Demographic Attributes Prediction Using Extreme Learning Machine

Ying Liu, Tengqi Ye, Guoqi Liu, Cathal Gurrin and Bin Zhang

Abstract Demographic attributes prediction is fundamental and important in many applications in real world, such as: recommendation, personalized search and behavior targeting. Although a variety of subjects are involved with demographic attributes prediction, e.g. there are requirements to recognize and predict demography from psychology, but the traditional approach is dynamic modeling on specified field and distinctive datasets. However, dynamic modeling takes researchers a lot of time and energy, even if it is done, no one has an idea how good or how bad it is. To tackle the problems mentioned above, a framework is proposed in this chapter to predict using classifiers as core part, which consists of three main components: data processing, predicting using classifiers and prediction adjustments. The component of data processing performs to clean and format data. The first step is extracting relatively independent data from complicated original dataset. In the next step, the extracted data goes through different paths based on their types. And at the last step, all the data will be transformed into a demographic attributes matrix. To fulfill prediction, the demographic attributes matrix is taken as the input of classifiers, and the testing

Y. Liu (✉)
School of Software Engineering, Northeastern University, Shenyang 110819, China
e-mail: Liuy@swc.neu.edu.cn

T. Ye · C. Gurrin
School of Computing, Dublin City University, Dublin, Ireland
e-mail: yetengqi@gmail.com

C. Gurrin
e-mail: cathal@gmail.com

G. Liu
School of Computing, Northeastern University, Shenyang 110819, China
e-mail: liuguoqi@mail.neu.edu.cn

B. Zhang
College of Information Science and Engineering, Northeastern University,
Shenyang 110819, China
e-mail: zhangbin@mail.neu.edu.cn

F. Sun et al. (eds.), *Extreme Learning Machines 2013: Algorithms and Applications*, Adaptation, Learning, and Optimization 16, DOI: 10.1007/978-3-319-04741-6_11,

dataset comes from the same matrix as well. Classifiers in the experiments includes conventional state-of-the-art ones and Extreme Learning Machine, a new outstanding classifier. From the results of experiments based on two unique datasets, it is concluded ELM outperforms others. In the stage of prediction adjustments, two kinds of adjustments strategies are proposed corresponding to single target attributes and multiple target attributes separately, where single target attributes adjustments strategies include: adjusting the parameters of classifiers, adjusting the number of classes of target attributes and adjusting the public attributes. And multiple target attributes adjustment utilizes the outputs of first prediction as the inputs of second prediction to improve the accuracy of the first prediction. The framework proposed in this chapter consumes less time compared with traditional dynamic modeling methods, and there is no need to fully study the knowledge in various subjects for researchers using the framework because of the regular patterns. In addition, adjustment strategies have no restriction on the datasets; hence it will be useful universally. However, in some cases, dynamic modeling has the advantage of precision, resulting in better accuracy, but the results from the framework proposed in the chapter could provide as a comparison. In this work, a universal demographic attributes prediction framework is proposed to work on a variety of dataset with Extreme Learning Machine (ELM). The framework consists of three main components: First, processing raw data and extracting attribute features depending on different data types; Second, predicting desired attributes by classification; Third, improving the accuracy of classifiers through various adjustment strategies. Two experiments of different data types on real world prediction problems are conducted to demonstrate our framework can achieve better performance than other traditional state-of-the-art prediction methods with respect to accuracy. *abstract* environment.

Keywords Demographic attributes prediction · Extreme learning machine

1 Introduction

Demographic attributes prediction is to predict desired attributes information of human after gathering and analyzing all the attributes information of others. It is important and fundamental for many applications, such as recommendation, personalization, and behavior targeting [1]. Some but not many chapters and methods have been proposed to perform demographic attributes prediction [2, 3]. However, almost all methods require constructs diverse models depending on different datasets; while rest methods apply similar model on different curriculum. After performing discretization on raw data, the key of prediction is classification. Thus, as an excellent classifier, Extreme Learning Machine is taken as an important section in the model which could apply on diverse dataset and situations with little modification.

In the last few years, demographic attributes prediction attracted great attention from all over the world. In January 2009 Nokia Research Center Lausanne and its Swiss academic partners Idiap and EPFL started gathering demographic data through mobile phones and afterwards hold Mobile Data Challenge based on the

data [4]. There were three tasks in the Challenge and the third one was about demographic attributes prediction. Sanja Brdar [5] proposed k-nearest neighbors, radial basis function network and random forest as classification access. Kaixiang Mo [6] proposed Support Vector Machine to perform classification, which obtains better accuracy. NishKam Ravi [7] studied activity recognition from accelerometer data which accessed amazing prediction accuracy.

In previous work, almost all researches have analyzed their collected dataset and construct corresponding model which could not be applied to other datasets. The main contribution of this chapter is a model proposed which could be widely applied to variety of demographic attributes prediction with only little adjustment in the data processing stage. Raw data is divided into two groups: continuous variables and discrete variables. The two groups are different in data processing stage but similar in prediction stage. Concluded from the results, although the model is universal, it can produce fabulous accuracy sometimes.

The rest chapter is organized as follows: Sect. 2 briefly introduces ELM and SVM algorithms. Section 3 presents process on raw datasets. Section 4 demonstrates prediction and prediction adjustment. Section 5 compares experimental results of ELM with that of SVM. Conclusions and future work are in Sect. 6.

2 Brief Introduction of ELM

Countless classification theories and methods have been proposed to solve the problem and a lot of them are successful. One of the goals of this work is to compare the performance of Extreme Learning Machine with other classifiers [8].

Traditional learning speed of feedforward neural networks is in general far slower than required and it has been a major bottleneck in their applications for past decades [9]. Mainly because of two reasons: (*a*) Usual gradient-based learning algorithms are slow; and (*b*) all the parameters of the networks are tuned iteratively by using such algorithms. To tackle the issues and improve learning speed of feedforward neural networks [10], Huang et al proposed Extreme Learning Machine from single-hidden layer feedforward neural networks (SLFNs) which can randomly select the initial values for the hidden layer bias and input weights at the condition that the activation functions are infinitely differentiable [11].

For N arbitrary distinct samples $(\mathbf{x_i}, \mathbf{t_i})$, where $\mathbf{x_i} = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in \mathbf{R}^n$ are data vectors and $\mathbf{t_i} = [t_{i1}, t_{i2}, \cdots, t_{in}]^T \in \mathbf{R}^m$ are the target classes, standard SLFNs with $\tilde{N}$ hidden nodes and activation function $g(x)$ can be mathematically modeled as

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(\mathbf{x_j}) = \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w_i} \cdot \mathbf{x_j} + b_i) = \mathbf{o_j}, \quad j = 1, \cdots N \tag{1}$$

where $\mathbf{w_i} = [w_{i1}, w_{i2}, \cdots, w_{in}]^T$ is the weight vector connecting the ith hidden node and the input nodes, $\beta_\mathbf{i} = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the weight vector connecting the ith hidden node and the output nodes, and b_i is the threshold of the ith hidden node. In addition [12], $\mathbf{w_i} \cdot \mathbf{x_j}$ represents the inner product of $\mathbf{w_i}$ and $\mathbf{x_j}$. And Extreme Learning Machine with $\tilde{N}$ hidden nodes with activation function $g(x)$ will approximate these N samples with zero error means that $\sum_{i=1}^{\tilde{N}} \left|\left|\mathbf{o_j} - \mathbf{t_j}\right|\right| = 0$, i.e., there exist β_i, $\mathbf{w_i}$ and b_i such that

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w_i} \cdot \mathbf{x_j} + b_i) = \mathbf{t_j}, \quad j = 1, \cdots N \tag{2}$$

The above N equations can be denoted in the form of matrix as

$$\mathbf{H}\beta = \mathbf{T} \tag{3}$$

where

$$\mathbf{H}(\mathbf{w_1}, \cdots \mathbf{w}_{\tilde{\mathbf{N}}}, \mathbf{b_1}, \cdots \mathbf{b}_{\tilde{\mathbf{N}}}, \mathbf{x_1}, \cdots \mathbf{x_N}) = \tag{4}$$
$$\begin{bmatrix} g(\mathbf{w_1} \cdot \mathbf{w_2} + b_1) & \cdots & g(\mathbf{w}_{\tilde{\mathbf{N}}} \cdot \mathbf{x_1} + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w_1} \cdot \mathbf{w_N} + b_1) & \cdots & g(\mathbf{w}_{\tilde{\mathbf{N}}} \cdot \mathbf{x_N} + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad and \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_{\tilde{N}}^T \end{bmatrix}_{N \times m} \tag{5}$$

and $\mathbf{H}$ is the hidden layer output matrix of the neural network; the ith column of $\mathbf{H}$ is the ith hidden node output with respect to inputs $\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}$.

3 Data Process

As different raw datasets are usually in different patterns and more or less contain some errors or exceptions, processing the raw datasets before taking them as prediction input is necessary and important. In addition, processing before predicting will improve the accuracy of result and efficiency of prediction. Raw data can be divided into two types: continuous variables and discrete variables. In fact, there is no continuous value in continuous variables. Continuous variables contain observations with no obvious intervals. Data process is shown as Fig. 1.

In our model, demographic attributes matrix provides as direct input of classifiers for prediction. As original datasets are complicated and differs from each other, our

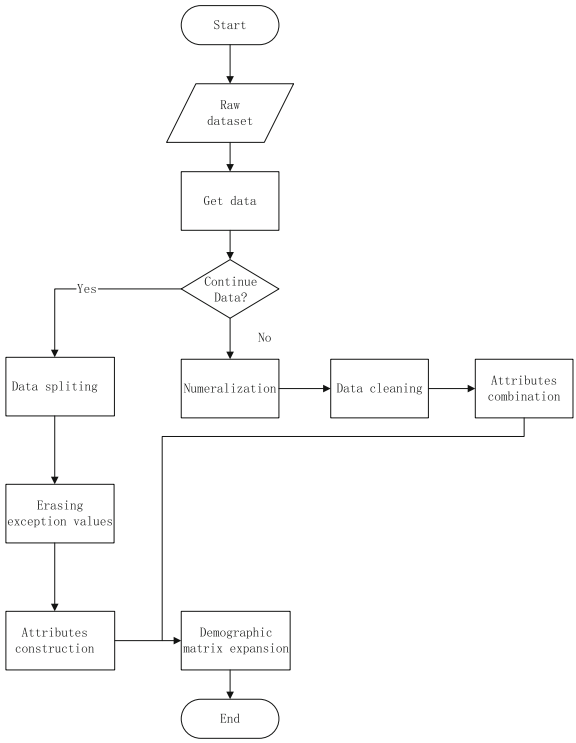

Fig. 1 Process on different types of data

model will transform the original datasets to demographic attributes matrices in the data processing stage. Data cleaning and attributes constructions are two main goals in data processing, as shown in Fig. 1. The first step is to extract logical consistent data from the original dataset. Continuous data and discrete data will go through different processing, which will be detailed explained later. After all the processing, at the end, demographic attributes matrix will be constructed for prediction.

Logical consistent data refers to data of same meaning: data from an extraction could be of same attributes from all people or of same attributes from one person, e.g. phone call records of same person or all sex choices from investigation. After an extraction, the following process differs depending on different data types, continuous data or discrete data. Normally speaking, discrete data comes from data of same attributes from group of people.

3.1 Demographic Attributes Matrix Representation

A demographic attributes matrix is a matrix represents information of a group of individuals, where each line vector is the attributes set of corresponding individual and each column vector represents corresponding attribute of every individuals.

Demographic attributes matrix is denoted by D and D_i^j denotes the element of ith line and jth column, representing the value of ith individual on jth attribute. The individual in the matrix could be other than person. If attributes of people are to be predicted, the individual in the matrix refers to person; while attributes between people, like closeness level, are to be predicted, the individual refers to pair of people.

To construct a demographic attributes matrix, three steps should be done: individuals identifying, attributes selecting and determining every elements in it. Individuals identification is easy to operate based on what to be predicted, while the other two steps are more difficult and more complex.

3.2 Continuous Variables Process

Data collected by sensors, such as accelerators, usually consists of continuous variables. As classifier is not able to work directly on continuous variables, several statistical characteristics are selected to represent the raw data. Resulted from the problem of sensors and rare situations, few data are extreme values, mostly extremely large. Although the number is few, these exceptions would have great impact on statistical characteristics presenting the original data. Thus, exceptions would be erased before feature construction. In addition, as enough number of testing cases is necessary for classifier to learn, the original data should be split wisely. In conclusion, continuous raw data will go through data splitting, data cleaning and feature construction in turn.

Hampel identifier are widely used to detect extreme values which defines Z':

$$Z' = \frac{|X_i - Median|}{\left(\frac{MAD}{.6745}\right)} \tag{6}$$

where X_i is each observation in the dataset, Median is the median of the X_i, and MAD is median absolute deviation (MAD) between X_i and Median. In the experiments, Hampel identifier is improved to perform much better according to the characteristic of the data. Pseudocode is shown below to utilize the improved Hampel identifier to detect exception values:

EXCEPTION_DETECTION($[x_1, x_2, \cdots, x_n]$)

1 $Exception\_detection \leftarrow []$ ▷ exception set is empty at the beginning
2 $med \leftarrow Median([x_1, x_2, \cdots, x_n])$ ▷ find median of input
3 **for** x_i in $[x_1, x_2, \cdots, x_n]$
4 **do** $x_i' = |x_i - median|$
5 $MAD \leftarrow Mean([x_1', x_2', ..., x_n'])$ ▷ figure out average distance
6 **for** x_i' in $[x_1', x_2', ..., x_n']$
7 **if** $\frac{|x_i - median|}{\frac{MAD}{0.6745}} > threshold$ ▷ compare the result with threshold
8 **then** $exception\_set \leftarrow [exception\_set, x_i]$

The MAD in the pseudocode prefers to mean absolute deviation instead of median absolute deviation in original Hampel identifier.

There are several statistical characteristics could represent original data, such as average number, variance and median, etc. Sometimes, rate and percentage could be more useful.

3.3 Discrete Variables Process

Discrete variables normally come from surveys or counters instead of sensors. Not like continuous variables, there is no extreme value in discrete variables, but instead null values may appear. Not all null values will affect the result, but those may affect will be deleted. Since some questions in survey have connections, related attributes will be erased as well. As answers from investigation in sentences could not be directly used in classifier, numeralization will turn the sentences into numbers before classification. At classification stage, attributes should be independent of each other; thus, related attributes should be combined into one. For a question, if all answers are of strings, they are numbered in continual positive integers. If all answers are of numbers, they remain. If in continuous data, data cleaning focuses on cleaning extreme values, however, null values are focused in discrete data.In continuous data, data cleaning focus on cleaning extreme values, however, null values are focus in discrete data.

Assume there are n questions, Q^i is the ith question. If the number of answers to Q^i is limited and values of answers are discrete, all the choices of answers are $< C_i^1, C_i^2, C_i^k >$, where k is the number of choices. If the choice of Q^i will influence the scope of choices of Q^j, Q^j depends on Q^i. If Q^c depends on Q^b and Q^b depends on Q^a, then Q^c depends on Q^a. Situations of circular dependency rarely happen, so they are not in the consideration, like Q^i depends on Q^j and Q^j depends on Q^i. The dependency relationship can be one to many, many to one or many to many. A dependency cluster contains elements where each two have dependency relationship and no one has relationship with elements on the outside.

Based on the assumptions above, it is feasible to combine all the answers of a dependency cluster to only one attribute. Because the number of questions is limited and there is no circular dependency, there will be elements which depend on no one and elements which no one depends on them. Combination algorithms are showed below:

COMBINATION($Q^1, Q^2, \cdots, Q^n$)

```
find elements which no one depends on them, D
for each Q^j in U
    do n_j ← 0
       for each C_i^k in answer of Q^i ▷ find all choices
           do for each C_j^k in answer of Q^j
                  do if it is possible C_i^k and C_j^k appears at the same time
                         then n_j ← n_j + 1
       < C_j^1, C_j^2, ··· , C_j^n > ←< 1, 2, ··· , n_j > ▷ rearrange choices
D ← D − {Q^i}
if Q^j depends on other elements
    then D ← D + {Q^i}
```

4 Prediction Using Different Classifiers and Prediction Adjustment

After previous work on raw data, data at this stage is in the form of matrix, each row of which is an observation and each column is an attribute. The whole data will go into two dataset: training dataset and testing dataset.

It is possible that our universal model could produce results with accuracy much lower than expected. There are three possibilities which could cause the problem: (*a*) the attributes are unpredictable; (*b*) classification is not suitable for predicting the attributes; (*c*) processing on the data is not suitable not enough or the original dataset is fake. Although in the first two conditions, there is a great chance traditional dynamic modeling on specified dataset outperforms our model, it also could provide as a comparison to indicate how well the traditional model is. In the last condition, some adjustments could be done to improve the accuracy of classifiers.

4.1 Demographic Prediction Analysis

The columns of a demographic attributes matrix are consist of various attributes, some are always directly accessible and the rest are not always known. The directly accessible attributes attributes in demographic attributes matrix D, are denoted by $D \cdot \boldsymbol{a}$. For the rest attributes in the matrix, sometimes the values of them are unknown, thus prediction is used to figure out the values of those target attributes. Target attributes in a demographic attributes matrix are denoted by $D \cdot \boldsymbol{t}$. If all values of target attributes are unknown, the corresponding demographic attributes matrix is predicting matrix, denoted by P. In the experiments, classifiers are used to predict $P \cdot t$. In fact, the results of prediction will never be known exactly, but accuracy is an important factor for evaluating the classifiers.

It is feasible to obtain accuracy rate by testing classifiers using known data. The classifiers provide as a mapping $y = f(x, c)$, where c is parameters unrelated with dataset. c can be worked out using $T \cdot \boldsymbol{t} = f(T \cdot \boldsymbol{a}, c)$, since T is entirely known. Thus $P \cdot \boldsymbol{t}$ is predicted through $P \cdot \boldsymbol{t} = f(P \cdot \boldsymbol{a}, c)$. To get the accuracy of the a classifier, after c is figured out, $V \cdot \boldsymbol{t}'$ results from $V \cdot \boldsymbol{t}' = f(V \cdot \boldsymbol{a}, c)$. And the the accuracy of the mapping is the accuracy between $V \cdot \boldsymbol{t}'$ and $V \cdot \boldsymbol{t}$.

Based on the number of labels, prediction can be divided into two types: predicting one attribute and predicting multiple attribute. And predicting multiple attributes is implemented by predicting one attribute separately multiple times.

4.2 Prediction with Various Classifiers

Concerning the number of target attributes, there are 2 types of prediction methods: single target attribute prediction and multiple target attributes prediction. Pseudocode of single attribute prediction is shown below:

Input: T, $P.\boldsymbol{a}$
Output: $P.\boldsymbol{t}$

SINGLE_TARGET_PREDICT(T, P)

1 $classifier\_train(T.\boldsymbol{t}, T.\boldsymbol{a})$ ▷ use training set train classifier
2 $P.t_i \leftarrow single\_target\_predict(P.\boldsymbol{a})$

As almost all classifiers are not able to predict multiple attributes directly, multiple attributes prediction is performing single target attribute prediction multiple times.

Decision tree [13], Naive Bayes algorithm and SVM [14] are used as other classifiers to compare with ELM in performance. Multiple target attributes prediction pseudocode is shown below:

Input: T, $P.\boldsymbol{a}$
Output: $P.\boldsymbol{t}$

MULTIPLE_TARGET_PREDICT(INPUT)

1 **for** $P.t_i$ in $P.\boldsymbol{t}$
2 **do** $P.t_i \leftarrow single\_target\_predict(T, P.\boldsymbol{a})$ ▷ utilize $single\_target\_predict$
3 $P.\boldsymbol{t} \leftarrow [P.\boldsymbol{t}, P.t_i]$

4.3 Prediction Adjustment Strategies

There are 2 types of prediction methods, so there are 2 types of prediction adjustment strategies: single target attribute prediction adjustment strategies and multiple target attributes prediction adjustment strategy. Because multiple target attributes prediction is based on single target attribute prediction, single target attribute prediction adjustment strategies could also influence multiple target attributes prediction.

Single target attribute prediction adjustments strategy

1. Adjusting the parameters of classifiers is able to improve the accuracy of prediction. Some classifiers require parameters unrelated with dataset to predict, like ELM. For them, trying different parameters leads to different accuracy and highest accuracy will be chosen.
2. Adjusting the attributes is able to improve the accuracy of prediction. Normally speaking, more attributes provide more information, which improve the accuracy of prediction. And sometimes, key attributes contribute most to accuracy. The process is in Fig. 2. In the process, for every loop, there is one more attribute to be added. If the accuracy decreases, the attribute will be erased. As a result, all chosen attributes perform positive effect to the final prediction.
3. Reducing the number of target attribute levels leads to improve accuracy of prediction. Normally the number of target attribute levels is smaller, the accuracy is higher. Pseudocode of target attribute levels adjusting is shown below, where k is the final number of levels and $[C_1, C_2, ..., C_n]$ is original class indexes:

Input: $[C_1, C_2, \cdots, C_n], k$
Output: $[C_1, C_2, \cdots, C_n]$

CLASSIFICATION(INPUT)

```
1  range ← n/k
2  remainder ← n%k
3  for i in [1, ··· , (1 + range) ∗ remainder] ▷ add remainder
4        do C_i ← (i − 1)/(range + 1)
5  for i in [(1 + range) ∗ remainder + 1, ··· , n]
6        do C_i ← (i − (1 + range) ∗ remainder)/range + remainder
```

As the original number of target attributes levels may not be divided exactly, classes will expand 1 scope.

Multiple target attributes prediction adjustments strategy Normally, the accuracy of classifiers increases when the number of attributes increases. Thus, it is possible to improve the accuracy of multiple target attributes prediction by taking predicted target attributes as known attributes to predict other target attributes. Corresponding pseudocode is shown below:

Input: T, $P.\boldsymbol{a}$
Output: $P.\boldsymbol{t}$

MULTI_TARGET_IMPROVE(INPUT)

```
1  P.t ← multi_target_predict(Input)▷ utilizemulti_target_predict
2  for t_i in P.t
3        do if classifier_vadiate(T, [T.a, T.t_i], i) > classifier_vadiate(T, T.a, i)
4              then t_i' ← single_target_predict([Input, t_1, ··· , t_i − 1, t_i + 1, ··· , t_m])
5                   t_i ← t_i'
```

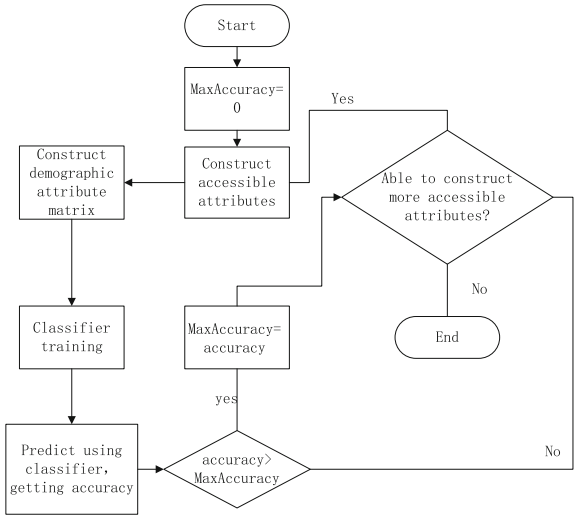

Fig. 2 Adjusting process of attributes

5 Experimental Results and Analysis

In this section, detailed experimental results of every components of model will be shown based on two datasets and performances of four different classifiers are compared for evaluation. There are three components in the model: data processing, classifying and adjusting classification, where performance comparisons between classifiers takes place in the classifying step. The four classifiers are decision tree, Naive Bayes algorithm, Support Vector Machine and Extreme Learning Machine.

In the experiments, all algorithms are implemented in Visual Studio 2010 and MATLAB R2012b. Experiments are run on a PC with Intel Corei5 2400, 3.10 GHz CPU, 3GB RAM and Windows XP operating system.

5.1 Raw Datasets Description

Two separate datasets are used to evaluate our model. The first dataset is full of accelerometer records and activity labels and the second dataset contains data from surveys and software applications in mobile phones.

The goal for first dataset is to predict the state of motions based on accelerometer records, which contains 7 labels of activity states: standing, walking, running, climbing up stairs, climbing down stairs, transporting and resting. Besides the labels, the original datasets mainly contains accelerations in three orthogonal directions and other support information, including timestamps, corresponding action labels and base station coordinates of recording. The 20.6 MB original dataset was split into matrix of 1436 observations [15].

Table 1 Files introduction from the second dataset

File name	Content	Collecting tool	Total line
closeness_adj.csv	Closeness value between each two informant	Survey	56
couples.csv	Marriage status and family information	Survey	56
BluetoothProximity.csv	Records of bluetooth contacts	Sensor	469924
SMS.csv	Records of SMS	Sensor	5301
VoiceCall.csv	Records of phone calls	Sensor	103468

There are two goals for second datasets: one is predicting closeness levels between arbitrary two people using data gathered through survey and data collected by software in cell phones; the other is predicting closeness levels and is couple or not between each pair using same data. It is assumed that is couple or not is known in first goal while unknown in second goal.

Instead of first dataset, data is collected by both surveys and sensors. The dataset has information of marriage status, closeness feeling, voice call records, message records and Bluetooth records. Marriage status includes which two is couple, sex of each one and how many children the one has. Closeness feeling indicates the closeness index with scope from 0 to 10. Voice call records shows when someone calls the other and if the other missed or not. Message records are similar to voice call but messages would surely arrive. Bluetooth records display when which two people get close enough. After decomposition, the 287 MB dataset consists of 5 files, detailed explanation is revealed in Table 1. Software applications on mobile phones are considered as sensor for convenience.

closeness_adj.csv consists of closeness levels between each two informants and there are 11 levels from 0 to 10, higher level indicating more close, where 0 denoting not known at all. couples.csv has information of which two informants is couple, how many children and sex. The software applications are more like counters rather than sensors: a Bluetooth contact, SMS receiving or sending and a phone call will trigger a recording.

5.2 Experimental Results of Data Processing

Exception detecting operations on first dataset is shown in Fig. 3. In the figure, red inverted triangle, blue circle and green point denote accelerations of three different orthogonal directions. The reason to utilize different shapes of different color is to distinguish each other. The black line in the figure is around 10^{11} of its original value to be distinguished from points below. And points above the black line are exception values.

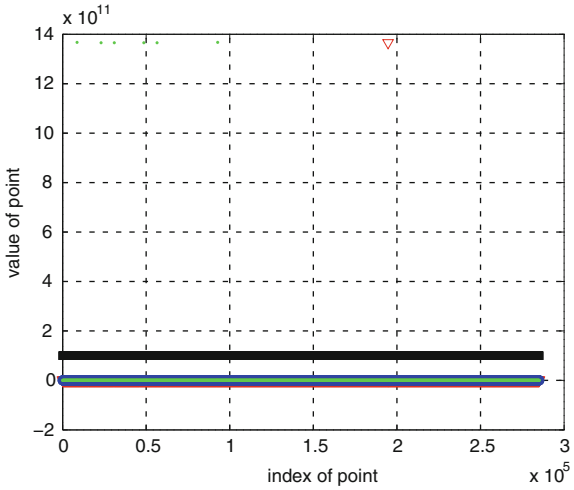

Fig. 3 Data processing of exceptions detection

Table 2 Performance of various classifiers using first dataset

Datasets	Training data	Training time	Testing data	Testing time	Accuracy
Decision tree	1436	0.0106	1436	0.0019	0.9582
Naive bayes	1436	0.1068	1436	1.5136	0.9582
SVM	1436	0.1829	1436	0.0248	0.9616
ELM	1436	0.3659	1436	0.1019	0.9999

5.3 Comparison Between Performances of Various Classifiers

In the section of classification, decision tree, Naive Bayes algorithm and Support Vector Machine provide as comparisons for Extreme Learning Machine.

1. Mean, standard deviation and median are selected as attributes from first dataset to predict motion states. There are 7 levels in target attribute, and RBF is chosen as activation function with parameter of 1, the result is shown in Table 2.
   Decision tree costs less training time and testing time than other algorithms, but with relatively low accuracy. Training time of Naive Bayes is about the same as that of others, but corresponding testing time is around hundred times of that of others. Accuracy of SVM is higher than that of decision tree and Naive Bayes, but consumes more training time. Accuracy of ELM is highest among 4 classifiers, but requires more time than SVM on both training time and testing time.
2. Is-couple, the number of children, ratio of Bluetooth contacts, ratio of SMS sending and receiving, and ratio of phone calls are selected as attributes from second dataset to predict closeness levels. There are 11 levels in target attribute, from 0 to 10, and RBF is chosen as activation function with parameter of 1, the result is shown in Table 3.

Table 3 Performance of various classifiers using second dataset to predict 11 target attribute levels

Datasets	Training data	Training time	Testing data	Testing time	Accuracy
Decision tree	3080	0.0115	3080	0.0185	0.8373
Naive bayes	3080	0.1257	3080	4.1247	0.8737
SVM	3080	0.0747	3080	0.0583	0.8386
ELM	3080	0.7819	3080	0.3874	0.8390

Table 4 Performance of various classifiers using second dataset to predict 3 target attribute levels

Datasets	Training data	Training time	Testing data	Testing time	Accuracy
Decision tree	3080	0.1136	3080	0.0012	0.8929
Naive bayes	3080	0.0482	3080	3.1450	0.8929
SVM	3080	0.4727	3080	0.0583	0.8996
ELM	3080	0.4988	3080	0.0731	0.9006

Decision tree costs less training time and testing time than other algorithms, but with lowest accuracy among 4 classifiers. Training time of Naive Bayes is about the same as that of others, but corresponding testing time is around hundred times of that of others, and it has the highest accuracy among 4 classifiers. Accuracy of ELM is second highest among 4 classifiers, but requires relatively more time than others.

3. Is-couple, the number of children, ratio of Bluetooth contacts, ratio of SMS sending and receiving, and ratio of phone calls are selected as attributes from second dataset to predict closeness levels. There are 3 levels in target attribute, from 0 to 2, which are classified from the original 11 levels. Linear kernel is chosen as activation function with parameter of 1, the result is shown in Table 4.
   Decision tree still costs less training time and testing time than other algorithms, but with relatively low accuracy. Training time of Naive Bayes is about the same as that of others, but corresponding testing time is around hundred times of that of others. Although ELM consumes more time on training and testing, its accuracy is highest among 4 classifiers, and only its accuracy is beyond 0.9.

5.4 Prediction Accuracy Results of Various Classifiers

There are single target attribute prediction adjustments strategy and multiple target attributes prediction adjustments strategy depending whether there is one target attribute or multiple target attributes to predict.

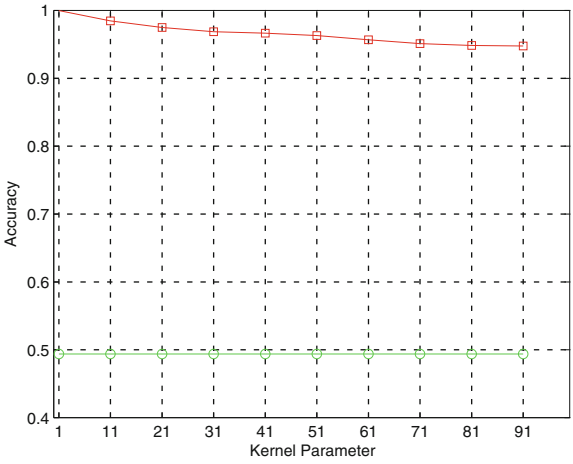

Fig. 4 Influence on accuracy with different parameters for ELM using first dataset (*Red bar*: RBF kernel, *green line*: linear kernel)

Single target attribute prediction adjustments strategy

1. Adjusting the parameters of classifiers is able to improve the accuracy of prediction. In this section, adjustment is performed on ELM using both datasets. Mean, standard deviation and median are selected as attributes from first dataset to predict motion states and there are 7 levels in target attribute. Under those conditions, linear kernel and RBF kernel is chosen separately as activation function with several corresponding parameters, as shown in Fig. 4.
   Using first dataset, RBF kernel performs much better than linear kernel and its accuracy gets higher with lower kernel parameter. However, the accuracy of linear kernel stays the same with different kernel parameters and its accuracy is much lower than that of RBF kernel. The highest accuracy from RBF kernel is close to 1, with kernel parameter 1.
   Is-couple, the number of children, ratio of Bluetooth contacts, ratio of SMS sending and receiving, and ratio of phone calls are selected as attributes from second dataset to predict closeness levels and there are 11 levels in target attribute, from 0 to 10. Under those conditions, linear kernel and RBF kernel is chosen separately as activation function with several corresponding parameters, as shown in Fig. 5.
   Using second dataset, accuracy of RBF kernel increases as corresponding kernel parameter decreases. However, the accuracy of linear kernel stays the same with different kernel parameters. The highest accuracy from RBF kernel is higher than that of linear kernel, but most values of RBF kernel is lower than linear kernel.
   Is-couple, the number of children, ratio of Bluetooth contacts, ratio of SMS sending and receiving, and ratio of phone calls are selected as attributes from second dataset to predict closeness levels and there are 3 levels in target attribute, from 0 to 2. The 3 levels comes from original 11 levels: 0 to 3 denotes almost unknown (0 now), 4–7 denotes known (1 now), 8–10 denotes known very well (2 now). Under those conditions, linear kernel and RBF kernel is chosen separately as

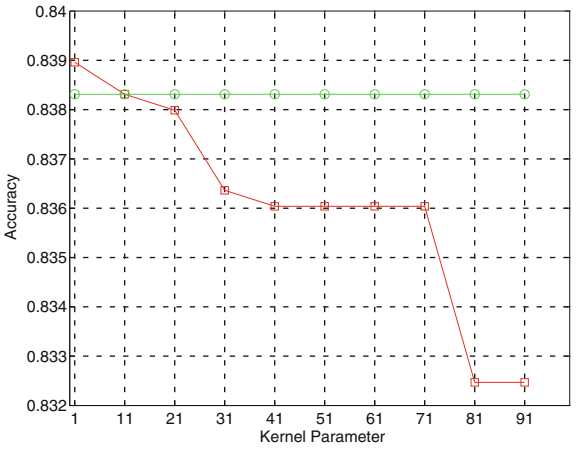

Fig. 5 Influence on accuracy with different parameters for ELM using second dataset with 11 target levels (*Red bar*: RBF kernel, *green line*: linear kernel)

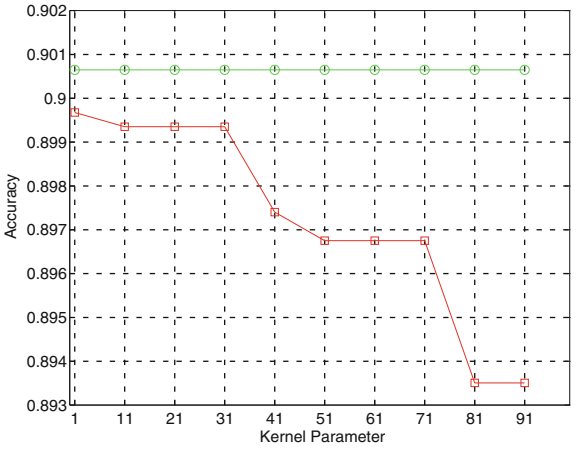

Fig. 6 Influence on accuracy with different parameters for ELM using second dataset with 3 target levels (*Red bar*: RBF kernel, *green line*: linear kernel)

activation function with several corresponding parameters, as shown in Fig. 6. Using second dataset, RBF kernel performs worse than linear kernel and its accuracy gets higher with lower kernel parameter. However, the accuracy of linear kernel stays the same with different kernel parameters and its accuracy is much higher than that of RBF kernel.

2. Adjusting the attributes is able to improve the accuracy of prediction. In the section, adjustment is performed on ELM using only first datasets.
   There are 7 levels in target attribute from first dataset to predict motion states. Different attributes combination are selected from mean, standard deviation and median. Linear kernel and RBF kernel is chosen separately as activation function with several corresponding parameters, as shown in Fig. 7. Every result shown in the figure is the mean from corresponding results of several different kernel parameters, because the highest accuracies are almost the same.

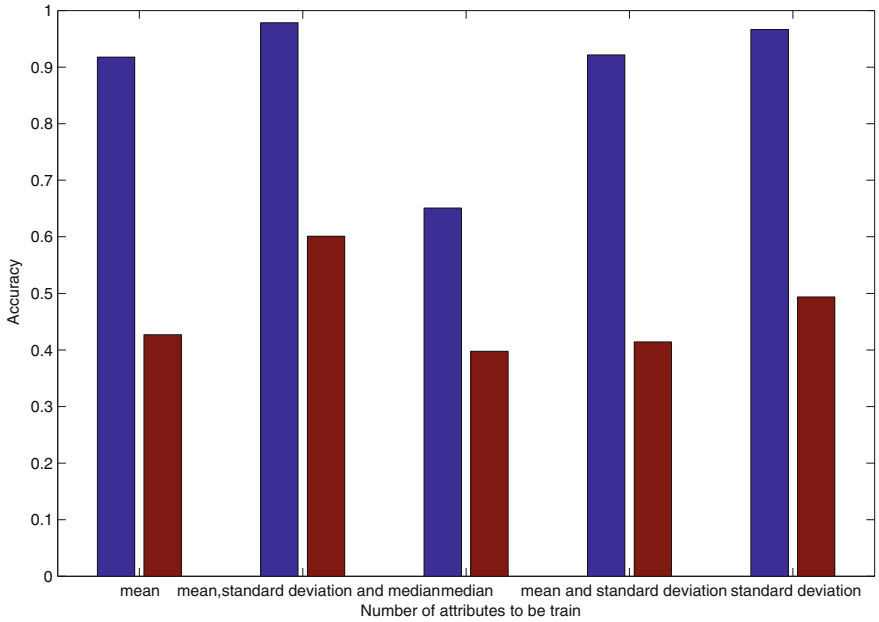

Fig. 7 Influence on accuracy with different parameters for ELM using first dataset (*Red bar*: RBF kernel, *green bar*: linear kernel)

Chosen attributes combinations are: mean, standard deviation, median, (mean and standard deviation), and (mean, standard deviation and median). It can be concluded from the figure that the last combination gets highest accuracy and normally, accuracy increase as the number of attributes increases.

3. Reducing the number of target attribute levels leading to advancing accuracy of prediction. In the section, adjustment is performed on ELM using only second datasets.
   The reason to use only second database is the target attribute levels in first dataset is fixed. Figure 8 shows the specified operation on changing the levels of target attribute from second dataset. There are original 11 levels and they are divided into 2, 3 and 5 levels as even as possible separately.
   Is-couple, the number of children, ratio of Bluetooth contacts, ratio of SMS sending and receiving, and ratio of phone calls are selected as attributes from second dataset to predict closeness levels. Every result shown in the figure is the highest from corresponding results of several different kernel parameters. It is concluded from Fig. 9 that accuracy of prediction increases as the number of target attribute levels decreases. However, in fact, 3 levels of target attribute is better than others, because index of emotions can not be too precise and 2 levels are too ambiguous.

Multiple target attributes prediction adjustments strategy As mentioned above, the accuracy of classifiers increases when the number of attributes increases. Thus, it

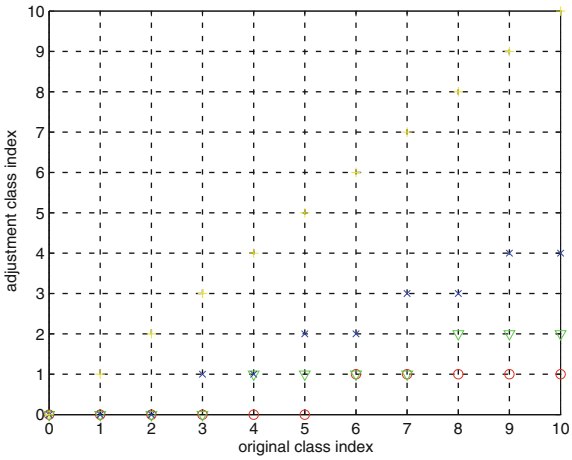

Fig. 8 Division on original 11 target attribute levels (*Red circle*: 2 levels; *green inverted triangle*: 3 levels; *blue cross*: 5 levels; *yellow* add sign: original 11 levels)

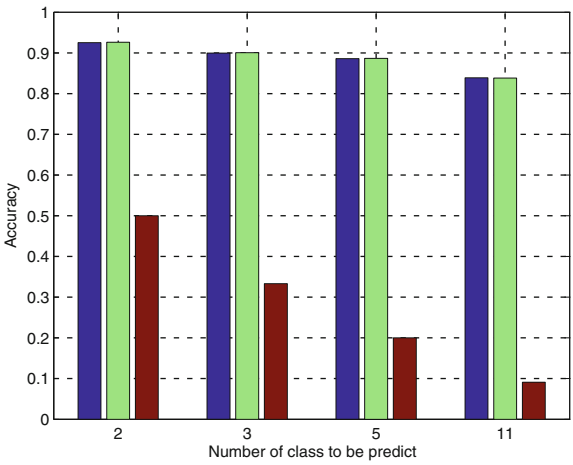

Fig. 9 Influence on accuracy with different number of target attribute levels for ELM using second dataset (*Red bar*: random algorithm; *green bar*: RBF kernel; *blue bar*: linear kernel)

is possible to improve the accuracy of multiple target attributes prediction by taking predicted target attributes as known attributes to predict other target attributes. The chosen predicted target attributes should be well predicted, corresponding to high prediction accuracy.

Only second dataset is used: assuming closeness levels and is-couple relationship are both target attributes. There are two choices of is-couple relationship, yes or no. And original 11 levels are chosen for closeness levels.

Firstly, two independent predictions are performed for the two target attributes separately using other known attributes: the number of children, ratio of Bluetooth contacts, ratio of SMS sending and receiving, and ratio of phone calls. Highest results are selected using ELM from different parameters in Table 5.

Table 5 Direct independent multiple target attributes prediction using ELM from different parameters

Target attribute	Kernel function	Kernel parameter	Accuracy
is-couple	RBF	1	0.9974
is-couple	RBF	11	0.9968
is-couple	linear	1	0.9655
is-couple	linear	11	0.9655
closeness	RBF	1	0.8377
closeness	RBF	11	0.8364
closeness	linear	1	0.8367
closeness	linear	11	0.8367

Table 6 Second time independent multiple target attributes prediction using data derived from Table 5

Target attribute	Kernel function	Kernel parameter	Accuracy
is-couple	RBF	1	0.9974
is-couple	RBF	11	0.9971
is-couple	linear	1	0.9961
is-couple	linear	11	0.9961
closeness	RBF	1	0.8377
closeness	RBF	11	0.8370
closeness	linear	1	0.8373
closeness	linear	11	0.8373

Table 7 Third time multiple target attributes prediction using complete attributes

Target attribute	Kernel function	Kernel parameter	Accuracy
is-couple	RBF	1	0.9974
is-couple	RBF	11	0.9968
is-couple	linear	1	0.9969
is-couple	linear	11	0.9969
closeness	RBF	1	0.8390
closeness	RBF	11	0.8383
closeness	linear	1	0.8383
closeness	linear	11	0.8383

Secondly, take is-couple as known attribute with other attributes to predict closeness levels and take closeness information for is-couple as well. Then predict both of them separately once again. Highest results are selected using ELM from different parameters in Table 6.

Compared with Table 5, almost all accuracies of is-couple some accuracy of closeness get higher and few accuracies of closeness become lower. The results show our multiple target attributes prediction adjustments strategy does work.

Thirdly, real is-couple value will be with other attributes to predict closeness levels and similar operation will be performed for is-couple as well. The highest results are selected using ELM from different parameters in Table 7.

There is no wonder almost all accuracies in Table 7 are higher than that of other tables, because real value is with 'prediction accuracy' of 1. And the results from our strategy are very close to directly prediction.

6 Conclusion and Future Work

The main contributions of our chapter includes:

1. constructing an universal model which could apply on a variety of dataset;
2. utilizing multiple state-of-art classifiers to compare with Extreme Learning Machine;
3. proposing some categories to improve the performance of Extreme Learning Machine.

Regarding our work, we are going to continue to improve the performance of our model based on ELM, which is intended to improve the accuracy of demographic attributes prediction. As to predicting action types based on accelerometer data, we are planning to increase the variety of action types. Besides predicting, more works will be done to study how to prevent predicting demographic information by protecting key demographic information.

Acknowledgments This work is supported by the National Natural Science Foundation of China under Grand No.61073062, No. 61100027, No.61202085, National Research Foundation for the Doctoral Program of Higher Education of China under Grand No. 20120042120010, Liaoning Province Doctor Startup Fund under Grand No.20111001, No.20121002, Fundamental Research Funds for the Central Universities under Grand No.N110417001No.N110417004.

References

1. J. Hu, H.-J. Zeng, H. Li, C. Niu, Z. Chen, Demographic prediction based on user's browsing behavior, in *Proceedings of the 16th international conference on World Wide Web* (ACM, 2007) pp. 151–160
2. S.M.S.J.T. Nadeem, M.C.Weigle, Demographic prediction of mobile user from phone usage. Age **1**, 16–21 (2012)
3. J.E. Blumenstock, D. Gillick, N. Eagle, Whos calling? demographics of mobile phone use in rwanda. Transportation **32**, 2–5 (2010)
4. J.K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, The mobile data challenge: Big data for mobile computing research, pp. 1–8, 2012
5. S. Brdar, D. ulibrk, V. Crnojevi, Demographic attributes prediction on the real-world mobile data, in *MDC* 2012
6. E.Z. Kaixiang Mo, B. Tan, Q. Yang, Report of task 3: your phone understands you (2012)
7. N. Ravi, N. Dandekar, P. Mysore, M.L. Littman, Activity recognition from accelerometer data. Am. Assoc. Artif. Intell. **20**(3), 1541 (2005)
8. F. Cao, B. Liu, D. Sun Park, Image classification based on effective extreme learning machine. Neurocomputing **102**, 90–97, 15 February 2013

9. G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006)
10. C. Pan, D.S. Park, H. Lu, X. Wu, Color image segmentation by fixation-based active learning with elm. Soft Comput. **16**(9), 1569–1584 (2012)
11. X.-G. Zhao, G. Wang, X. Bi, P. Gong, Y. Zhao, Xml document classification based on elm. Neurocomputing **74**(16), 2444–2451 (2011)
12. I. Marques, M. Graña, Face recognition with lattice independent component analysis and extreme learning machines. Soft Comput. **16**(9), 1525–1537 (2012)
13. S.R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology. Syst. Man Cybern. IEEE Trans. **21**(3), 660–674 (1991)
14. C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines. Neural Networks IEEE Trans. **13**(2), 415–425 (2002)
15. N. Eagle, A. Pentland, Reality mining: sensing complex social systems. Pers. ubiquit. comput. **10**(4), 255–268 (2006)