This article can be cited as X.-W. Mu, X.-P. Shen and J. Kirby, Support Vector Machine Classifier Based on Approximate Entropy Metric for Chatbot Text-based Communication, International Journal of Artificial Intelligence, vol. 15, no. 2, pp. 1-16, 2017.

Support Vector Machine Classifier Based on Approximate Entropy Metric for Chatbot Text-based Communication

Xuewen Mu[1], Xiaoping Shen[2] and John Kirby[2]
[1] School of Mathematics and Statistics, Xidian University
Xi'an, Shaaxi 710071, China
Email: xdmuxuewen@hotmail.com]
[2] Department of Mathematics, Ohio University
Athens, OH 45701, USA
Email: shenx@ohio.edu

Abstract

Chatbot is a computer program designed to simulate conversation with human users over the Internet. Chatbot has been found on a number of chat systems, including large commercial chat networks. However, their use as malicious tools has made them a growing nuisance and security concern. We present a support vector machine training algorithm for classification on human and bots in chatbot text-based communications. We use data from the annual Loebner competition to distinguish between bots and humans. The normalized approximate entropy of Message size and inter-message delays at each conversation are introduced. Coupled with the mean and the normalized Shannon entropy of two features, they were considered as the input data. Simulation results have shown that the support vector machine is an efficient method for chatbot data classification.

Keywords: Chatbot, support vector machine, Approximate entropy, Shannon entropy, text-based communications, Loebner competition

Mathematics Subject Classification: 94A17, 93C41, 68Q32, 68T05, 68T40.

Computing Classification System: I.2.0, I.2.4 and J.0

1. INTRODUCTION

Internet chat is a popular application that enables real-time text-based communication. Millions of people around the world use Internet chat to exchange messages and discuss a broad range of topics online. Internet chat is also a unique networked application because of its human-to-human interaction and low bandwidth consumption [5]. However, the large user base and open nature of Internet chat make it an ideal target for malicious exploitations [10].

Chatbot is a computer program designed to simulate conversation with human users, especially over the Internet. They have been found on a number of chat systems, including large commercial chat networks, such as AOL Instant Messenger [9], Yahoo! [19], and MSN Messenger [11]. Unfortunately, the abuse of chat services by automated programs is becoming a serious network security issue. Chatbots exploit these online systems to send spam, spread malware, and mount phishing attacks [32] and [10]. Furthermore, more threats involve chatbots to steal personal information from on-line users [21] and [18].

In Section 2, we review the previous work in Chatbot data classification. Section 3 addresses Support Vector Machine Training Algorithm. Section 4 presents experimental result of chatbot data sets classification using SVM with three kernel functions. In Section 5, we summarize this work and introduce some references to supplement whatever important aspects the authors have indeed hardly touched upon in this paper.

2. RELATED WORK

Some efficient and reliable methods must be proposed to eliminate the threat from chatbots. Keyword-based filtering and human interactive proofs are two different methods to combat chatbots. In paper [3], the authors use human interactive proofs, such as CAPTCHAs, to stop chatbots from entering chat rooms, but, it is not effective. Keyword based filtering and related spam detection methods are common but seem to be having limited success with chatbots [17]. In paper [10], Based on the measurement study, the authors propose a classification system to accurately distinguish chatbots from humans, which consists of an entropy-based classifier and a Bayesian-based classifier. The experimental shows that the proposed classification system is highly effective in differentiating bots from humans. However, the chat data they analyzed was limited to public chat rooms with many multiple simultaneous communicators, and was primarily focused on bots that were attempting to get users to click on hyperlinks [18]. In paper [17], the authors use a different smaller data set to test the gross behavioral metrics, and find that they could be useful for passively distinguishing between humans and bots.

In paper [17], a more detailed graphical and statistical analysis of related measures is presented using a similar but larger data set. The normalized entropy and normalized entropy rate are introduced to measure the messaging complexity of human and chatbot. Entropy was calculated using Shannon’s information entropy approach [28]. Approximate Entropy (ApEn) is a statistical measure that quantifies the complexity in a signal. It is useful in classifying complex systems [2] and quantifies the unpredictability (randomness) in a time series data set. Approximate entropy can obtain more stable using shorter time-series However, ApEn is a biased statistic [2] and [27]. It lacks relative consistency and the result shows much dependence on data length. Here, we propose the metric, which is normalized approximate entropy, to analyze the messaging complexity of human and chatbot. Based on the two metric, we apply the support vector machine (SVM) method to distingush human to chatbot.

Support vector machine is a promising supervised machine learning algorithm for data classification and regression [4] and [7]. Furthermore, SVM also has achieved excellent generalization performance in a wide variety of applications, such as handwritten digit recognition [22] and [6], categorization of Web pages [1], and face detection [33]. In addition, SVMs are also useful in medical science to classify proteins with up to 90% of the compounds classified correctly. Now, the SVM training algorithms are designed to to deal large data sets [8] and [29]-[31]. The data sets from Chatbot are available in some websites, such as publicly-available transcripts of the Loebner Prize [20]. These data resources can provide large data sets for us to test the performance of the classification algorithms for chatbots.

In the paper, we present a support vector machine training algorithm for classification on humans and bots in chatbot text-based communications. We use data from the annual Loebner competition to distinguish between chatbots and humans. Message size and inter-message delays were considered as the input data.Simulation results have shown that the support vector machine is an efficient method for distinguishing humans from bots.

3. NORMALIZED APPROXIMATE ENTROPY FOR THE CHATBOT DATA SETS

In this section, we will give the analysis of the chatbot data sets, sepecially we will use the Approximate Entropy Rate as a metric to chatbot data sets.

3.1 About the chatebots data

The raw chatbots data sets are are gathered from the publicly-available transcripts of the Loebner Prize in Artificial Intelligence [20] which is also used in the papers by John McIntire et al [16] and [17]. The Loebner Prize is a formal public Turing Test competition in which the worlds best chatbots programs compete to be the most human in terms of conversational capabilities. In paper [17], the authors explained three reasons to use the Loebner Prize data sets. The first reason is that the communicators are all unambiguously defined as either humans or chatbots. Another reason is that a large proportion of the conversations involve chatbots, whereas other public chat data sets do not typically possess such high frequency of bot communications. In addition, the bots are specifically trying to carry on human-like conversations with people for lengthy periods, whereas in many public data sets bots are mostly trying to convince users to click hyperlinks posted in their messages or in their online profiles.

From the data sets in [20], we analyzed five separate competitions, from the years 1996, 1997, 2004, 2005, and 2008. These data sets include a total of 9206 messages in 254 brief conversations involving over 50 individual humans and 22 chatbots. The research results in paper [17] show that there are two marked features for the chatbots data sets, which are IMDelay and wordcount. IMDelay is the inter-message delay and wordcount is the number of words in the message. The value of IMDelay is equal to the amount of time (in seconds) for the participant to respond. The value of Wordcout is the length of the message sent.

We should preprocess the data sets before they are inputed into the SVM algorithm, and extract some metric for the data sets. Firstly, we compute two metric values for the 9206 messages, which are IMDelay and wordcount. In addition, a class label must be assigned to each data point. For the chatbot data, a class label of 1 was given to a computer participant and a class label of −1 was given to a human participant. In addition, the value of IMDelay for the start of each conversation does not exist. Hence a value must been inserted in each of these slots. Several values can be inputted into these slots such as zero, the average, or the median of the data. Here, the values of these entries are replaced with the average values of the same talker in a conversation.

At each conversation, we gather all the data points in chronological orderfor the same human or chatbot. The time series are obtained for the two metric IMDelay and wordcount. Then we obtain 508 time series of IMDelay and wordcount base on the 254 brief conversations from the 9206 message points respectively.

3.2 Calculate Metrics

From the 508 time series of IMDelay and wordcount, we compute three values, including arithmetic mean, the normalized Shannon entropy and normalized ApEn of the time series.

Given the time series of IMDelay or wordcount for each conversation is $\{u_{1j,} u_{2j,} \cdots u_{kj,}\}$, where k is the length of the time series at the conversation. In paper [17], normalized Shannon entropy is used to measure messaging complexity. Entropy was measured by first calculating the empirical probability $p(u_{ij})$ of occurrence of each IMDelay and wordcount. The probabilities for each category were then used in the Shannon entropy equation:

$$H_j = \sum_i p(u_{ij}) \times \log(p(u_{ij})),\ j = 1,2,\cdots,508 \tag{1}$$

Pincus [24] presented approximate entropy (ApEn) as a measure of complexity that is applicable to noisy, medium-sized datasets. ApEn is applicable to noisy, medium-sized datasets. Approximate entropy can obtain more stable using shorter time-series data. Here, we use ApEn as a measure of complexity of chatbot data sets at each conversation. Based on papers [24] and [25], ApEn can be calculated following the algorithm below:

Step 1: Form a time series of data $\{u_{1j,} u_{2j,} \cdots u_{kj,}\}$, where k is the length of time series.

Step 2: Assume integer l represent the length of compared run of data, and positive real number r repesent a filtering level.

Step 3: Form a sequence of vectors $v_{1j,} v_{2j,} \cdots v_{(k-l+1)j}$ in R^l, which is real l dimensional sapce defined by

$$v_{ij} = \{u_{ij}, u_{(i+1)j}, \cdots, u_{(i+l-1)j}\} \tag{2}$$

Step 4: Use the sequence $v_{1j,} v_{2j,} \cdots v_{(k-l+1)j}$ to construct, for each $i, 1 \le i \le k-l+1$

$$C_i^l(r) = \frac{number\ of\ v_{hj}\ such\ that\ d[v_{ij} - v_{hj}] < r}{k-l+1}, \tag{3}$$

where $d[v_{ij} - v_{hj}]$ is defined as

$$d[v_{ij} - v_{hj}] = \max_{q=0,\cdots,l-1} \left| u_{(i+q)j} - u_{(h+q)j} \right|. \tag{4}$$

The signal d represents the distance between the vectors v_{ij}, v_{hj} ,given by the maximum difference in their the maximum difference in their respective scalar components.

Step 5: Define

$$\Phi^l(r) = (k-l+1)^{-1} \sum_{i=1}^{k-l+1} \log(C_i^l(r)). \tag{5}$$

Step 6: Define approximate entropy as

$$ApEn = \Phi^l(r) - \Phi^{l+1}(r), \tag{6}$$

where $\log$ is the natural logarithm.

However, ApEn is a biased statistic [24] and [25], and it lacks relative consistency and the result shows much dependence on data length. Here, we propose the metric, which are approximate entropy rate, to analyze the messaging complexity of human or chatbot at each conversation. Normalization was accomplished by taking the observed entropy divided by the maximum possible entropy given the sample size k :

$$NApEn = ApEn / \log(k). \tag{7}$$

These normalized ApEn provided measures of complexity that could more be easily compared across the varying the sample sizes used in the analysis.

3.3 Support vector machine training algorithm

Given a training set $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where each point x_i belong to R^m and y_i is a label that identifies that the class of point x_i. Our goal is to determine a function

$$f(x) = w^T \phi(x_i) + b \quad , \tag{8}$$

where w is the normal vector to the hyperplane, b is a real number, and $\phi(x_i)$ is a mapping from R^m to a higher dimensional space, such as R^n.

The idea of the learning program SVM is to find a hyperplane such that the distance between the closest vectors (called the margin) from each class is maximized. This is done by solving the following optimization problem [13]:

$$\begin{aligned} \min \quad & \frac{1}{2} w^T w \\ s.t. \quad & y_i (w^T \phi(x_i) - b) \geq 1, i = 1,2,\cdots,n \end{aligned} \tag{9}$$

The hyperplane obtained from the optimization problem is Hard-Margin SVM. However, it is not always possible to find a hyperplane such that the data is completely separated; that is, every data point of class 1 is on one side of the hyperplane and every data point of class 2 is on the other side of the hyperplane. Therefore the optimization problem must be modified to allow some misclassifications. The Soft-Margin SVM [14] and [15] can be obtained by solving the following modified optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2} w^T w + \frac{1}{2} \lambda \sum_{i=1}^{n} \xi_i^2 \\ s.t. \quad & y_i (w^T \phi(x_i) - b) \geq 1 - \xi_i, i = 1,2,\cdots,n \end{aligned} \tag{10}$$

where ξ_i are error variables and λ is the regularization parameter

Instead of calculating the mapping ϕ, support vector machines using a function called the kernel function. The three standard kernel functions are linear, polynomial, and radial basis function (RBF). These are given by:

Linear kernel: $K(x_i, x_j) = x_i^T x_j$

Polynomial kernel: $K(x_i, x_j) = (x_i^T x_j + c)^d$, where c, d are parameters

RBF kernel: $K(x_i, x_j) = \exp(\frac{\left\| x_i - x_j \right\|^2}{\sigma^2})$, where σ is parameter

There are some efficient algorithms to the L2-SVM problems. Based on the dual programming of problem (9) or (10), a primal-dual interior-point method is proposed for solving the SVM problems [12]. Keerthi and DeCoste in paper [14] propose modified Newton methods. A trust region Newton method [15] is proposed for logistic regression L2-SVM. A Dual Coordinate Descent Method for Large-scale

Linear SVM is proposed in paper [1].

4. EXPERIMENTAL CLASSIFICATION RESULTS AND ANALYSIS

In this paper, the SVM with kernel functon is calculated by the software LS-SVMlab version 1.7 [3]. LS-SVMlab is a toolbox for the computing software Matlab. For details of the software and its application, we refer the reader to the software website, http://www.esat. kuleuven.be/sista/lssvmlab. The graphical representation of the LS-SVMlab toolbox was slightly modified to convey the chatbot data. All the algorithms are run in the MATLAB 7.0 environment on a Inter Core2 D2.0GHz personal computer with 2.0 GB of RAM.

4.1 The analysis and preprocessing of chatbots data sets

We modify the data sets before they are inputed into the SVM algorithm. For each point in the 508 sample points, four values are computed, including arithmetic mean of IMDelay, arithmetic mean of wordcount, the normalized Shannon entropy or normalized ApEn of IMDelay, and the normalized Shannon entropy or normalized ApEn of wordcount. For the chatbot data, a class label of 1 was given to a computer participant and a class label of −1 was given to a human participant.

The number of conversations and the number of sample points at each year are shown in Table 1. In addition, we also give the ratio between number of conversation and number of sample points. The ratio denotes the average number of message at each conversation in the same year. Table 1 shows that the largest average number of message is in 2008 year among the five years. In an internet conversation, smaller number of messages mean more difficult to distinguish human from robot. We pick up the data sets in 2008 to test the accuracy results with the sample points at the same year. The results will be shown in Table 2. Table 1 also shows that the average number of sample point in 2004 is smaller than that in the other years except 1996. For the normalize Shannon Entropy and normalized ApEn, the longer the time series is, the more efficient the entropy value is. We pick up the data sets in 2004 to test the accuracy results with the sample points at the same year. The results will be shown in Table 3.

Table 1. Data Sets at each year

Year	Number of conversations	Number of Message	Ratio
1996	30	3125	104.1
1997	50	1496	29.9
2004	24	1302	54.3
2005	29	1124	38.8

2008	121	2259	18.7

4.2 Parameter design

As is known to all, parameter estimation in kernel function for SVM is very important. For example, parameter σ is very important for RBF kernel, and degree parameter d is also crucial to polynomial kernel. Appropriate parameters mean higher accuracy for SVM.

LS-SVMlab software provide some cross validation methods to estimate and tune the parameters, such as leave-one-out cross validation, L-fold cross validation, generalized cross validation, etc. However, these methods cost a considerable amount of CPU time for tuning the parameters. So we obtain the parameter based on an exhaustive search in a limited range. We obtain parameter σ by an exhaustive search in a range from 0.1 to 2.5 with searching interval 0.1. The degree parameter d is generated by exhaustive search in a range from 1 to 10. We select sample points from 5, 10, 15 and 20 conversations in 2008 as the training data with two values are selected, only including normalized ApEn of IMDelay and wordcount, and use SVM with polynomial kernel to test the variety of accuracy performance with parameter d . The simulation result is shown in Figure 1. Figure 1 shows the best degree for the polynomial kernel is two to our test data sets. The better degree for the polynomial kernel mainly focuses on low degree from 1 to 4, that is to say $d \in \{1,2,3,4\}$ is the better choice.

Although the exhaustive search in a limited range is simple, it can obtain a relative better parameter with lower CPU time cost.

For Normalized ApEn, we select parameter with $m \in \{1, 2\}, r = 0.25$. The destination of simulation is to distinguish the robot from human in the conversations, so we use the number of conversations to determine the training and testing sample points. The random sample points from different number of conversations are generated as the training sets and the remaining sample points besides the training sample points are generated as the testing sets. In addition, the data sets are divided into 5 data sets based on the different year.

In the first simulation example, we select different kinds of number of conversations as the training sample points to test accuracy of remaining data based on the two different kernel functions at the same year. We only choose the data sets from 2008 and 2004. We do 50 random simulations, and obtain the average accuracy and CPU time performances. The results for the LS-SVM using the random sample with the polynomial, and RBF kernels are given in Table 2 and Table 3, respectively.

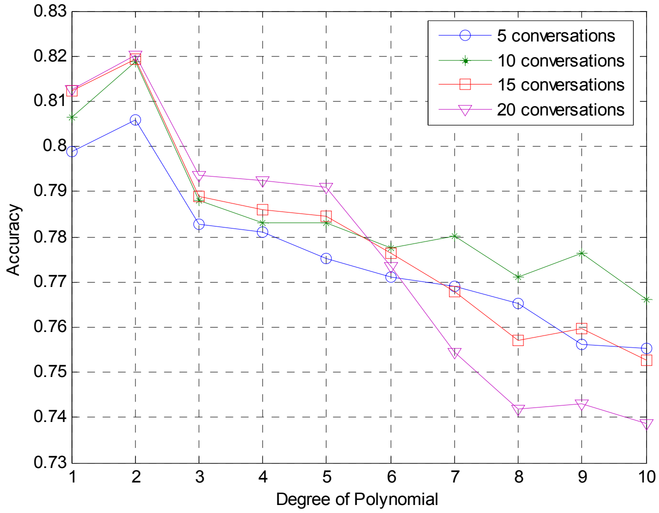

Figure 1. The variety of accuracy with the parameter *d*

4.3 Simulation results

Table 2. The simulation results in 2008 year

Number	Polynomial Kernel			RBF Kernel		
	Mean	Normalized Entropy	Normalized ApEn	Mean	Normalized Entropy	Normalized ApEn
5	78.98%	76.34%	73.60%	81.63%	80.71%	80.17%
10	80.75%	79.86%	77.13%	82.10%	81.19%	80.65%
15	81.27%	81.60%	80.23%	83.30%	81.83%	81.03%
20	81.94%	83.41%	81.43%	84.17%	82.96%	81.25%
25	83.02%	85.54%	82.80%	84.72%	83.32%	81.81%
30	83.90%	85.89%	83.72%	85.54%	83.88%	82.42%
35	84.26%	86.40%	84.40%	85.26%	84.16%	83.10%
40	84.43%	86.18%	84.92%	84.96%	84.03%	83.37%
45	84.88%	86.56%	85.51%	84.41%	83.80%	83.36%
50	85.02%	87.00%	86.32%	83.84%	83.81%	83.74%
55	84.76%	86.77%	86.24%	83.73%	84.01%	83.87%
60	84.55%	86.56%	86.47%	84.03%	84.06%	84.22%

In Tables 2-4, “Number” denotes the number of conversations for training data sets, “mean” presents the accuracy to distinguish human from robot only using the arithmetic mean of IMDelay and wordcount, “Normalized Entropy” means the accuracy using four values, including the arithmetic mean and normalized Shannon entropy of IMDelay and wordcount, and “Normalized ApEn” denotes the accuracy using four values, including the arithmetic mean and normalized ApEn of IMDelay and wordcount. The results in Table 2 show the following conclusions: Firstly, the more sample points can obtain higher accuracy. But for the polynomial kernel based on “Mean” and “Normalized Entropy” data sets, when the number of conversations exceeds 50, the overfitting phenomenon occurs. For the RBF kernel based on “Mean” and “Normalized Entropy” data sets, when the number of conversations exceeds 30 and 35, the overfitting phenomenon occurs. Secondly, the accuracy of the SVM based on the “Normalized Entropy” data set is higher than that of the SVM based on “Normalized ApEn”.

Table 3. The simulation results in 2004 year

Number	Polynomial Kernel			RBF Kernel		
	Mean	Normalized Entropy	Normalized ApEn	Mean	Normalized Entropy	Normalized ApEn
2	69.37%	73.23%	68.69%	77.68%	76.71%	76.13%
4	81.88%	84.43%	81.18%	86.29%	86.52%	89.08%
6	87.90%	86.48%	88.47%	90.61%	91.46%	95.02%
8	90.73%	90.55%	89.83%	91.62%	92.69%	96.96%
10	90.80%	91.03%	92.64%	91.72%	92.87%	97.24%
12	92.00%	91.07%	93.84%	92.00%	95.07%	97.84%
14	91.34%	93.07%	93.07%	91.34%	94.80%	98.70%
16	94.11%	94.11%	96.07%	92.81%	96.73%	99.34%
18	93.40%	96.70%	97.80%	98.90%	98.90%	100%
20	93.33%	97.77%	97.77%	100%	100%	100%

The results in Table 3 show the following conclusions: Firstly, the more sample points used, the higher accuracy we can obtain. But for the polynomial kernel based on “Mean” data sets, when the number of conversations exceeds 16, the overfitting phenomenon occurs. Secondly, the accuracy of the SVM based on the “Normalized ApEn” data set is higher than that of the SVM based on “Normalized Entropy”.

Coupled with the results in Table 1, the results in Tables 2 and 3 show the normalized ApEn is a more efficient metric when the length of time series is longer. So, the normalized ApEn is a good metric for chatbot data sets to distinguish human from bots.

At the same time, we select 30 conversations as the training samples to predict the remaining data based on the polynomial kernel and RBF kernel in 2008. Two kinds of different sample data sets with two values are selected, only including normalized Shannon Entropy or normalized ApEn of IMDelay and wordcount. The results are shown in Figures 2-5.Firgure 2 and Firgure 5 show the training results based on normalized ApEn of IMDelay and wordcount. Firgure 3-4 show the training results based on normalized Shannon Entropy.

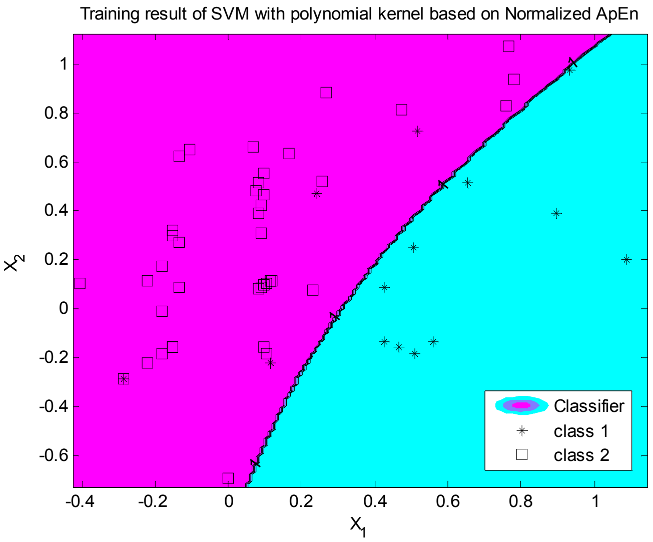

Figure 2: Training result of SVM with polynomial kernel based on Normalized ApEn

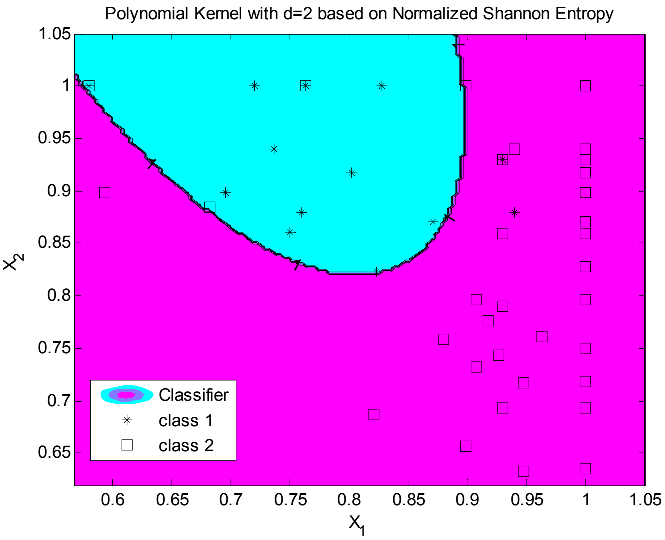

Figure 3: Training result of SVM with polynomial kernel based on Normalized Shannon Entropy

Figures 2 to 5 show the training results of SVM with RBF kernel function and polynomial kernel are efficient both based on normalized Shannon entropy and normalized ApEn data set in selected year. We also can find that the SVM is a good classifier for chatbot data sets.

In the second simulation example, we use the different data sets from different years to test the accuracy of the data sets in all the years. We select the training samples from 10 conversations, and we also use the RBF kernel function. We do 50 random simulations, and obtain the ave rage accuracy and CPU time performances. The results are shown in Table 4. In Table 4, “Max” represents the maximal accuracy among the results based on three kinds of sample points in the same year.

The results in Table 4 show that the accuracy predicted by the same year is higher than that by the different year except 2005 year. The minimal accuracy among the ”Max” rows is 67.23%, which use the train sample point in 2004 to classify the data in 1996.

From above analysis, LS-SVM method can obtain good classification results for chatbot data sets.

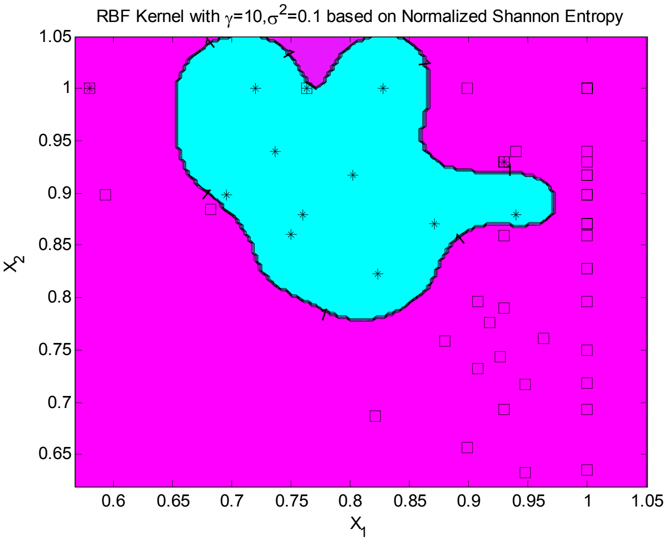

Figure 4: Training result of SVM with RBF kernel based on Normalized Shannon Entropy

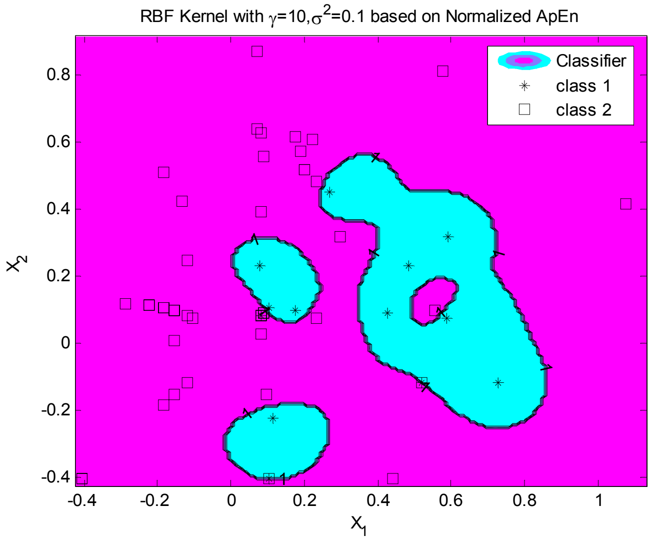

Figure 5: Training result of LS-SVM with RBF kernel based on Normalized ApEn

Table 4. Accuracy comparison among different year's data set

Year	Metric	1996	1997	2004	2005	2008
1996	Mean	88.39%	79.01%	78.25%	78.95%	87.14%
	Normalized Entropy	83.28%	76.43%	77.54%	77.19%	83.73%
	Normalized ApEn	82.46%	75.87%	78.49%	76.80%	80.04%
	Max	88.39%	79.01%	78.25%	78.95%	87.14%
1997	Mean	75.77%	77.70%	63.20%	68.01%	76.30%
	Normalized Entropy	76.17%	84.09%	77.72%	74.72%	80.11%
	Normalized ApEn	73.38%	80.95%	77.60%	73.89%	79.01%
	Max	76.17%	84.09%	77.72%	74.72%	80.11%
2004	Mean	63.16%	72.05%	91.72%	90.06%	59.78%
	Normalized Entropy	67.23%	74.81%	92.87%	88.89%	75.93%
	Normalized ApEn	66.21%	74.28%	97.24%	85.61%	76.21%
	Max	67.23%	74.81%	97.24%	90.06%	76.21%
2005	Mean	71.36%	79.80%	91.81%	92.05%	68.65%
	Normalized Entropy	73.39%	80.00%	92.87%	86.67%	75.68%
	Normalized ApEn	71.36%	80.96%	93.09%	91.67%	75.46%
	Max	73.39%	80.96%	93.09%	92.05%	75.68%
2008	Mean	77.70%	69.90%	75.08%	73.92%	82.10%
	Normalized Entropy	68.47%	66.91%	78.63%	72.40%	81.20%
	Normalized ApEn	67.65%	68.40%	81.75%	75.83%	80.66%
	Max	77.70%	69.90%	81.75%	75.83%	82.10%

5. CONCLUSION AND FURTHER STUDY

In this paper, two marked features and three metric are used for the chatbot data sets to classify human with bots in the chatbot text-based communications. Among the three metrics, normalized ApEn is introduced by us. Simulation results have shown that the support vector machine classifier with the new metric is an efficient and reliable classification method for chatbot data sets. We will continue working on this research in the future about following topics. Firstly, we will find some efficient kernel functions specifically for the chatbot data sets. Now we are testing multiscale kernels, such as wavelet kernel functions. An efficient kernel will improve the accuracy of classification. Secondly, the differences between the distributions for the IMDelay data for the computer and for the human data indicate that the IMDelay parameter is more sensitive than the Wordcount parameter. Since this is the case, IMDelay will be better than Wordcount in analysis. Our future study is to build a model for discrimination testing between human and machine. We will introduce weight functions so that the IMDelay parameter has more influence than the Wordcount parameter. Finally, we will extract other features from the raw chatbot data sets as the input data with IMDelay and Wordcount.

We also noticed that there are many other new methods based on optimization, system control theory and such have been developed in the related areas, for example, [16], [23], [26], and [29], to name a few. We will integrate these new methods to supplement whatever important aspects to our future study. It is also remarkable that the advatage of ApEn(m, r) is its multiscale property and robostics, which determined by the window size (or the embeding dimension) parameter m, and the thresholding parameter r. These two parameters are tunable and data dependent. The experiment results reported in this paper are based on fixed parameter ApEn(m, 0.25), m=1, 2. The selection rule for optimal parameters for natual languenge processing related problem is under our investigation with potential results will be reported in near future.

Acknowledgement

The first author was supported by a research grant from National Science Foundation for Young Scientists of China [Grant No.11101320]. The work is also supported by China Scholarship Council (CSC).

REFERENCES

Chang, K. W., Hsieh, C. J. and Lin, C. J., 2008, Coordinate descent method for large-scale L2-loss linear SVM, *Journal of Machine Learning Research*, **9**, 1369-1398.

Cristianini, N. and Taylor, J. S., 2000, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge.

De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., and Suykens, J. A. K., LS-SVMLab website, 2011, Available: http://www.esat.kuleuven.be/sista/lssvmlab.

DeCoste, D. and Scholkopf, B., 2002, Training invariant support vector machines, *Machine Learning*, **46** (1-3), 161-190.

Dewes, C., Wichmann, A., and Feldmann, A., 2003, An analysis of Internet chat systems, *Proceedings of 2003 ACM SIGCOMM Conference on Internet Measurement,* 51-56.

Dong, J.-X, Krzyzak, A. and Suen, C. Y., 2005, Fast SVM training algorithm with decomposition on very large data sets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27** (4), 603-618.

Dong, J. X., Suen, C. Y. and Krzyzak, A., 2003, A fast SVM training algorithm, *International Journal of Pattern Recognition and Artificial Intelligence*, **17** (3), 367-384.

Ferris, M. C. and Munson, T. S., 2003, Interior point methods for massive support vector machines, *SIAM Journal on Optimization*, **13** (3), 783-804.

Festa, P, 2010, AOL spam petitions cut both ways, *CNET News December. 22, 2010 [Online].* Available: http://news.cnet.com/AOL-spam-petitions-cut-both-ways/2100-1024_3-1015385.html.

Gianvecchio, S., Xie, M., Wu, Z., and Wang, H., 2011, Humans and bots in internet chat: Measurement, analysis, and automated classification, *IEEE/ACM Transactions on Networking*, **19** (5), 1557-1571.

Hinze-Hoare, V., 2004, Should cyberspace chat rooms be closed to protect children? *Comput. Res. Repository*.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundararajan, S., 2008, A dual coordinate descent method for large-scale linear SVM, *Proceedings of 25th International Conference on Machine Learning*, 408-415.

Joachims, T., 1998, Text categorization with support vector machine: Learning with many relevant features, *Proceedings of 10th European Conference on Machine Learning*, 137-142.

Keerthi, S. S. and DeCoste, D., 2005, A modified finite Newton method for fast solution of large scale linear SVMs, *Journal of Machine Learning Research*, **6**, 341-361.

Lin, C.-J., Weng, R. C., and Keerthi, S. S., 2008, Trust region Newton method for large-scale logistic regression, *Journal of Machine Learning Research*, **9**, 627-650.

Loebner Prize in Artificial Intelligence: The First Turing Test, annual competition website [Online], Available: http://www.loebner.net/Prizef/loebner-prize.html.

Martin, D., del Toro, R., Haber, R., and Dorronsoro, J., 2009, Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process, *International J of Innovative Computing, Information and Control*, **5** (10 (B)), 3405-3414.

McIntire, J., McIntire, L., and Havig, P., 2010, Methods for chatbot detection in distributed text-based communications, *Proceedings of 2010 International Collaborative Technologies and Systems Symposium*, Chicago, IL, USA.

McIntire, J., Havig, P., Farris, K. and McIntire, L., 2010, Graphical and statistical communication patterns of automated conversational agents in collaborative computer-mediated communication systems, *Proceedings of IEEE 2010 National Aerospace and Electronics Conference*, Fairborn, OH, USA.

Mohta, A, 2007, Yahoo Chat: CAPTCHA check to remove bots, Technospot.Net. Available: http://www.technospot.net/blogs/yahoo-chat-captcha-check-to-remove-bots/.

Naughton, P., 2007, Flirty chat-room bot out to steal your identity, FOXNews web article [Online]. Available: http://www.foxnews.com/story/2007/12/12/flirty-chat-room-bot-out-to-steal- your- identity/.

Osuna, E., Freund, R., and Girosi, F, 1997, Training support vector machines: an application to face detection, *Proceedings of 1997 IEEE. Conference on Computer Vision and Pattern Recognition*, 130-136.

Pincus S. M., 1991, Approximate entropy as a measure of system complexity, *Proceedings of the National Academy of Sciences of the United States of America*, **88** (6), 2297–301.

Pincus, S. M. and Goldberger A. L., 1994, Physiological time-series analysis: What does regularity quantify?, *American Journal of Physiology: Heart and Circulatory Physiology*, **266** (4 35-4): 1643–56.

Precup, R.-E., Dragos, C.-A., Preitl, S., Radac, M.-B., and Petriu, E. M., 2012, Novel tensor product models for automatic transmission system control, *IEEE Systems Journal*, **6** (3), 488-498.

Ramírez-Ortegón, M. A, Märgner, V., Cuevas, E., and Rojas, R., 2013, An optimization for binarization methods by removing binary artifacts, *Pattern Recognition Letters*, **34** (11), 1299-1306.

Scholkopf, B. and Smola, A. J., 2002, *Learning with Kernels*, MIT Press, Cambridge, MA, USA.

Shannon, C., 1948, A mathematical theory of communication, *Bell System Technical Journal*, **27**, 379-423, 623-656.

Solos, I. P., Tassopoulos, I. X., and Beligiannis, N., 2016, Optimizing shift scheduling for tank trucks using an effective stochastic variable neighbourhood approach, *International Journal of Artificial Intelligence*, **14** (1), 1-26.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J., 2002, *Least Squares Support Vector Machines*, World Scientific Pub. Co., Singapore.

Suykens, J. A. K. and Vandewalle, J., 1999, Least squares support vector machine classifiers, *Neural Processing Letters*, **9** (3), 293-300.

Thakur, S., 2013, AOL no more chat room spam petition, Petition Online, online public web petition, Available: http: //www.petitiononline.com/chatspam/petition.html.

Tsang, W., James, T., Kwok, T., Cheung, P.-M., and Cristianini, N., 2005, Core vector machines: Fast SVM training on very large data sets, *Journal of Machine Learning Research*, **6**, 363-392.

Vapnik, V. N/, 1998, *Statistical Learning Theory*, Wiley, New York.