Commands

- **\pagestyle{options}**
  - **plain (空头注，页码在脚注区),**
  - **empty (空头注，空脚注)，**
  - **headings (头注由文献形式确定,article节号和页号作为头注)**
  - **myheadings (头注内容自定义,单面输出可用)**
- **\markright{右页眉内容}**
- **\markleft{左页眉内容}**
- **\markboth{左页头注内容}{右页头注内容}**
- **\thispagestyle{页眉排版方式}**
- **\topmargin size**
- **\headheight size**
- **\headsep size**
- **\footnote[num编号]{脚注内容}**
- **Etc……**

4、 Algorithm

```
\usepackage{algorithm}
\usepackage{algorithmic}
\usepackage[ruled,linesnumbered,lined,commentsnumbered]{algorithm2e}

\begin{algorithm}[position]
\caption{……}
\label{……}
    ……
    ……
    ……
\end{algorithm}
```

```
\begin{algorithm}[t]\small
\caption{Basic Framework for SVM-based MIAL}
\label{alg.MIALframework}
\KwIn{\\
    \quad Labeled set $\mathbb{L}=\{(\mathcal{B}_i,y_i)\}_{i=1}^{l}$;\\
    \quad Unlabeled pool $\mathbb{U}=\{\mathcal{B}_i\}_{i=l+1}^{l+u}$;\\
    \quad Parameters for training SVM.
}

\KwOut{\\
    \quad SVM solution $(\mathbf{w},b)$.
}

Train mi-SVM or MI-SVM on $\mathbb{L}$ to get SVM solution $(\mathbf{w},b)$;\\

\While{$\mathbb{U}$ is not empty}{

\uIf{stop criterion is met}{

\Return{$(\mathbf{w},b)$};\\

}\Else{

Calculate the informativeness of each $\mathcal{B}_i \in \mathbb{U}$, denoted as $\mathcal{I}(\mathcal{B}_i)$;\\

Select $\mathcal{B}^*=\argmax_{\mathcal{B}_i\in \mathbb{U} } \mathcal{I}(\mathcal{B}_i)$;\\

Query the label of $\mathcal{B}^*$, denoted by $y^*$;\\

Let $\mathbb{U}=\mathbb{U}\setminus \mathcal{B}^*$, and $\mathbb{L}=\mathbb{L}\cup (\mathcal{B}^*,y^*)$;\\

Update SVM solution $(\mathbf{w},b)$ based on new $\mathbb{L}$;\\

}}

\Return{$(\mathbf{w},b)$}.\\

\end{algorithm}
```

Algorithm 1: Basic Framework for SVM-based MIAL

```
Input:
    Labeled set 𝕃 = {(ℬ_i, y_i)}_{i=1}^{l};
    Unlabeled pool 𝕌 = {ℬ_i}_{i=l+1}^{l+u};
    Parameters for training SVM.
Output:
    SVM solution (w, b).
Train mi-SVM or MI-SVM on 𝕃 to get SVM solution (w, b);
while 𝕌 is not empty do
    if stop criterion is met then
        return (w, b);
    else
        Calculate the informativeness of each ℬ_i ∈ 𝕌, denoted as ℐ(ℬ_i);
        Select ℬ* = argmax_{ℬ_i ∈ 𝕌} ℐ(ℬ_i);
        Query the label of ℬ*, denoted by y*;
        Let 𝕌 = 𝕌 \ ℬ*, and 𝕃 = 𝕃 ∪ (ℬ*, y*);
        Update SVM solution (w, b) based on new 𝕃;
    end
end
return (w, b).
```

5、 References

1、 Create a .bib file

2、 Edit the .bib file

3、 Add the .bib file to the tex file

\bibliographystyle{IEEEtran}

\bibliography{reference}

4、 Using command “cite{}” to cite an article

```
@string{TKDE="Transactions on Knowledge and Data Engineering"}
@string{TEVC="Transactions on Evolutionary Computation"}
@string{TMM="Transactions on Multimedia"}
@string{TSMCB="Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics"}
@string{TITS="Transactions on Intelligent Transportation Systems"}
@string{TPDS="Transactions on Parallel and Distributed Systems"}
@string{TVCG="Transactions on Visualization and Computer Graphics"}
@string{IJUFKS="International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems"}
```

```
@article{hartigan1979algorithm,
  title={Algorithm {AS} 136: {A} k-means clustering algorithm},
  author={Hartigan, J. A. and Wong, Ma. A.},
  journal={Applied Statistics},
  pages={100--108},
  year={1979},
}

@book{nigrin1993neural,
  title={Neural networks for pattern recognition},
  author={Nigrin, A.},
  year={1993},
  publisher={The MIT press}
}

@article{wang2012study,
  title={A study on random weights between input and hidden layers in extreme learning machine},
  author={Wang, Ran and Kwong, Sam and Wang, Xizhao},
  journal={Soft Computing},
  volume={16},
  number={9},
  pages={1465--1475},
  year={2012},
  publisher={Springer}
}

@article{wang2013analysis,
  title={An analysis of {ELM} approximate error based on random weight matrix},
  author={Wang, Ran and Kwong, Sam and Wang, Debby Dan},
  journal={IJUFKS},
  volume={21},
  number={supp02},
  pages={1--12},
  year={2013},
  publisher={World Scientific}
}
```

REFERENCES

[1] S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan, "VAIT: A visual analytics system for metropolitan transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1586–1596, Dec. 2013.

[2] Y. Liu, C. Kang, S. Gao, Y. Xiao, and Y. Tian, "Understanding intra-urban trip patterns from taxi trajectory data," *J. Geographical Syst.*, vol. 14, no. 4, pp. 463–483, 2012.

[3] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, 2011, pp. 89–98.

[4] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. V. D. Wetering, "Visual traffic jam analysis based on trajectory data," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2159–2168, Dec. 2013.

[5] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "T-Drive: Enhancing driving directions with taxi drivers' intelligence," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 220–232, Jan. 2013.

[6] J. Yuan, et al., "T-Drive: Driving directions based on taxi trajectories," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 99–108.

[7] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger?" in *Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, 2011, pp. 109–118.

[8] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-Finder: A recommender system for finding passengers and vacant taxis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2390–2403, Oct. 2013.

[9] G. Qi, et al., "How long a passenger waits for a vacant taxi? Large-scale taxi trace mining for smart cities," in *Proc. IEEE Int. Conf. Green Comput. Commun.*, 2013, pp. 1029–1036.

[10] B. Li, et al., "Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2011, pp. 63–68.

[11] S. Ma, Y. Zheng, and O. Wolfson, "T-Share: A large-scale dynamic taxi ridesharing service," in *Proc. IEEE 25th Int. Conf. Data Eng.*, 2013, pp. 410–421.

[12] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "On predicting the taxi-passenger demand: A real-time approach," in *Progress in Artificial Intelligence*. Berlin, Germany: Springer, 2013, pp. 54–65.

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *2015 IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[15] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 29th Int. Conf. Mach. Learn.*, 2015, pp. 807–814.

[16] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Sci.*, vol. 313, no. 5, pp. 504–507, 2006.

[27] J. W. Powell, Y. Huang, F. Bastani, and M. Ji, "Towards reducing taxicab cruising time using spatio-temporal profitability maps," in *Proc Int. Symp. Spatial Temporal Databases*, 2011, pp. 242–260.

[28] D. Zhang and T. He, "pCruise: Reducing cruising miles for taxicab networks," in *Pro. 33rd IEEE Real-Time Syst. Symp.*, 2012, pp. 85–94.

[29] Y. Zhu, Y. Zheng, L. Zhang, D. Santani, X. Xie, and Q. Yang, "Inferring taxi status using GPS trajectories," MSR Tech. Rep. MSR-TR-2011-144, 2011, https://www.microsoft.com/en-us/research/publication/inferring-taxi-status-using-gps-trajectories/

[30] Y. Huang and J. W. Powell, "Detecting regions of disequilibrium in taxi services under uncertainty," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2012, pp. 139–148.

[31] Y. Li, J. Luo, C.-Y. Chow, K.-L. Chan, Y. Ding, and F. Zhang, "Growing the charging station network for electrical vehicles with trajectory data analytics," in *Proc. IEEE 25th Int. Conf. Data Eng.*, 2015, pp. 1376–1387.

[32] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1782–1795, Jul. 2015.

[33] D. Zhang, T. He, Y. Liu, and J. A. Stankovic, "CallCab: A unified recommendation system for carpooling and regular taxicab services," in *Proc. IEEE Int. Conf. Big Data*, 2013, pp. 439–447.

[34] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He, "coRide: Carpool service with a win-win fare model for large-scale taxicab networks," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst.*, 2013. Art. no. 9

[35] D. Zhang, T. He, S. Lin, S. Munir, and J. A. Stankovic, "Dmodel: Online taxicab demand model from big sensor data in a roving sensor network," in *Proc. IEEE Int. BigData Congr.*, 2014, pp. 152–159.

[36] H. Tang, M. Kerber, Q. Huang, and L. Guibas, "Locating lucrative passengers for taxicab drivers," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2013, pp. 504–507.

[37] S. Van Dongen , "Graph clustering by flow simulation," Ph.D. dissertation, University of Utrecht, Utrecht, Netherlands, 2000.

[38] S. Van Dongen, "Graph clustering via a discrete uncoupling process," *Siam J. Matrix Anal. Appl.*, vol. 30, no. 1, pp. 121–141, 2008.

[39] S. Haykin, *Neural Network: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice Hall, 1998.

[40] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[41] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[42] X. M. Chen, W. Q. Liu, J. H. Lai, Z. Li, and C. Lu, "Face recognition via local preserving average neighborhood margin maximization and extreme learning machine," *Soft. Comput.*, vol. 16, no. 9, pp. 1515–1523, 2012.

[43] Y. J. Li, Y. Li, J. H. Zhai, and S. Shiu, "RTS game strategy evaluation using extreme learning machine," *Soft. Comput.*, vol. 16, no. 9, pp. 1627–1637, 2012.

The End！