Personality Prediction for Microblog Users with Active Learning Method

Xiaoqian Liu[1], Dong Nie[2], Shuotian Bai[2], Bibo Hao[2], and Tingshao Zhu[1(✉)]

[1] Institute of Psychology, Chinese Academy of Sciences, Lincui Road No. 16, Chaoyang District, Beijing 100101, China
{liuxiaoqian,tszhu}@psych.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
ginobilinie@gmail.com, baishutian10@mails.gucas.ac.cn, haobibo12@mails.ucas.ac.cn

Abstract. Personality research on social media is a hot topic recently due to the rapid development of social medias well as the central importance of personality in psychology, but it is hard to acquire adequate appropriate labeled samples. Our research aims to choose the right users to be labeled to improve the accuracy of predicting. a few labeled users, the task is to predict personality of other unlabeled users Given a set of Microblog users' public information (e.g., number of followers) and. The active learning regression algorithm has been employed to establish predicting model in this paper, and the experimental results demonstrate our method can fairly well predict the personality of Microblog users.

Keywords: Active learning · Personality · Online behavior

1 Introduction

Personality can be defined as a set of characteristics which make a person unique, and the study of personality is of central importance in psychology. Among personality theories, Big-Five theory is the most broadly used one, which proposes five basic traits to form human personality: extraversion (Extr.), agreeableness (Agre.), conscientiousness (Cons.), neuroticism (Neur.) and openness (Open.) [5].Conventional personality research usually uses self-reported inventory [24], which is inefficient. The rapid development of internet makes it possible to analyze behaviors and personality traits of Web users, which attracts much attention of scientists from different disciplines [8,3,23].With the wide spread of Microblog nowadays, Sina Microblog (Weibo) becomes one of the most popular internet services in mainland China with more than 300 million register users [13]. Weibo has become an important part of normal life for some people [4]. Microblog provides an ideal online platform for personality research.

Much work has been done to identify the relationship between social media usage and personality [6,18,8], and some focus on predicting personality for social media users [1]. It is difficult to acquire labeled data from social media, not only time-consuming and expensive, but also privacy concerns. Therefore, it is very important

Q. Zu et al. (Eds.): HCC 2014, LNCS 8944, pp. 41–54, 2015.
DOI: 10.1007/978-3-319-15554-8_4

to find appropriate users to be invited to complete inventory test as labeled samples, as it is much cheaper to obtain user data (unlabeled).With sampling technique, it is possible to choose the users.

Recently, there has been much work on active learning, which initially copes with only the unlabeled portion of a pool of examples drawn from underlying distribution. A good classifier or regression model can be learned with significantly fewer labels by actively directing the queries to find informative examples. There are significant practical benefits in minimizing the number of labeled examples in settings where there is no shortage of unlabeled data but labels are expensive.

In this paper, we propose a pool-based active learning regression model to choose the most appropriate Microblog users from a large amount of Sina Microblog user data. For these selected users, we conduct personality tests to acquire their personality trait values, and then a predictive regression model is trained. The typical batch learning directly focuses on the labeled data, our method can exploit the cheaper unlabeled data to choose the best points as the labeling data, and thus, we can improve the efficiency of predictive process.

The rest of the paper is organized as following: Section 2 introduces some related work conducted by other researchers. We describe the details of our data in Section 3. Section 4 mainly describes our exploited methods, and the results will be thoroughly discussed in Section 5. Section 6 concludes our work in this paper and gives a brief discussion on future work.

2 Related Work

Factor Personality analysis based on social media has acquired considerable attention recently [6,14,1,8]. They mainly collected Internet data and corresponding labeled data, and then applied supervised learning approaches, such as, classification or regression, to build the model.

Given the training data is always scarce, it is often easy to retrieve large scale of unlabeled data. Many methods have been proposed to address this problem, such as active learning algorithms [17] for sample selection and semi-supervised learning algorithms [25]. In this paper, we mainly focus on active learning methods. A lot of work has been done on active learning classifiers. For example, Query by Committee (QBC) method trains a committee of students on the same dataset, and choose the next query according to principle of maximal disagreement [21]. Uncertainty sampling [10] selects the example on which the current learner has lowest certainty. These active learning methods aims to reduce version space [15]. Recently many new active learning methods are proposed based on objective function optimization. For example, [20] re-derive the variance reduction method known in experimental design circles as ‘A-optimality’. [2] finds Agnostic Active can achieves an exponential improvement. There are also works on active learning regression models. [22] discusses the problem of active learning in linear regression scenarios, and adopts weighted least-squares learning to form a new active learning regression method. As the distribution of testing samples may be unknown sometimes, [11] proposes pool-based active learning algorithm using bias re-sampling technique to solve the unknown density problem.

The key idea behind the above active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. However, many algorithms are just for single dependent variable prediction. As for our case, there are five personality dimensions that have to be predicted. Therefore, it is very suitable to propose a multi-dependent variable active learning method to continue our research.

3 Data

The dataset consists of labeled Weibo users and huge un-labeled ones, and we acquire 1792 copies of Sina Microblog users' information together with personality scores as label. As it is much cheaper to retrieve the Microblog user data while much more expensive to acquire users' personality trait values, therefore, our research is a perfect scene for active learning to choose the most useful test input points.

In this paper, the 1792 users are used as a pool for active learning.

3.1 Data Collection

Using Sina Microblog API (http://open.weibo.com), we first collected 999, 9999 Microblog user IDs, then randomly chosen 100,000 user IDs, and crawled down their Microblog data. Using Weibo "@ function", we invited volunteers to complete big-five inventory online, and acquired 1792 copies of qualified questionnaires. Hence, we had 1792 copies of personality labeled data. The whole process took over two months, and volunteers had got reimbursement in return. The collected Microblog-user dataset (1792 copies of labeled Microblog data) is examined in our experiment. The specific process is depicted in Fig. 1.

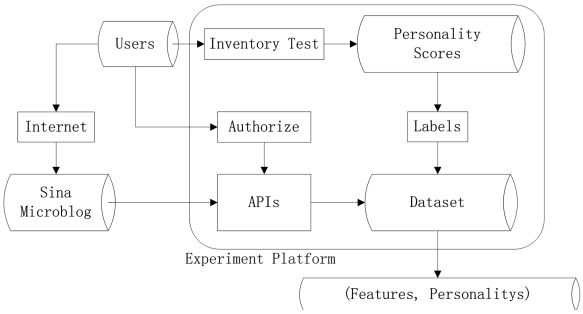

Fig. 1. Flowchart of data collection for our research

3.2 Feature Extraction

As our collected Microblog dataset is relatively simple (contain little behavior data), the preprocessing is straight forward. There are two types of features, summarized features from raw data directly and statistical features. We have totally extracted 47 features for each user from the Microblog data, and the complete feature list is

presented in Table 4 in Appendix. The extracted features can be divided into several categories as bellow:

(1) personal profile, including nickname, address, gender, tags, birthday, personalized domain url, description and so on.
(2) friends and followers.
(3) statues, re-tweeted status, annotations, pictures and comments.
(4) time to post status.
(5) microblog involvement degree.
(6) others.

For some features, we just process the original data directly, for example, the number of posts. We analysis users' description to identify each description is positive, neuter or negative. To deal with the time of posting, we divide a whole day into 7 periods: 0:00-6:00, 6:00-8:00, 8:00-11:00, 11:00-13:00, 13:00-17:00, 17:00-20:00, and 20:00-24:00 , and they correspond to period0, period1, period2, period3, period4, period5 and period6 respectively. We then count the number of posts that the user creates in each period.

After feature extraction, we normalize the data to make the data equally distributed as follows,

$$x = (x - MinValue) / (MaxValue - MinValue) \tag{1}$$

where x is the value of one dimension for a user, while *MinValue* and *MaxValue* respectively represent the maximum and the minimum value of this feature dimension for all users.

3.3 Feature Reduction

Usually, multidimensional data may be represented approximately in fewer dimensions due to redundancies in data, which may improve the accuracy of prediction [19]. Since the original feature space has 47 dimensions, we attempt to take singular value decomposition (SVD) method [7], which is a well-known matrix factorization technique, to reduce dimensionality of feature space [9,12]. We describe the SVD methods as follows.

Suppose $A_{n\times 47}$ is the original data space, then we use SVD technique to factor A into three matrices:

$$A_{n\times 47} = U_{n\times r} \sum_{r\times r} V_{r\times 47} \tag{2}$$

where A is users' online feature space, matrix $\sum$ is a diagonal matrix containing the singular values of the matrix A, here are exactly r singular values, where r is the rank of matrix A. The rank of a matrix is the number of linearly independent rows or columns in the matrix, and it means independent information in our data space here.

We can simply keep the first k singular values in $\sum$, where $k \le r$. This can give us the best rank-k approximation to original data space A, and thus has effectively reduced the dimensionality of our original space. In our experiment, the dimensionality of original feature space is reduced to 28.

4 Methods

We collected 1792 Microblog users' public information using Sina Microblog API, and then extracted features, conducted active learning method to choose 100 users to complete personality test, then a predictive regression model is trained, and we tested the model over whole user set to verify the performance.

As personality traits are all measured by real-value scores between 0 and 5, thus the problem is the prediction of continuous value, in other words, a fitting process. As we need to pay a high price for acquiring personality labels, we choose the most appropriate users to conduct a self-report test. In this paper, we adopt pool-based active learning in approximate linear regression (P-ALICE) algorithm [11] to make the best choice.

4.1 Formulation of Pool-Based Active Learning Method

The goal of P-ALICE is, from the pool of whole users $\{\mathrm{x}_j^{te}\}_{i=1}^{n_{te}}$, to choose the most useful users $\{\mathrm{x}_j^{tr}\}_{i=1}^{n_{tr}}$ for obtaining personality trait values $\{y_i^{tr}\}_{i=1}^{n_{tr}}$, as personality have five dimensions, here, $y_i^{tr} = (y_{E_i}^{tr}, y_{C_i}^{tr}, y_{A_i}^{tr}, y_{N_i}^{tr}, y_{O_i}^{tr})$. The details of the algorithm are described as follows.

4.2 Weighted Least-Squares for Linear Regression Models

To predict the continuous personality traits values, the following linear regression model is used:

$$\hat{f}(\mathrm{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\mathrm{x}) \tag{3}$$

where the true value for x is f(x). Therefore we can calculate the generalization error over the whole test input points:

$$G = \int (\hat{\mathrm{f}}(\mathrm{x}) - \mathrm{f}(\mathrm{x}))^2 \, p_{test}(\mathrm{x}) \, \mathrm{dx} \tag{4}$$

where $p_{test}(\mathrm{x})$ is the density of the whole test input points. Correspondingly, the expected generalization error is:

$$\mathrm{EG} = \mathrm{B} + \mathrm{V} \tag{5}$$

where

$$B = \int (\mathrm{E}\hat{\mathrm{f}}(\mathrm{x}) - \mathrm{f}(\mathrm{x}))^2 p_{test}(\mathrm{x})\,\mathrm{dx} \tag{6}$$

$$V = \int (\mathrm{E}\hat{\mathrm{f}}(\mathrm{x}) - \hat{\mathrm{f}}(\mathrm{x}))^2 p_{test}(\mathrm{x})\,\mathrm{dx} \tag{7}$$

From Equation (5), it is impossible to estimate generalization error before observing output samples because the bias depends on the unknown target function f(x). If the model is correctly specified, the bias term is zero, and in this case, it is possible to choose the best labeling points. However, in our research, the model cannot be correctly specified, which means we have to take some measures to reduce the bias term to zero.

We make a further step about f(x):

$$\mathrm{f(x)=g(x)+r(x)} \tag{8}$$

where $g(\mathrm{x}) = \sum_{i=1}^{b} \alpha_i^* \varphi_i(\mathrm{x})$ and $r(\mathrm{x}) \approx 0$. Obviously, g(x) is an approximately correct model. We can now conduct a further decomposition of bias.

$$B = B_{out} + B_{in} \tag{9}$$

where

$$B_{out} = \int (g(\mathrm{x}) - \mathrm{f}(\mathrm{x}))^2 p_{test}(\mathrm{x})\,\mathrm{dx} \tag{10}$$

$$B_{in} = \int (E\hat{f}(\mathrm{x}) - g(\mathrm{x}))^2 p_{test}(\mathrm{x})\,\mathrm{dx} \tag{11}$$

As B_{out} is always constant, it can be ignored. However, the difference of input distributions causes B_{in} not to be zero. In active learning, this problem is called covariate shift and it is well studied by [16]. Weighted least-squares method can be used for learning the parameters of Equation(3).

$$\hat{\alpha} = \arg\min_{\alpha}[\sum_{i=1}^{n_{tr}} \omega(\mathrm{x}_i^{tr})(\hat{\mathrm{f}}(\mathrm{x}_i^{tr}) - y_i^{tr})^2] \tag{12}$$

where $\omega(\mathrm{x})$ is a weight function. Let X be the $n_{tr} \times t$ matrix with the (i,l) element $X_{i,l} = \varphi_l(\mathrm{x}^{tr}_i)$. Let W be the $n_{tr} \times n_{tr}$ diagonal matrix with the i-th diagonal element

$$W_{i,i} = \omega(\mathrm{x}_i^{tr}) \tag{13}$$

Then $\hat{\alpha}$ is given by $\hat{\alpha} = L\mathrm{y}^{tr}$, where

$$L = (\mathrm{X}^T \mathrm{W} \mathrm{X})^{-1}\, \mathrm{X}^T W \tag{14}$$

and

$$y^{tr} = (y_1^{tr}, y_2^{tr}, \ldots, y_{n_{tr}}^{tr})^T \tag{15}$$

Algorithm 1 Pseudo code of P-ALICE algorithm

Require: Test input points $\{x_j^{te}\}_{i=1}^{n_{te}}$, basis functions $\{\varphi_l(x)\}_{l=1}^{t}$, parameters $\{\lambda_i\}_{i=1}^{b}$ and training set size n_{tr}

Ensure: Learned parameters $\widehat{\alpha}$

Compute $t \times t$ matrix $\widehat{U}$ with $\widehat{U} \leftarrow \frac{1}{n_{te}} \sum_{m,n}^{t} \varphi_m(x) \varphi_n(x)$

while $\lambda \in \{\lambda_i\}_{i=1}^{b}$ **do**

Compute $\{b_\lambda(x_j^{te})\}_{j=1}^{n_{te}}$ with

$$b_\lambda(x) \leftarrow \left(\sum_{m,n}^{t} \left[\widehat{U}^{-1}\right]_{m,n} \varphi_m(x) \varphi_n(x) \right)$$

Choose $X_\lambda^{tr} \leftarrow \{x_i^{tr}\}_{i=1}^{n_{tr}}$ from $\{x_j^{te}\}_{i=1}^{n_{te}}$ using bias resmapling function $b_\lambda(x)$

Compute the $n_{tr} \times t$ matrix X_λ with $[X_\lambda]_{i,l} \leftarrow \varphi_l(x_i^{tr})$

Compute the $n_{tr} \times n_{tr}$ diagonal matrix W_λ with $[W_\lambda]_{i,i} \leftarrow (b_\lambda(x_i^{tr}))^{-1}$

Compute

$$L_\lambda \leftarrow \left(X_\lambda^T W_\lambda X_\lambda\right)^{-1} X_\lambda^T W_\lambda$$

Compute

$$P - ALICE(\lambda) \leftarrow tr\left(\widehat{U} L_\lambda L_\lambda^T\right)$$

end while

Compute

$$\widehat{\lambda} \leftarrow \arg\min_{\lambda} P - ALICE(\lambda)$$

Obtain training output values $y^{tr} \leftarrow (y_1^{tr}, y_2^{tr}, ..., y_{n_{tr}}^{tr})$ at the above $X_{\widehat{\lambda}}^{tr}$

Compute $\widehat{\alpha} \leftarrow L_{\widehat{\lambda}} y^{tr}$

Compute regression model

$$\widehat{f}(x) \leftarrow \sum_{i=1}^{t} \widehat{\alpha}_i \varphi_i(x)$$

4.3 Pool-Based Active Learning Regression

Pool-based active learning algorithm adopts bias sampling method to choose candidate training input points, and the resampling bias function is b(x). According to [11], a heuristic resampling bias functions with parameter λ are given by:

$$b_\lambda(\mathrm{x}) = \left(\sum_{m,n=1}^{t} [\hat{U}^{-1}]_{m,n} \varphi_m(\mathrm{x}) \varphi_n(\mathrm{x}) \right)^{\lambda} \tag{16}$$

where $\hat{U}$ is a $t \times t$ matrix with the (m,n) element:

$$\hat{U}_{m,n} \leftarrow \frac{1}{n_{te}} \sum_{i=1}^{t} \varphi_m(\mathrm{x}_i) \varphi_n(\mathrm{x}_i) \tag{17}$$

As taken bias re-sampling technique, the relation between training set density and test set density becomes:

$$p_{train}(\mathrm{x}) = p_{test}(\mathrm{x})\, \mathrm{b}(\mathrm{x}) \tag{18}$$

The quality of the candidate training input points $\{x_i^{tr}\}_{i=1}^{n_{tr}}$ is evaluated by

$$P - ALICE = tr(\hat{U}LL^T) \tag{19}$$

The weight function $\omega(x)$ included in Equation (13) is defined as

$$\omega(x_j^{tr}) = \frac{1}{b(x_j^{tr})} \tag{20}$$

Thus, the chosen points are calculated as follows:

$$p_{train}(x) = \min_{p_{train}} P - ALICE \tag{21}$$

A pseudo code of the proposed pool-based active learning algorithm is described in Algorithm 1.

5 Experiments and Results

In our experiment, we first adopted P-ALICE algorithm to choose 100 users, and then using Weibo "@ function", we invited the 100 users to complete big-five inventory online, thus we acquired the 100 users' personality trait values. Correspondingly, we successfully trained an approximately linear regression model.

The 1792 samples are used as the pool of test input points $(n_{te} = 1792)$, and we set the training set size to be at 100 $(n_{tr} = 100)$. As we cannot exactly know the relation between personality and online behaviors, we employ the following linear regression model for learning:

$$\hat{f} = \sum_{l=1}^{t} \alpha_l \exp(-\frac{\|x - c_l\|^2}{2}) \tag{22}$$

where $\{c_l\}_{l=1}^{t}$ are template points randomly chosen from the pool of test input. In fact, here we set t = 15 and we will present the details next.

As for parameter λ in Equation (16), we set the value to 0.6, and the chosen method will be discussed as follows.

5.1 Experimental Results

To evaluate the active learning regression method, we adopted the following baseline methods (these three methods are all passive algorithms:

(1) OLS: Training points are drawn uniformly from the 1792 users pool, and the other settings are the same with P-ALICE except $\lambda = 0$ for Equation (16).

(2) LR: Linear regression is a broadly used parametric model, it calculates model parameter β via the training set, and then the prediction value can be computed by the trained model:

$$Y_i = \beta_1 X_{i1} + \ldots + \beta_p X_{id} + \varepsilon_i = X_i^T \beta + \varepsilon_i, \mathrm{i} = 1, \ldots, \mathrm{n} \quad (23)$$

(3) LLKR: Local linear kernel regression is a widely used non-parametric model, we select gaussian kernel in this paper.

$$\min(\mathrm{Y}_i - \mathrm{m} - (\mathrm{X}_i - \mathrm{x})^T \beta)^2 K(\frac{\mathrm{X}_i - \mathrm{x}}{h}) \quad (24)$$

with respect to m and β, where h is bandwidth.

In our research, we take three criterions to measure the results.

(1) Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{f}(x_i) - \mathrm{y}_i \right| \quad (25)$$

(2) Root Mean Squared Error (RMSE):

$$RMSE = \frac{1}{n} \sqrt{\sum_{i=1}^{n} (\hat{f}(x_i) - \mathrm{y}_i)^2} \quad (26)$$

(3) Correlation Coefficient (CORR)

$$\rho_{X,Y} = \frac{\mathrm{cov}(\mathrm{X}, \mathrm{Y})}{\sigma_X \sigma_Y} = \frac{\mathrm{E}[(X - \mu_X)(\mathrm{Y} - \mu_Y)]}{\sigma_X \sigma_Y} \quad (27)$$

The results of MAE, RMSE and CORR are shown in Table1, Table2 and Table3, and we have conducted experiments over 10 trials, the best results are highlighted in bold.

From Table 1 and Table 2, P-ALICE outperforms all three other algorithms on each personality dimension. The OLS algorithm are almost the same settings with P-ALICE, while it performs much worse than the latter, this fully illustrates the significance of active learning. Local linear kernel regression is already a well-performed passive, while P-ALICE also performs better.

Table 3 presents the results of correlation coefficients by all algorithms, the data in this table shows P-ALICE can give a much better results. The correlation coefficient for Cons is 0.2102, even the lowest correlation coefficient given by P-ALICE is more than 0.10 (Extr.), considering the training set size is 100, the performance is well satisfied. To better compare the correlation coefficients, we draw them in Fig.2.

Table 1. MAE achieved by different algorithms for each personality dimension

Dimension	LR	LLKR	OLS	P-ALICE
Extr.	0.7780	0.5655	0.8205	**0.5322**
Agre.	0.8084	0.4502	0.4742	**0.4318**
Cons.	0.8064	0.5379	0.674	**0.5001**
Neur.	0.8856	0.6199	0.6611	**0.5822**
Open.	0.8353	0.5143	0.7171	**0.4825**

Table 2. RMSE achieved by different algorithms for each personality dimension

Dimension	LR	LLKR	OLS	P-ALICE
Extr.	0.9765	0.7031	0.9927	**0.6647**
Agre.	0.9802	0.5669	0.6116	**0.5461**
Cons.	1.0020	0.6710	0.8778	**0.6282**
Neur.	1.0233	0.7671	0.8266	**0.729**
Open.	0.9928	0.6452	0.8805	**0.6018**

Table 3. Correlation coefficients achieved by different algorithms for each personality dimension

Dimension	LR	LLKR	OLS	P-ALICE
Extr.	0.0854	0.0677	-0.0721	**0.108**
Agre.	0.0045	0.1125	0.1083	**0.1516**
Cons.	0.0322	0.0820	0.0704	**0.2102**
Neur.	0.0790	0.1021	-0.0034	**0.1779**
Open.	0.0455	0.1075	-0.0178	**0.1952**

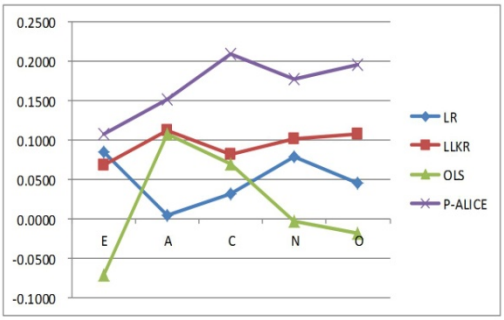

Fig. 2. Correlation coefficients of different algorithms for five personality dimensions

5.2 Parameter Selection Fort

For parameter t, which is the number of Gaussian basis functions, we conduct a search over $t \in \{5, 10, 15, \ldots, 45, 50\}$, and we find 15 as the optimum value. To identify the impact of parameter t, we describe the details in Fig.3. From Fig.3, it is obvious that t = 15 is the best choice for all three criterions.

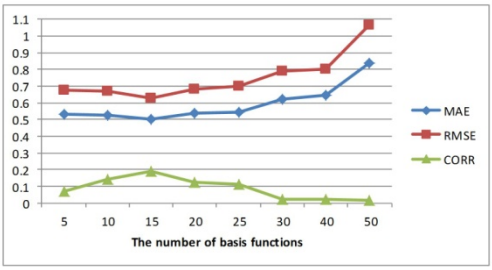

Fig. 3. The impact of parameter t for MAE, RMSE and CORR

5.3 Parameter Selection for λ

For parameter λ, we use the following steps to decide:

(1) First, we roughly draw parameter λ from the following candidate set

$$\lambda \in S_{coarse} = \{0, 0.1, 0.2, \ldots, 1\} \tag{28}$$

and we assume the selected parameter to be λ_{coarse}.

(2) Second, we make it fine, we make the candidate set like the following:

$$\lambda \in S_{fine} = \lambda_{coarse} = \{-0.10, -0.09, \ldots, 0, 0.01, \ldots, 0.10\} \tag{29}$$

and we eventually choose a perfect parameter λ which is set at 0.6.

5.4 The Choice of Basis Function

P-ALICE is used to estimate generalization error before observing output variables, therefore, it is very suitable to use this algorithm in our research. As there are totally five personality traits, and there are five predictive models respectively, if the P-ALICE still efficient in this case? In Section 5, we choose Gaussian kernel function to be basis function (25). This guarantees the feasibility of multiple dependent variable prediction. If we choose polynomial function to be basis function (e.g., $\{x, x^2, \ldots, x^n\}$), it will not work due to the difference of the highest model orders.

6 Conclusions

In this paper, we have investigated personality traits prediction based on Microblog users' public information, and pool-based active learning regression methods are employed to choose the appropriate input samples, the corresponding regression models are trained at the same time. Experiment is conducted over our dataset, and the results demonstrate that our models perform well.

In near future, we intend to continue our experiment on Sina Microblog, inviting more participants to acquire more labeled data and eventually improve active learning

regression model. Furthermore, we plan to take Microblog content into account. We also want to conduct further research on the behavior patterns of other psychological attributes.

Acknowledgment. The authors gratefully acknowledge the generous support from NSFC (61070115), Strategic Priority Research Program (XDA06030800) and 100-Talent Project (Y2CX093006) from Chinese Academy of Sciences.

Appendix

The full list of extracted features is depicted in Table 4.

Table 4. Extracted Features

Feature ID	Feature Name	Description
V1	original_status_count	The number of user's original statuses
V2	status_count	The total number of user's statuses (including retweeted statuses)
V3	picture_count	The total number of user's pictures (pictures in all statuses)
V4	repost_status_count	The total number of user's reposted statuses
V5	comment_count	The total number of user's comments
V6	annotation_count	The total number of user's annotations
V7	original_status_rate	The rate of original statuses
V8	comment_average	The average number comments for each status
V9	profile_degree	The complete degree of personal profile
V10	screen_name_length	The length of screen name
V11	description_length	The length of user's description
V12	followers_count	The number of user's followers
V13	bi_followers_count	The number of user's mutual followers
V14	friends_count	The number of user's friends
V15	fav_status_count	The number of user's favourite statuses
V16	description_evaluation	The sentiment evaluation of user's description
V17	original_pic_count	The total number of user's original pictures
V18	original_pic_rate	The rate of user's original pictures
V19	original_pic_average	The average number original pictures for each status
V20	annotation_average	The average number of user's annotations for each status
V21	domain_url	Whether the user have a personalized domain address
V22	tags_count	The total number of tags
V23	medal_count	The total number of micro-medal on the user's account
V24	microblog_level	The current level of user's microblog account
V25	first_status_period	The period user most likely to give first status per day
V26	last_status_period	The period user most likely to give last status per day
V27	fav_status_period	The period user most likely to give most statuses per day
V28	first_p0	The days user created first status between 0:00 and 6:00
V29	first_p1	The days user created first status between 6:00 and 8:00
...	...	...
V33	first_p6	The days user created first status between 20:00 and 24:00
V34	last_p0	The days user created last status between 0:00 and 6:00
...	...	...
V40	last_p6	The days user created last status between 20:00 and 24:00
V41	fav_p0	The days user created most statuses between 0:00 and 6:00
...	...	...
V47	fav_p6	The days user created most statuses between 20:00 and 24:00
...	...	...

References

1. Bai, S., Zhu, T., Cheng, L.: Big-five personality prediction based on user behaviors at social network sites. arXiv preprint arXiv:1204.4809 (2012)
2. Balcan, M.F., Beygelzimer, A., Langford, J.: Agnostic active learning. J. Comput. Syst. Sci. **75**(1), 78–89 (2009)
3. Buchanan, T., Smith, J.L.: Using the internet for psychological research: Personality testing on the world wide web. Br. J. Psychol. **90**(1), 125–144 (1999)
4. Cao, B.: Sina's weibo outlook buoys internet stock gains: China overnight. Technical report, Bloomberg (2012)
5. D. Funder: Personality. Annu. Rev. Psychol. **52**, 197–221 (2001)
6. Golbeck, J., Robles, C., Turner, K.: Prediciting personality with social media. In: The 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems, pp. 253–262. ACM, New York (2011)
7. Singular value decomposition and least squares solutions: G. H Golub, C. Reinsch. Numer. Math. **14**, 403–420 (1970)
8. Gosling, S., Augustine, A., Vazire, S., Holtzman, N., Gaddis, S.: Manifestations of personality in online social networks: Self-reported facebook related behaviors and observable profile information. Cyberpsychology Behavior and Social Networking **14**, 483–488 (2011)
9. Bourlard, H., Kamp, Y.: Auto-association by multilayer perceptrons and singular value decomposition. Biol. Cybern. **59**, 291–294 (1988)
10. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3–12. Springer, New York (1994)
11. Sugiyama, M., Nakajima, S.: Pool-based active learning in approximate linear regression. Mach. Learn. **75**(3), 249–274 (2009)
12. M.E. Wall, A. Rechtsteiner, L. M. Rocha: Singular value decomposition and principal component analysis. In: A Practical Approach to Microarray Data Analysis, pp. 91–109 (2003)
13. Millward, S.: China's forgotten 3rd twitter clone hits 260 million users. Technical report, techinasia.com (2012)
14. Minamikawa, A., Fujita, H., Hakura, J., Kurematsu, M.: Personality estimation application for social media. In: Frontiers in Artificial Intelligence and Applications, vol. 246 (2012)
15. Mitchell, T.M.: Generalization as search. Artif. Intell. **18**(2), 203–226 (1982)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
17. Prince, M.: Does active learning work? a review of the research. J. Eng. Educ. **93**(3), 223–231 (2004)
18. Qiu, L., Lin, H., Ramsay, J., Yang, F.: You are what you tweet: Personality expression and perception on twitter. J. Res. Pers. **46**, 710–718 (2012)
19. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science **41**(6) (1990)
20. Schein, A.I., Ungar, L.H.: Active learning for logistic regression: an evaluation. Mach. Learn. **68**(3), 235–265 (2007)
21. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Fifth Annual Workshop on Computational Learning Theory, pp. 287–294. ACM, New York (1992)

22. Sugiyama, M.: Active learning in approximately linear regression based on conditional expectation of generalization error. The Journal of Machine Learning Research **7**, 141–166 (2006)
23. Sumner, C., Byers, A., Shearing, M.: Determining personality traits and privacy concerns from facebook activity. Black Hat Briefings **11** (2011)
24. Thompson, E.: Development and validation of an international english big-five mini-markers. Personality Individ. Differ. **45**(6), 542–548 (2008)
25. Zhu, X.: Semi-supervised learning literature survey. Computer Science. University of Wisconsin-Madison (2006)