Predicting Users' Age Range in Micro-blog Network

Chengyu Wang, Bing Xiao, Xiang Li, Jiawen Zhu,
Xiaofeng He, and Rong Zhang

Software Engineering Institute, East China Normal University, Shanghai, China
{chengyuwang,bingxiao,xiangli,jiawenzhu}@ecnu.cn,
{xfhe,rzhang}@sei.ecnu.edu.cn

Abstract. In this report, we present our work on WISE 2013 Challenge Track II to predict the age range of Weibo users. In this challenge, a dataset consisting of Sina Weibo user information was presented. The goal of the challenge is to predict users age range. With personal information and original tweets for over one million users as training data, we analyze and process the dataset, and experiment a series of prediction methods including SVM, decision tree etc. The result shows that ensemble classifiers based on AdaBoost achieves the best prediction results in this challenge.

Keywords: WISE challenge, classification, AdaBoost.

1 Introduction

Sina Weibo (http://weibo.com) is one of the most popular micro-blog services in China[1]. In this challenge, we are given a dataset consisting of Weibo users with personal information. The goal of the challenge is to predict the age range of a Weibo user.

Parsing the original data is needed firstly. Files of data where Chinese words have been mapped to numbers were too large and interspersed, so we remap the numbers in every attributes together. As thus we can use these unique numbers directly rather than separate by the attribute name. Feature selection begins after finding that the number of features is over 300 thousand and most of the words are rarely used. The lengths of job, education and other attributes of different users are regarded as features here. Entropies of different attributes in each age class can help in the feature selection process. The remainder of this paper proposes several prediction methods and models that have been applied during the experiment. Three categories of prediction approaches are introduced including building classifiers, classifying via regression and ensemble classifiers. The result suggests that adaptive boosting methods can greatly improve the performance and achieve higher accuracy in prediction. The conclusion is given in the last section.

X. Lin et al. (Eds.): WISE 2013, Part I, LNCS 8180, pp. 516–521, 2013.

2 Dataset Description and Data Preprocessing

The WISE 2013 Challenge Track II Dataset includes five data files with trainIDBIRTH.txt, trainInfos.txt and traintweets.txt being training set, while testInfos.txt and testtweets.txt being the testing set.

The data formats of each data file are listed:

- trainIDBIRTH.txt: userid birthdate
- trainInfos.txt and testInfos.txt: userid tags jobs education description
- traintweets.txt and testtweets.txt: userid timestamp1,tweet1 timestamp2, tweet2 timestamp3,tweet3...

In the dataset, only original tweets are preserved and segmented. Note that contents with Chinese are preprocessed by mapping each Chinese character to a unique number. So there are only alphanumeric contents in the dataset that has been optimized and preprocessed. The size of training data is 1126049. The size of testing data is 17519.

In trainIDBIRTH.txt, users' IDs and birthdates are listed, where we can get the age of each user first. In trainInfos.txt, each line is corresponding to one user and his or her personal information(user ID, tags, jobs, personal description, age, gender and education) is included.

We write Java programs to parse the original data files, namely trainIDBIRTH.txt, trainInfo.txt and traintweets.txt. The age is calculated using the birthdate of the corresponding user. The user ages are split into following four age ranges.

- Range 1: age younger than or equal to 18 years old;
- Range 2: age older than 18 but younger than or equal to 24 yesrs old;
- Range 3: age older than 24 but younger than or equal to 35 yesrs old;
- Range 4: age older than 35 years old.

The class is equivalent of the age range of the user.

As for text, different elements from different attributes can have the same number since contents in Chinese are preprocessed by mapping each Chinese word to a unique number. However, the same number from different attributes should not be treated as the same. We remap these numbers into a new vector space.

After the remapping process, we generate the following statistics.

3 Feature Creation and Selection

3.1 Feature Creation

We first use the bag-of-words model to generate features. It is commonly used in methods of document classification where the occurrence of each word is used as a feature for training a classifier. We can apply this method to different attributes mentioned above and all the words have already been remapped.

However, this method should not be completely applied in our problem because of the following reasons:

Table 1. Remapping Data

Attribute	Range	Total Number
tags	0-58944	58945
jobs	58945-80747	21803
education	80748-94027	13327
description	94028-218885	124881
tweet	21886-391821	172936

1. The number of features is over 300 thousand thus is already too large to handle. The training process can be extremely time-consuming and memory-consuming.
2. According to the statistics, over 90 percent of the words are used by only a small portion of the users and only a little part of the vocabulary are frequently used.

According to our dataset, the tweets posted by different micro-blog users may have various lengths. It proves true for personal information provided by users, too. For example, younger users, especially teenagers, have a tendency to provide little information in their jobs column since most of them do not have a job. We retrieve the lengths of job, education and other attributes of different users and regard them as features.

3.2 Feature Selection

As mentioned above, only a small part of words are frequently used. We employ a larger threshold for tweet words while for others, we suggest a smaller threshold. Next, we calculate the number of word occurrences in data from four classes (age ranges) for every word. Entropy is defined as:

$$H(x) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{1}$$

Then, we calculate the entropies of different tags. Similarly, we get entropies of jobs, education, etc. as well. According to information theory, lower entropy suggests that only certain group of users have a high probability to use the corresponding word. Thus, we use features that have low entropies.

We also apply forward search algorithm to further reduce the number of features in order to find a good feature subset. Also, Principle Components Analysis (PCA) can reduce dimensionality but our experiments show that it will make the performance of our classifier worse.

4 Age Range Prediction Methodology

4.1 Classification

In our age range prediction approach, we define four classes {1, 2, 3, 4} where class 1 is defined as age range 1 (less than 18) and class 2 is defined as age range

2 (larger than or equal to 18 but less than 24) and so on. The object is to learn a model f such that f: X ⇒ {1, 2, 3, 4}. In this section, we introduce a series of classifiers we used, including Naive Bayes and Support Vector Machine with different kernels.

Naive Bayes classifiers[2] are simple classifiers based on Bayes' theorem. Although strong independence assumptions almost never hold true in the real world, Naive Bayes classifiers can be trained efficiently in supervised learning.

A support vector machine[3] constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space, making the separation easier in that space. The mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function selected to suit the problem. We use two kinds of kernel functions: polynomial kernel and Gaussian radial basis function. The formulas can be written as:

Polynomial kernel:

$$k(x_i, x_j) = (x_i^T . x_j + 1)^d \tag{2}$$

Gaussian radial basis function:

$$k(x_i, x_j) = exp(-\gamma |x_i - x_j|^2) \tag{3}$$

Since we wish to solve a multiclass classification problem, we reduce the single multiclass problem into multiple binary classification problems. Using the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with the most votes determines the instance classification.

4.2 Classification via Regression

Because we can use a regression model to predict the age of micro-blog users, we can solve the classification problem in a regression approach. The problem can be solved in two steps:

Step1: build a model f such that f: X ⇒ Y where Y is the predicted age.

Step2: calculate the class label L such that:

$$\begin{gathered} Y \in [0, 18] \Rightarrow L = 1 \\ Y \in (18, 24] \Rightarrow L = 2 \\ Y \in (24, 35] \Rightarrow L = 3 \\ Y \in (35, +\infty) \Rightarrow L = 4 \end{gathered}$$

We again use SVM to perform the regression task. Same to SVM classifiers, a non-linear function is produced by linear learning machine mapping into high dimensional kernel induced feature space. The capacity of the system is controlled by parameters that do not depend on the dimensionality of feature space.

4.3 Ensemble Classifiers

We use machine learning meta-algorithm such as bagging and boosting to train weak classifiers and then add them to a final strong classifier. The combination of weak classifiers can improve the performance of our classification model.

A random forest[4] is a classifier consisting of a collection of tree-structured classifiers. It constructs a multitude of decision trees at training time and outputs the class that is the majority of the classes output by individual trees. The method combines the bagging method and the random selection of features in order to construct a collection of decision trees with controlled variation.

The Adaboost[5] algorithm maintains a set of weights over the original training set and adjusts these weights after each classifier is learned. On each round, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased.The new classifier focuses on the examples which have been classfied incorrectly.

We employ AdaBoost in conjunction with Decision Tree learning algorithms to improve the performance.

5 Experiments and Results

In this section, we report our results regarding the performance of Naive Bayes Model, Support Vector Machine with polynomial kernel, Support Vector Machine with Gaussian radial basis kernel, Classification via Support Vector Regression, Random Forest Model and AdaBoost Decision Tree.

We employ accuracy to evaluate the performance of our prediction models. Recall that for classification tasks, the terms true positives, true negatives, false positives, and false negatives compare the results of the classifier under test with trusted external judgments. The terms positive and negative refer to the classifier's prediction (expectation), and the terms true and false refer to whether that prediction corresponds to the external judgment (observation). The term accuracy is defined as:

$$Accurary = \frac{tp + tn}{tp + tn + fp + fn} \tag{4}$$

The accuracies of different models are shown in the table.

Table 2. Accuracies

Model	Accuracy(%)
Naive Bayes	58.6
SVM with polynomial kernel	75.43
SVM with RBF kernel	78.65
SVM Regression	76.42
Random Forests	65.25
AdaBoost Decision Tree	**83.2**

From the statistics, we observe that simple model such as Nave Bayes is least accurate. This is because the Nave Bayes model has a relatively large bias against the real model but the training time is the shortest among all models.

Support Vector Machine can efficiently perform non-linear classification thus yield a much better result than Nave Bayes model. It is also suggested that using Gaussian radial basis kernel to map inputs into high-dimensional feature spaces has a slightly better performance than polynomial kernel. Using Support Vector Machine for Regression can improve performance, too. The main drawbacks of Support Vector Machine are that the training process consumes much larger memory space and longer time, especially when the parameter C is relatively large. The trained models have a smaller bias but a larger variance on different testing sets.

Since single strong classifiers do not enjoy a satisfying accuracy, we continue to use ensemble classifiers. Random Forests are applied to construct trees considering randomly selected features. Although the model enjoys a good accuracy in training, it has a tendency to over-fit when different testing tests are used. However, AdaBoost Decision Tree works well and is less susceptible to the over-fitting problem than other learning algorithms in our experiments. Although a single Decision Tree model does not produce a high accuracy, it is still useful in the final linear combination of classifiers.

6 Conclusions

In this paper, we introduced several approaches based on machine learning techniques to predict Micro-blog users' age range in WISE 2013 Challenge. The dataset provided by the WISE 2013 Challenge Committee was optimized and processed. A series of features were generated from the dataset. We presented and analyzed prediction models and it was suggested that ensemble classifiers, especially using adaptive boosting methods, can greatly improve the performance and achieve higher accuracy in prediction. Therefore, we can come to the conclusion that the approaches we presented were valid and effective.

References

1. Wu, X., Wang, J.: How about micro-blogging service in China: analysis and mining on sina micro-blog. In: Proceedings of 1st International Symposium on From Digital Footprints to Social and Community Intelligence. ACM (2011)
2. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI 1998 Workshop on Learning for Text Categorization, vol. 752 (1998)
3. Gunn, S.R.: Support vector machines for classification and regression. ISIS technical report 14 (1998)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40(2), 139–157 (2000)