

PAPER • OPEN ACCESS

Joint Learning with both Classification and Regression Models for Age Prediction

To cite this article: Jing Chen *et al* 2019 *J. Phys.: Conf. Ser.* **1168** 032016

View the [article online](#) for updates and enhancements.

You may also like

- [Data-driven assessment of cardiovascular ageing through multisite photoplethysmography and electrocardiography](#)
Antonio M. Chiarelli, Francesco Bianco, David Perpetuini et al.
- [Spectroscopic Distance, Mass, and Age Estimations for APOGEE DR17](#)
Alexander Stone-Martinez, Jon A. Holtzman, Julie Imig et al.
- [A comparison between diffusion tensor imaging and generalized q-sampling imaging in the age prediction of healthy adults via machine learning approaches](#)
Yingying Guo, Xi Yang, Zilong Yuan et al.

Joint Learning with both Classification and Regression Models for Age Prediction

Jing Chen¹, Long Cheng¹, Xi Yang¹, Jun Liang¹, Bing Quan¹ and Shoushan Li^{2,a}

¹China Mobile (Suzhou) Software Technology Co., LTD, 215000 Suzhou Jiangsu, China

²College of Computer Science, Soochow University, 215000 Suzhou Jiangsu, China

^aCorresponding author: lishoushan@suda.edu.cn

Abstract. Age classification and regression are two main approaches to age prediction in social media, and these two approaches have their own characteristics and strength. For instance, the classification model can flexibly utilize distinguished models in machine learning, while the regression model can capture the connections between different ages. In order to exploit the advantages of both age classification and regression models, a novel approach to age prediction is proposed, namely joint learning for age prediction. Specifically, an auxiliary Long-Short Term Memory (LSTM) layer is employed to learn the auxiliary representation from the classification setting, and simultaneously join the auxiliary representation into the main LSTM layer for the age regression setting. In the learning process, the auxiliary classification LSTM model and the main regression LSTM model are jointly learned. Empirical studies demonstrate that our joint learning approach significantly improves the performance of age prediction using either individual classification or regression model.

1.Introduction

The tremendous growth of the social media, such as Twitter and Facebook, has raised the need for technologies of automatically analyzing online content and users. One basic and fundamental technology is known as age prediction which determines the age information of online users. Figure 1 shows an online user in a social media website. Even though the age information is not available from his/her profiles, we could exactly infer her age to be 23 from her published message “I’M 23”. Age prediction has been an essential pre-processing procedure in many social applications, such as intelligent marketing [1, 2], online advertising [3], and personality analysis [4].

User attribute information: Name: ***; Gender: male; Age: ***;
Social information: #Messages: 100; Followers: ‘2919393812’, ‘3044343944’, ... Followings: ‘1976649967’, ‘2286980683’,...
Textual information(e.g., Messages) <i>UGH I don't wanna go to school tomorrow. DONT WANNA SEE A TEACHERS FACE AGAIN. OH WAIT I'M 23! WHAT TIME IS IT?! GAME TIME!</i>

Figure 1. A user example in a social media



Conventional approaches to age prediction are mainly based on supervised learning approaches where sufficient labeled data is essential for training the model. For example, extracting textual features from the messages posted by the user in Figure 1 to train an age prediction model. Generally, age classification and age regression are two implementation models in age prediction. The former model of age classification aims to classify the users into several different age groups [5], while the later model of age regression focuses on predicting the user's age with a discrete variable representing an exact age number [6-8]. These two different prediction models have their own distinctive characteristics. For instances, the classification model could flexibly take advantage of the discriminative models in machine learning, thus obtaining better performance on age prediction; the regression model is able to capture connections between different ages, meanwhile, has a better modeling for labels of age classification.

In order to exploit advantages of both the classification and regression models, we propose a novel approach, namely joint learning for user age prediction, which learns better representation by integrating both regression and classification representations generated from the classification and regression models respectively. Specifically, we firstly separate the age prediction task into a main task (age regression) and an auxiliary task (age classification), and then propose a joint learning approach to boost the performance of the main task with the help of the auxiliary task. A neural network architecture, namely Aux-LSTM, is proposed for the joint learning, which first learns an auxiliary representation from the auxiliary task with an auxiliary Long Short-Term Memory (LSTM) layer, and then integrate the auxiliary representation into the main task for joint learning.

2. Related work

2.1 Age Classification

In the last decade, there are various studies which focus their researches on age prediction. Most of previous studies model age prediction as a classification problem and conduct their empirical studies on two main domains, i.e., blog [9-11] and social media [12].

In the blog domain, Schler et al. [11] focus on textual features extracted from the blog text, such as word context features and parts-of-speech (POS) stylistic features for age classification. Burger and Henderson [10] explore some social features, such as location, time, and friend features, related to blogger ages. Ikeda et al. [13] is one exception which proposes a semi-supervised approach to age prediction by training multiple sub-classifiers with textual features. Some other studies, such as Rosenthal and McKeown [14] and Goswami et al. [15], explore both the textual and social features in automatic age classification.

In the social media domain, Mackinnon and Warren [12] explore some kinds of social features, i.e., the relationship between users to predict a user's age and country of residence in a social network. Peersman et al. [5] apply a text categorization approach to age classification with textual features only. More recently, Marquardt et al. [16] propose a multi-label classification approach to predict both the gender and age of authors from texts.

2.2 Age Regression

In contrast to age classification, much few studies model age prediction as a regression problem.

Nguyen et al. [6] explore textual features, such as word unigrams, POS unigrams and bigrams, together with gender features in age regression via a linear regression model. Their empirical studies find that word unigrams can achieve reasonable performance and that POS patterns are strong indicators of the old age. Nguyen et al. [7] further explore age prediction of Twitter users with a linear regression model. They find that an automatic system can achieve better performance than a human being. Chen et al. [8] apply active learning approach to age regression for better exploiting the unlabeled data to improve the performance. Chen et al. [9] explore deep learning approach to age regression and combine results of age regression and classification linearly to improve the performance of age prediction.

Different from the above research, we employ a deep learning architecture to model age prediction. More importantly, we propose a joint learning approach with both age classification and regression models. To the best of our knowledge, this is the first attempt to address the age prediction when the two different models are both involved.

3. Background

In this section, we give the background on data collection and features (including textual features and social features) utilized for age prediction.

3.1 Data Collection

Our data is collected from Sina Micro-blog (<http://weibo.com/>), a famous Micro-blogging platform in China. The information of users can be acquired through analyzing their profiles. From the website, we crawl each user's homepage which contains user information (e.g., name, age, gender, Followers, Followings), and their posted messages. The data collection process starts from some randomly selected users, and iteratively gets the data of their followers and followings. We remove those unsuitable users who are verified as organizations because the age attributes of these users make no sense. Besides, although the posted messages are the basic and major factors to predict user age [6], some users post very few messages. To guarantee the reliability of the data, we remove those somehow non-active users who post less than 50 messages. In total, we collect the homepages of about 12000 users, together with their posted messages.

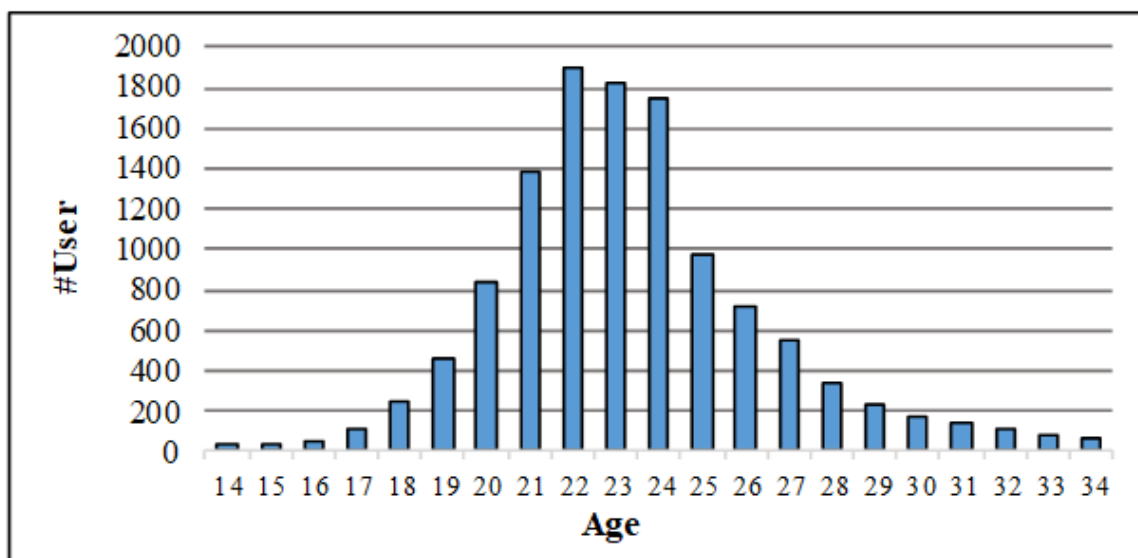


Figure 2. User distribution in different ages

Figure 2 shows the user distribution in different ages. From this figure, we can see that the data distribution of user ages is rather imbalanced. Most users are young whose ages are in the range of 19 to 28.

3.2 Textual and Social features

Each user is represented by a feature vector, i.e., $x_i \in R^d$ as the input in prediction models. In the literature, various features, such as word unigrams, and social behaviors, have been successfully adopted on age prediction [14]. In this study, we categorize these features into two main groups, textual and social features. The former contains the features generated from the user-generated messages, e.g., word unigrams, while the latter contains the features generated from the user social behaviors, e.g., follower list and following list. Table 1 shows all the features of two categories.

Table 1. Textual and Social features in age prediction

	Feature	Remarks
Textual Features	BOW	Word unigrams in the user-generated messages
	POS Patterns	Top trigrams of the POS tag in user-generated messages
Social Features	Statistics	# of Messages, # of Comments, # of Followers, # of Followings
	Time	Probability distribution of the user posts messages over 24 hours (00-23)
	Follower List	All IDs of the followers
	Following List	All IDs of the followings

Among textual features, BOW features are most popular in age prediction, and proven very effective due to the fact that word features reflect concerning topics, which can distinguish users of different ages. POS patterns are also popular textual features to capture the writing styles of the users.

Among social features, four statistical features, i.e., those with # of, capture the social behaviors of a user. For example, # of Messages indicate the numbers of the messages posted by a user and # of Comments indicate the numbers of messages commented from other users. The Time features capture the user habits on posting messages. For example, users of 20-24 ages might be more likely to post their messages very late at night. Followings and followers reflect the interests of users which provide an effective window to infer users' ages.

4. Basic LSTM Models for Age Prediction

4.1 Basic LSTM Network

Long short-term memory network (LSTM) is proposed by Hochreiter and Schmidhuber [17] to specifically address this issue of learning long-term dependencies. The LSTM maintains a separate memory cell inside it that updates and exposes its content only when deemed necessary. A number of minor modifications to the standard LSTM unit have been made. In this study, we apply the implementation used by Graves [18] to map the input sequence of main task to a fixed-sized vector.

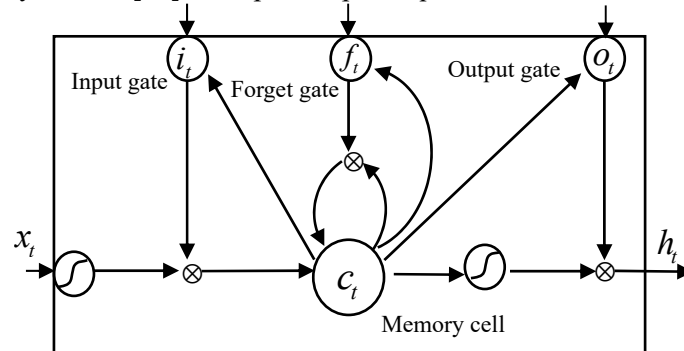


Figure 3. A long short-term memory unit

Figure 3 demonstrates the architecture of a LSTM unit which consists of an input gate i , an output gate o , a forget gate f , a hidden state h , and a memory cell c . At each time step t , the LSTM unit is updated as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}) \quad (1)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}) \quad (2)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1}) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

Where x_t denotes the input at time step t , σ denotes the logistic sigmoid function, $\odot$ denotes elementwise point multiplication. W , U and V represent the corresponding weight matrices connecting them to the gates. Intuitively, the forget gate controls the amount of which each unit of the memory cell is erased, the input gate controls how much each unit is updated, and the output gate controls the exposure of the internal memory state.

4.2 LSTM Model for Age Classification

Figure 4 illustrates the model architecture of age classification with a LSTM layer. We utilize T^{input} to represent the input, and the input propagates through the LSTM layer, yielding the high-dimensional vector, i.e.,

$$h = LSTM(T^{input}) \quad (7)$$

Where h is the output from the LSTM layer.

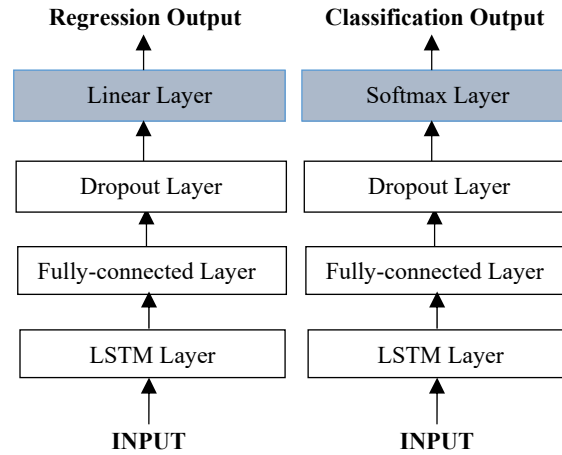


Figure 4. Age Classification and Regression with LSTM

Subsequently, the fully-connected layer which is similar to a hidden layer in the conventional multi-layer perceptron, accepts the output from the previous layer, weighting them and passing through a normally activation function as follows:

$$h^* = \text{dense}(h) = \phi(\theta^T h + b) \quad (8)$$

Where ϕ is the non-linear activation function, employed “relu” in our model. h^* is the output from the fully-connected layer.

The dropout layer has been very successful on feed-forward networks [19]. By randomly omitting feature detectors from network during training, it can obtain less interdependent network units and achieve better performance, which is used as hidden layer in our framework, i.e.,

$$h^d = h^* \cdot D(p^*) \quad (9)$$

Where D denotes the dropout operator, p^* denotes a tune-able hyper parameter (the probability of retaining a hidden unit in the network), and h^d denotes the output from the dropout layer.

The softmax output layer is used for a classification task. The output from the previous layer are then fed into the output layer to get the prediction probabilities, i.e.,

$$p = \text{softmax}(W^d h^d + b^d) \quad (10)$$

Where p is the set of predicted probabilities of the age classification, W^d is the weight vector to be learned, and the b^d is the bias term.

Our age classification model for age prediction is trained to minimize a categorical cross-entropy loss function. Specially, the loss function is defined as follows:

$$loss_c = -\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^l y_{ij} \log p_{ij} \quad (11)$$

Where $loss_c$ is the loss function of the age classification, m is the total number of samples, l is the number of age categories, y_{ij} indicates whether i -th sample truly belongs to the j -th category, p_{ij} refers to the predicted probability.

4.3 LSTM Model for Age Regression

Figure 4 illustrates the model architecture of age regression with a LSTM layer simultaneously. Compare to the model architecture of age classification, we can see that most layers including the LSTM layer, the fully-connected layer, the dropout layer are the same as those in the age classification model, which have been described in subsection 4.2.

Different from age classification, age regression utilizes a linear output layer instead of a softmax layer, i.e.,

$$f = W^d h^d + b^d \quad (12)$$

Where W^d and b^d take the same meaning to age classification above. f is the predicted age value, which is a discrete variable.

For age regression, we employ “mean squared error” for loss function. Specially, the loss function is defined as follows:

$$loss_R = \frac{1}{2m} \sum_{i=1}^m \|f_i - y_i\|^2 \quad (13)$$

Where $loss_R$ is the loss function of age regression, and y_i is the ground truth label of the i -th sample, and f_i indicates the predicted age value of i -th sample, and m is the total number of the training samples.

5. Joint Learning for Age Prediction

Figure 5 delineates the overall architecture of Aux-LSTM which contains a main LSTM layer and an auxiliary LSTM layer. In our study, we consider the age regression task as the main task and the age classification task as the auxiliary task. The goal of the approach is to employ the auxiliary representation to assist the regression performance of the main task. The main idea of our Aux-LSTM approach lies in that the auxiliary LSTM layer is shared by both the main and auxiliary tasks so as to leverage the learning knowledge from both the classification and regression models.

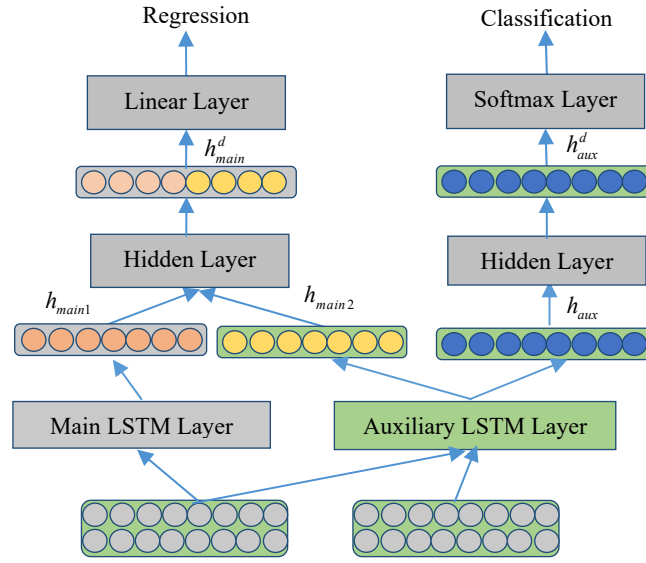


Figure 5. Overall architecture of Aux-LSTM

(1) The Main Task:

Formally, the main regression representation of the main task is generated from both the main LSTM layer and the auxiliary LSTM layer respectively:

$$h_{main1} = LSTM_{main}(T^{input}) \quad (14)$$

$$h_{main2} = LSTM_{aux}(T^{input}) \quad (15)$$

The output h_{main1} represents the representation for the regression model via the main LSTM layer and the output h_{main2} represents the representation for the regression model via the auxiliary LSTM layer.

Then we concatenate the two regression representation as the input of the hidden layer in the main task:

$$h_{main}^d = dense_{main}(h_{main1} \oplus h_{main2}) \quad (16)$$

Where h_{main}^d denote the outputs of fully-connected layer (dense layers) in the main task, and $\oplus$ denotes the concatenate operator.

(2) The Auxiliary Task:

The auxiliary classification representation is also generated by the auxiliary LSTM layer, which is a reused LSTM layer and is employed to bridge across the classification and regression models. The reused LSTM layer encodes both the same input sequence with the same weights:

$$h_{aux} = LSTM_{aux}(T^{input}) \quad (17)$$

Where h_{aux} represents the representation for the classification model via the reused LSTM layer.

Then a fully-connected layer is utilized to obtain a feature vector for classification, which is the same as the hidden layer in the main task:

$$h_{aux}^d = dense_{aux}(h_{aux}) \quad (18)$$

Where h_{aux}^d denote the output of fully-connected layer (dense layer) in the auxiliary task. Other layers including a softmax layer and a linear layer, as shown in Figure 5, are the same as those which have been described in Section 4.2.

(3) Joint Learning:

Finally, we define our joint cost function for Aux-LSTM as a weighted linear combination of the cost functions of both the main task (i.e., the regression task) and auxiliary task (i.e., the classification task) as follows:

$$loss_{Aux-LSTM} = \lambda(loss_R) + (1 - \lambda)(loss_C) \quad (19)$$

In the above equation, λ is the weight parameter, $loss_R$ is the loss function of age regression and $loss_C$ is the loss function of age classification. We take RMSprop [20] as the optimizing algorithm. All the matrix and vector parameters in neural network are initialized with uniform samples in $[-\sqrt{6/(r+c)}, \sqrt{6/(r+c)}]$, where r and c are the numbers of rows and columns in the matrices [21].

6. Experimentation

In this section, we systematically evaluate the performance of our joint learning for user age prediction with both the classification and regression models.

Experimental Settings

Data Settings: The data collection has been introduced in Section 3.1. We extract a balanced data set from the collected data by selecting 200 samples in each age and the age is limited in the range of 19 to 28, totally 10 age categories. We use 80% of the data in each age category as the training data and the remaining 20% data as the test data. We also set aside 10% users from the training as the validation data which are used to tune learning algorithm parameters.

Representations: The features employed in our experiment are separated into three groups, i.e., textual features, social features and joint features, where joint features mean the mixture of textual features and social features. To get the word, POS features, we use the public toolkit ICTCLAS¹ to perform word segmentation, POS tagging on the Chinese text. Each user is represented by a bag-of-feature model [22].

Basic Prediction Algorithms: (1) Maximum entropy (ME), one popular shallow-learning algorithm, is implemented with the Mallet Toolkit². (2) Support vector machine (SVM), another popular shallow-learning algorithm, is implemented with the libSVM³ Toolkit. What's more, we implement SVM regression algorithm with the linear kernel, namely SVR for age regression. (3) MLP, includes fully-connected layer, activation layer and dropout layer which is implemented with the tool Keras⁴. (4) CNN, a popular deep-learning algorithm is used for performance comparison. We implement classifier and regressor via CNN respectively and the parameter setting follows the work by Johnson and Zhang [20]. (5) LSTM, as the basic prediction algorithm in our approach, is implemented with the tool Keras⁴. It is used in both the classifier and regressor for age prediction.

Parameters Setting: (1) The parameters of SVM, ME and SVR are set defaults. (2) The hyper parameters of LSTM are well tuned on the validation data by the grid search method, and most important hyper parameters are shown in Table 2. The hyper parameters of MLP is the same to LSTM.

Table 2. Parameters setting in LSTM

Parameter description	Value
Size of features	30000
Dimension of the LSTM layer output	128
Dimension of the full-connected layer output	64

¹ <http://www.ictclas.org/ictclas/download.aspx>

² <http://mallet.cs.umass.edu/>

³ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

⁴ <https://github.com/fchollet/keras>

Learning rate	0.01
Dropout probability for regression	0.25
Dropout probability for classification	0.5
Epochs of iteration	15

Evaluation Metric: We employ the coefficient of determination R^2 to measure the regression and classification performance. Coefficient of determination R^2 is used in the context of statistical models with the main purpose to predict the future outcomes on the basis of other related information. R^2 is a number between 0 and 1. R^2 nearing 1.0 indicates that a regression line fits the data well [23].

Significance Test: T-test is used to evaluate the significance of the performance difference between two approaches [24].

6.1 Experimental results on individual age classification and regression

For thorough comparison, we implement several approaches to age prediction for both the classification and regression tasks. These approaches are introduced as follows.

Age classification approaches:

SVM: The support vector machine classifier with all the parameters default.

ME: The maximum entropy classifier with all the parameters default.

C_CNN: The basic bow-CNN proposed in [20].

C_LSTM: The LSTM classification model which is described in Section 4.2.

Age regression approaches:

SVR: The support vector machine regressor with all the parameters default.

MLP: The standard MLP model for regression is implemented with the tool Keras⁴.

R_CNN: the overall architecture is similar to C_CNN, one exception is the output layer is linear layer.

R_LSTM: The LSTM regression model which is described in Section 4.3.

Table 3. The results of different age classification approaches with different feature groups

Classification method	Textual features	Social features	Joint features
SVM	0.251	0.250	0.334
ME	0.285	0.304	0.389
C_CNN	0.265	0.270	0.361
C_LSTM	0.326	0.348	0.395

Table 4. The results of different age regression approaches with different feature groups

Regression method	Textual features	Social features	Joint features
SVR	0.376	0.351	0.489
MLP	0.415	0.395	0.514
R_CNN	0.403	0.334	0.475

R_LSTM	0.468	0.416	0.553
--------	--------------	--------------	--------------

Table 3 shows the results of the four age classification approaches with three groups of features. From the table, we can see that C_CNN performs better than SVM, but worse than ME no matter what features are explored. Among all these approaches, C_LSTM performs best. Compared to SVM, ME and C_CNN, C_LSTM achieves the improvements of 7.5%, 4.1% and 6.1% with textual features, 9.8%, 4.4% and 7.8% with social features, and 6.1%, 0.6% and 3.4% with joint features. These results imply that the LSTM model is more appropriate for the age classification task than the other models. Significance test with t-test shows that our approach C_LSTM significantly outperforms the other approaches (p-value <0.05).

Table 4 shows the results of the four age regression approaches with three groups of features. From the table, we can see R_CNN performs rather poorly when social and joint features are employed. Among all these approaches, R_LSTM performs best. Compared to SVR, MLP and R_CNN, R_LSTM achieves the improvements of 9.2%, 5.3% and 6.5% with textual features, 6.5%, 2.1% and 8.2% with social features; 6.4%, 3.9%, and 7.8% with joint features. These results imply that LSTM is also more appropriate for the age regression task than the other models. Significance test with t-test shows that our approach R_LSTM significantly outperforms the other approaches (p-value <0.05).

Overall speaking, our empirical studies demonstrate that the LSTM model is a good choice to be the basic classification or regression algorithm in age prediction.

6.2 Experimental results of Joint Learning

For comparison, we implement following two approaches to joint learning on age prediction:

Hybrid model: This approach focuses on combining the result of age classifier and regressor linearly. Specifically, this approach can be divided into two stages, in the first stage, we train age classifier and regressor respectively based on the same features. In the second stage, we simply combine the results from the classifier and regressor by averaging them. For instance, if the result of the classifier is 18 years old and the result of the regressor is 20 years old, the combining result is the average of them, i.e., $(18+20)/2=19$ years old. This approach is a straightforward strategy to take advantage of classification and regression model simultaneously.

Aux-LSTM: This is our approach which learns an auxiliary representation for joint learning, which is also described in section 4.4 in detail. In our Aux-LSTM model, λ is set to 0.75.

Table 5. The results of our joint learning approach to age prediction with different feature groups

Approaches	Textual features	Social features	Merge features
C_LSTM	0.326	0.348	0.395
R_LSTM	0.468	0.416	0.553
Hybrid Model	0.423	0.419	0.509
Aux-LSTM	0.481	0.437	0.573

Table 5 shows the performance comparison of four approaches to age prediction where C_LSTM and R_LSTM are two baseline approaches which are introduced in the above subsection and Hybrid model and Aux-LSTM are two joint learning approaches. From the table, we can see that, Hybrid model performs even worse than R_LSTM when textual features and joint features are employed, and it gets a slight improvement of 0.3% over R_LSTM with social features. Among all these approaches, Our Aux-LSTM performs best, achieving 0.481, 0.437 and 0.573 performance in R^2 when textual features, social features and joint features are employed respectively. Compared to R_LSTM, Aux-LSTM results in an improvement of 1.3% with textual features, 2.1% with social features and 2.0%

with joint features. The experimental results demonstrate that our joint learning is consistently effective for performance improvement of age prediction. Significance test with t-test shows the improvement of our approach over Hybrid model is significant (p -value < 0.05).

7. Conclusion

In this paper, a novel approach is proposed, namely joint learning for age prediction, to exploit advantages of both the age classification and regression models. In our approach, an auxiliary LSTM layer is employed to learn the auxiliary representation in the age classification task (as the auxiliary task) and employ it in the age regression task (as the main task). To achieve this, a neural network based model, namely Aux-LSTM, is employed to bridge across the classification and regression models via a shared LSTM layer. Empirical studies demonstrate that the LSTM model is appropriate for both the age classification and regression task. Moreover, the results show that our joint learning approach significantly boost the performance of the main regression task with the help of the auxiliary classification task.

In our future work, we would like to seek better modification on Aux-LSTM for further improvement. Moreover, we would like to apply our proposed Aux-LSTM model in other NLP applications which involves both the classification and regression tasks.

References

- [1] D. Preotiuc-Pietro, V. Lampos, N. Aletras. *An analysis of the user occupational class through twitter content*. ACL, 1754–1764 (2015).
- [2] J. Wang, Y.Q. Zhu, D.Y. Li, et al. Joint User Attributes and Item Category in Factor Models for Rating Prediction. DASFAA, 277-296 (2016).
- [3] S. Volkova, T. Wilson, D. Yarowsky. *Exploring demographic language variations to improve multilingual sentiment analysis in social media*. EMNLP, 1815–1827 (2013).
- [4] B. O'Connor, R. Balasubramanyan, B.R. Routledge, et al. *From tweets to polls: Linking text sentiment to public opinion time series*. ICWSM, 1842-1850 (2010).
- [5] C. Peersman, W. Daelemans, L.V. Vaerenbergh. *Predicting Age and Gender in Online Social Networks*. SMUC, 37-44 (2011).
- [6] D. Nguyen, N.A. Smith, C.P. Rose. *Author Age Prediction from Text using Liner Regression*. ACL, 115-123 (2011).
- [7] D. Nguyen, R. Gravel, D. Trieschnigg, et al. *"How Old Do You Think I Am?": A Study of Language and Age in Twitter*. AAAI, 439-448 (2013).
- [8] J. Chen, S.S. Li, B. Dai, et al. *Active Learning for Age Regression in Social Media*. Springer International Publishing, 351-362(2016).
- [9] J. Chen, S.S. Li, J.J. Wang, et al. *User age prediction by combining classification and regression*. Sci Sin Inform, 47: 1095–1108(2017).
- [10] J.D. Burger, J.C. Henderson. *An Exploration of Observable Features Related to Blogger Age*. AAAI, 15-20 (2006).
- [11] J. Schler, M. Koppel, S. Argamon, et al. *Effects of Age and Gender on Blogging*. AAAI, 199-205 (2006).
- [12] I. Mackinnon, R. Warren. *Age and Geographic Inferences of the LiveJournal Social Network*. ICML, 176-178 (2006).
- [13] D. Ikeda, H. Takamura, M. Okumura. *Semi-Supervised Learning for Blog Classification*. AAAI, 1156-1164 (2008).
- [14] S. Rosenthal, K. McKeown. *Age Prediction in Blogs: A Study of Style, Content, and Online Behavior in Pre- and Post-Social Media Generations*. ACL, 763-772 (2011).
- [15] S. Goswami, S. Sarkar, M. Rustagi. *Stylometric Analysis of Bloggers' Age and Gender*. AAAI, 214-217 (2009).
- [16] J. Marquardt, G. Farnadi, G. Vasudevan, et al. *Age and gender identification in social media*. CLEF, 1129-1136 (2014).

- [17] S. Hochreiter, J. Schmidhuber. *Long Short-Term Memory*. NC, 9(8):1735–1780 (1997).
- [18] A. Graves. *Generating Sequences with Recurrent Neural Networks*. (2013).
- [19] G.E. Hinton, N. Srivastava, A. Krizhevsky, et al. *Improving Neural Networks by Preventing Co-adaptation of Feature Detectors*. Research Gate, 212-223 (2012).
- [20] T. Tieleman, G. Hinton. *RMSProp: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural Networks for Machine Learning (2012).
- [21] X. Glorot, Y. Bengio. *Understanding the difficulty of training deep feedforward neural networks*. AAAI, 249–256 (2010).
- [22] R. Johnson, T. Zhang. *Effective use of word order for text categorization with convolutional neural networks*. NAACL, 103–112 (2015).
- [23] A. Cameron, F. Windmeijer. *R-squared Measures for Count Data Regression Models with Applications to Health-care Utilization*. Journal of Business and Economic Statistics, 14(2): 209-220 (1993).
- [24] Y.M. Yang, X. Liu. *A Re-Examination of Text Categorization Methods*. SIGIR, 42-49 (1999).