Neural Comput & Applic (2013) 22:417–425
DOI 10.1007/s00521-012-0946-x

An extreme learning machine approach for speaker recognition

Yuan Lan · Zongjiang Hu · Yeng Chai Soh · Guang-Bin Huang

Received: 2 August 2011 / Accepted: 12 April 2012 / Published online: 19 May 2012

Abstract Over the last two decades, automatic speaker recognition has been an interesting and challenging problem to speech researchers. It can be classified into two different categories, speaker identification and speaker verification. In this paper, a new classifier, extreme learning machine, is examined on the text-independent speaker verification task and compared with SVM classifier. Extreme learning machine (ELM) classifiers have been proposed for generalized single hidden layer feedforward networks with a wide variety of hidden nodes. They are extremely fast in learning and perform well on many artificial and real regression and classification applications. The database used to evaluate the ELM and SVM classifiers is ELSDSR corpus, and the Mel-frequency Cepstral Coefficients were extracted and used as the input to the classifiers. Empirical studies have shown that ELM classifiers and its variants could perform better than SVM classifiers on the dataset provided with less training time.

Keywords Speaker verification · Extreme learning machine · Optimization method based extreme learning machine · Regularized extreme learning machine · Kernelized extreme learning machine · Support vector machine

Y. Lan · Z. Hu · Y. C. Soh (✉) · G.-B. Huang
School of Electrical and Electronic Engineering,
Nanyang Technological University, Nanyang Avenue,
Singapore 639798, Singapore
e-mail: eycsoh@ntu.edu.sg

Y. Lan
e-mail: lanyuan@ntu.edu.sg

Z. Hu
e-mail: hu0015ng@e.ntu.edu.sg

G.-B. Huang
e-mail: egbhuang@ntu.edu.sg

1 Introduction

Speaker recognition is the process of automatically recognizing who is speaking based on the speaker-specific information included in speech waves. Over the last two decades, automatic speaker recognition has been an interesting and challenging problem to speech researchers [1, 3, 5, 7, 9, 19, 21]. This technique can be used to verify people's identity and control access to services such as voice dialing, banking over a telephone network, telephone shopping, voice mail and remote access of computers.

Speaker recognition can be classified into two different categories: speaker identification and speaker verification. Speaker identification is a task of determining unknown speaker's identity. On the other hand, speaker verification is a task of accepting or rejecting the identity claim of a speaker. The key difference between identification and verification is the number of alternative decisions. In identification, the number of alternative decisions is equal to the size of population, whereas in verification, only two alternatives are available, accept or reject, regardless of the population size. In addition, speaker recognition can also be divided into text-dependent and text-independent recognition. The former requires the speaker to utter a prescribed text, which is then compared with some pre-stored pattern of the same text. In contrast, the later does not rely on any specific texts.

In this paper, a new classifier, extreme learning machine classifier, is introduced and evaluated on the text-independent speaker verification task and compared with SVM classifiers. Extreme learning machine (ELM) [15] proposed by Huang et al. is developed for generalized single hidden layer feedforward networks (SLFNs) with a wide variety of hidden nodes. Different from other learning algorithms, ELM randomly selects all the hidden node parameters, after which the network can be represented as a linear system and the output weights can be computed analytically. It tends to obtain the smallest training error and the smallest norm of weights that lead to good generalization performance. In addition, ELM has many variants including optimization method based ELM [14], regularized ELM [16] and kernelized ELM [16]. Optimization method based ELM extends preliminary ELM to support vector network by applying the standard optimization method with less optimization constraints, with which it becomes easier to implement support vector networks. Regularized ELM adds a positive value $\frac{1}{\lambda}$ in the calculation of output weights, so that the resultant solution could be stabler. Moreover, if the hidden layer feature mapping is unknown, kernelized ELM may come into play. ELM and its variants could provide good generalization performance for many regression and classification applications, and they may run much faster than SVM classifiers [13]. In this paper, we could like to apply ELM and its variants on the text-independent speaker verification task and compare them with SVM classifiers.

The rest of this paper is organized as follows. Section 2 provide brief reviews of the variants of ELM. A description of the application data, manipulation of the data, method of evaluation, the parameter selection of algorithms and the results are presented in Sect. 3. Finally, Sect. 4 is the conclusion of the paper.

2 A brief review of variants of extreme learning machine

2.1 Optimization method based extreme learning machine for classification

In ELM [11, 13], the input data are mapped from the input space to the L-dimensional hidden layer feature space (ELM feature space). The output of ELM is

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \mathrm{h}(\mathbf{x})\beta \tag{1}$$

where $\beta = [\beta_1, \ldots, \beta_L]^T$ is the output weight vector from hidden nodes to the output node. $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]$ is the row vector presenting the outputs of the L hidden nodes with respect to the input $\mathbf{x}$. In other words, $\mathbf{h}(\mathbf{x})$ maps the data from the d-dimensional input space to the L-dimensional hidden layer feature space H.

Given a set of training data $(\mathbf{x}_i, t_i), i = 1, \ldots, N$, where $\mathbf{x}_i \in \mathbf{R}^d$ and $t_i \in \{-1, 1\}$, according to ELM theory [12, 13], these data are linearly separable in ELM feature space with probability one. Instead of strictly zero training error, the training data are separated with an acceptable minimal training error to prevent from overfitting. Then, we have

$$\begin{cases} \beta \cdot \mathbf{h}(\mathbf{x}_i) \geq 1 - \xi_i & \textit{if} \quad t_i = 1 \\ \beta \cdot \mathbf{h}(\mathbf{x}_i) \leq -1 + \xi_i & \textit{if} \quad t_i = -1 \end{cases} \tag{2}$$

That is,

$$t_i \beta \cdot \mathbf{h}(\mathbf{x}_i) \geq 1 - \xi_i, \quad i = 1, \ldots, N \tag{3}$$

Different from the preliminary ELM, following the standard optimization theory and to minimize both training error and output weights, we have

$$\begin{aligned} &\text{Minimize:} && \mathbf{L}_P = \tfrac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N} \xi_i \\ &\text{Subject to:} && t_i\beta \cdot \mathbf{h}(\mathbf{x}_i) \geq 1 - \xi_i, \quad i = 1, \ldots, N \\ &&& \xi_i \geq 0, \quad i = 1, \ldots, N, \end{aligned} \tag{4}$$

where C is a user specified parameter and it provides a trade-off between the training error and norm of output weights. Equation (4) is different from conventional SVM's optimization problem, in which all the parameters of $\mathbf{h}(\mathbf{x})$ are chosen randomly. In addition, the separating hyperplane in ELM feature passes through the origin, so the network bias in SVM is not required in the ELM's optimization constrains.

We define an optimization method based ELM kernel function as

$$\begin{aligned} \mathbf{K}_{ELM} &= \mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j) \\ &= [G(\mathbf{a}_1, b_1, \mathbf{x}_i), \ldots, G(\mathbf{a}_L, b_L, \mathbf{x}_i)]^T \cdot \\ &\quad \times [G(\mathbf{a}_1, b_1, \mathbf{x}_j), \ldots, G(\mathbf{a}_L, b_L, \mathbf{x}_i)]^T \end{aligned} \tag{5}$$

where $G(\cdot)$ could be any piecewise continuous function satisfying the ELM universal approximation capability theorems [10, 12] and $\{(\mathbf{a}_i, b_i)\}_{i=1}^{L}$ are randomly generated according to any continuous probability distribution.

Thus, we obtain

$$\begin{aligned} &\text{Minimize:} && \mathbf{L}_D = \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} t_i t_j \mathbf{K}_{ELM}(\mathbf{x}_i, \mathbf{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i \\ &\text{Subject to:} && 0 \leq \alpha_i \leq C \quad i = 1, \ldots, N \end{aligned} \tag{6}$$

In [14], it has shown that the SVM's maximal margin property and the minimal norm of weights theory of ELM network are actually consistent. In addition, optimization method based ELM and SVM are very similar in the standard optimization method point of view, but optimization method based ELM has fewer optimization constraints.

2.2 Regularized extreme learning machine

Based on the preliminary ELM, by applying the orthogonal projection method [20], the Moore–Penrose generalized inverse of a matrix can be calculated by $\mathbf{H}^{\dagger} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$, when $\mathbf{H}\mathbf{H}^T$ is nonsingular or $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$, when $\mathbf{H}^T\mathbf{H}$ is nonsingular. In addition, a positive value can be added to the diagonal of $\mathbf{H}\mathbf{H}^T$ or $\mathbf{H}^T\mathbf{H}$ to ensure the invertibility. The use of regularization could lead to a satisfactory solution to balance the empirical risk and the structural risk, which makes the network more robust with respect to the noise and the overfitting problem.

Therefore, if the dimensionality of the feature space is large than the number of training data, that is, $L \gg N$, we have

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}\beta = \mathbf{h}(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \tag{7}$$

While, if the number of training data is much larger than the dimensionality of the feature space, that is, $N \gg L$, an alternative solution could be

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}\beta = \mathbf{h}(\mathbf{x})\left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{T} \tag{8}$$

In this paper, the number of training data of text-independent speaker verification is comparable to the dimensionality of the feature space. Hence, both Eqs. (7) and (8) could be applied. In this paper, Eq. (8) is applied in regularized ELM.

2.3 Kernelized extreme learning machine

In ELM, the feature mapping $\mathbf{h}(\mathbf{x})$ is usually known to the users. Nevertheless, if a feature mapping $\mathbf{h}(\mathbf{x})$ is unknown sometimes, Mercer's conditions could be applied on ELM and we may define a kernel matrix for ELM as

$$\Omega_{\mathrm{ELM}} = \mathbf{H}\mathbf{H}^T \quad : \Omega_{\mathrm{ELM}_{\mathrm{i,j}}} = h(\mathbf{x}_i) \cdot h(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

Thus, the output function of ELM classifier can be modified as

$$\begin{aligned}\mathbf{f}(\mathbf{x}) &= \mathbf{h}(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}\\ &= [K^T(\mathbf{x}, \mathbf{x}_1)\ , \ldots,\ K^T(\mathbf{x}, \mathbf{x}_N)]\left(\frac{\mathbf{I}}{\lambda} + \Omega_{\mathrm{ELM}}\right)^{-1}\mathbf{T}\end{aligned} \tag{10}$$

In kernelized ELM, the feature space $\mathbf{h}(\mathbf{x})$ and the dimensionality of feature space L need not be known to the user. Instead, the kernel $K(\mathbf{u}, \mathbf{v})$ (e.g., $K(\mathbf{u}, \mathbf{v}) = \exp(-\gamma\|\mathbf{u} - \mathbf{v}\|^2)$) is given to the user.

3 Speaker recognition experiment

Generally speaking, a speaker recognition system contains two sections, feature extraction and classification. After obtaining the database of speaker recognition, the desired features are extracted from the speech signal. Then, the extracted features are used as the input to the classifier. The final decision is made according to the output of the classifier. In this paper, the database used to evaluate the ELM and SVM classifiers is ELSDSR corpus, and the Mel-frequency Cepstral Coefficients (MFCC) are extracted from the speech signal and used as the input to the classifiers. The details of corpus, feature extraction, parameter selection of classifiers and experiment results are presented in this section.

3.1 Database

In this paper, we performed experiments on English Language Speech Database for Speaker Recognition (ELSDSR) corpus provided by the department of Informatics and mathematical Modeling (IMM) at Technical University of Denmark (DTU) [8]. For this corpus, voice messages from 22 speakers were collected and the age of the speakers covered from 24 to 63. The data were split into training set and testing set. There were seven paragraphs used for training, and each speaker read all the seven training paragraphs. One testing article was chosen from NOVA home [18], and each speaker read two sentences of the article. Thus, 154 (7 × 22) utterances were provided in the training set, while 44 (2 × 22) utterances were collected for testing. On average, the duration for reading the training data was 83s for all, while the duration for reading the testing data was 17.6s for all. More detailed information about the database can be found in [8].

When running the simulations, especially for parameter selection of different algorithms, it took a long time for each speaker. Due to the time constrain, 10 speakers were selected in the experiment. They are speaker 2, 4, 6, 7, 8, 9, 10, 13, 18, 20 in ELSDSR corpus.

3.2 Feature extraction

The Mel-Frequency Cepstral Coefficients (MFCC) are used in this paper for the front end processing. A general process flow of MFCC is shown in Fig. 1. The source code used for MFCC extraction is based on MFCC algorithm in Speech Processing MATLAB Toolbox with some modifications, and the toolbox could be found in [2].

All the speech signals collected in ELSDSR corpus were relatively noisy-free and sampled at 16,000 Hz with 16-bit resolution. To obtain the acoustic vector from a speech utterance, a 28-dimensional MFCC feature vector was

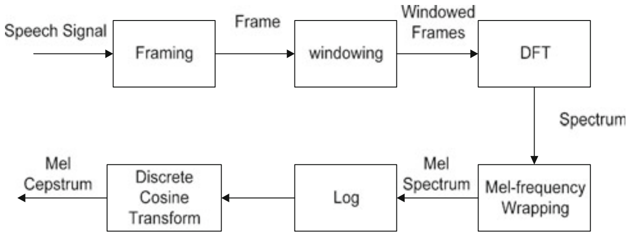

Fig. 1 MFCC block diagram [17]

extracted on 16 ms Hamming Windowing, processing at an 8-ms rate. The mel-cepstral vector was computed using a simulated triangular filter bank of 29 filters on the DFT spectrum. Band-limiting was performed by retaining only the filter-bank outputs from the frequency range 0–8,000 Hz. In addition, no log energy or delta coefficients were computed in the experiments. Furthermore, normalization is applied to the individual features among all the speakers.

After the feature extraction process, the original speech signals of 10 speakers have been converted to 28-dimensional samples. The detailed number of samples obtained for each speaker is presented in Table 1. We merged the samples from the original training set and the testing set into 10 data reservoirs. Each reservoir only contains the data from a specific speaker. The number of samples in the reservoirs is shown in Table 1 as well (i.e., total column).

3.3 Evaluation

The SVM package used for this task is SVM and Kernel Methods MATLAB Toolbox [4]. The source code of the preliminary ELM could be found on [23]. All the evaluations were carried out in the Matlab R2006b environment running on a desktop with Pentium(R) 4 CPU 3GHz and 1GB of RAM. The evaluation of SVM classifiers and classifiers based on ELM and its variants was conducted on 10 selected speakers, and it has two stages.

Table 1 Specifications of speaker verification data

Class	# Samples in original training set	# Samples in original training set	Total
2	9,769	1,197	10,966
4	10,817	2,647	13,464
6	9,579	2,273	11,852
7	12,350	2,997	15,347
8	10,016	2,297	12,313
9	12,963	1,972	14,935
10	9,927	3,136	13,063
13	11,441	1,848	13,289
18	10,353	2,522	12,875
20	10,905	1,160	12,065

In the first stage, for each speaker, three classifiers have been built based on SVM algorithm, optimization method based ELM and regularized ELM, respectively. The training and testing data for each classifier were formed by two classes, positive and negative classes. For both training and testing data, the positive class consisted of 1000 samples that randomly selected from the specific reservoir we interested, while the negative class consisted 999 samples that 111 samples were randomly selected from each remaining reservoir. A total of 20 trials of simulations have been conducted, and classifiers are compared based on the overall accuracy. The overall accuracy is defined using the confusion matrix shown in Table 2. The empirical results are presented in Sect. 3.4.

However, the overall testing accuracy may highly depend on the composition of the testing data. For example, there are two cases, case (a) and case (b). For case (a), the testing data consist of 100 positive samples and 900 negative samples. For case (b), the testing data consist of 100 positive samples and 90 negative samples. For the two cases, the true positive rate (i.e., recall) and false alarm rate (i.e., false positive rate) are the same. However, the overall accuracy increases due to the reduction in the number of negative testing samples.

(a)		(b)	
TP = 90	FP = 200	TP = 90	FP = 20
FN = 10	TN = 700	FN = 10	TN = 70
Acc = 79 %; Recall = 90 %; fp rate = 22 %		Acc = 84 %; Recall = 90 %; fp rate = 22 %	

Therefore, in the second stage, the performance of classifiers was compared according to ROC curve [6, 22]. An ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance. It is a two-dimensional graph in which the true positive rate (i.e., recall) is plotted against the false positive rate (i.e., false

Table 2 Confusion matrix in classification

Predicted class	True class	
	True	False
True	True positive (TP)	False positive (FP)
False	False negative (FN)	True negative (TN)

Overall Accuracy: Acc = (TP + TN)/(TP + FP + FN + TN)

True positive rate or Recall: Recall = TP/(TP + FN)

False positive rate or false alarm rate: fp rate = FP/(FP + TN) alarm rate). An ROC curve depicts relative trade-offs between benefits (true positives) and costs (false positives). The point in the top left corner (0, 1) of a ROC graph denotes perfect classification: 100 % true positive rate and 0 % false positive rate. The performance of any classifier has the smallest distance to the top left corner outperform the rest of classifiers. At this stage, for each speaker, three classifiers have been built based on SVM algorithm, regularized ELM and kernelized ELM, respectively. The samples for training were randomly picked similar to the first stage. For true positive testing, 1000 samples were randomly selected from the specific reservoir interested, while for false alarm testing, 111 samples were randomly selected from each of the remaining reservoir (i.e., total is 999 samples). A total of 20 trials of simulations have been conducted, and the results were used to plot the ROC curve. The empirical results are presented in Sect. 3.4.

3.4 Performance comparison among optimized ELM, regularized ELM and SVM based on overall testing accuracy

In this subsection, the optimization method based ELM (optimized ELM for short), regularized ELM, as well as SVM classifiers were tuned to obtain the optimal parameters, and their best generalization performance in terms of overall testing accuracy and training time will be compared.

For SVM classifiers, the popular Gaussian kernel function $K(u, v) = \exp(-\gamma\|u - v\|^2)$ was used. Since the generalization performance of SVM classifiers usually depends closely on the combination of (C, γ) and the best performance is usually achieved in very narrow range of such combinations, the combinations of cost parameter C and kernel parameter γ have to be chosen carefully to get the best results. In our simulation, for each speaker, 16 different values of C and 12 different values of γ were used to form 192 combinations of $(C, \gamma) : C \in \{0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 1000, 10000\}$; $\gamma \in \{0.1, 0.2, 0.4, 0.8, 1, 2, 5, 10, 20, 100, 1000, 10000\}$. The tuning results of class 9 are presented in Fig. 2 as an example. The optimal parameter is presented in Table 3.

There are two parameters for optimized ELM: cost parameter C and the number of hidden nodes in the hidden layer L. To investigate whether the performance of optimized ELM is sensitive to the combination of (C, L), we have conducted the simulations on a wide range. For each speaker, 15 different values of C and 15 different values of L were used to form 225 combinations of $(C, L) : C \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1, 2, 4, 5, 6, 8, 10, 50, 100, 1000\}$; $L \in \{10, 30, 50, 80, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000\}$. The tuning results of class 9 are presented in Fig. 3 as an example. The optimal parameter is presented in Table 3.

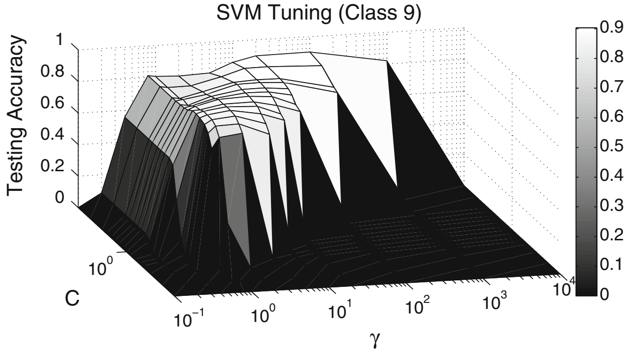

Fig. 2 SVM tuning (class 9)

Table 3 Optimal parameters

Class	SVM (C, γ)	Optimized ELM (C, L)	Regularized ELM (λ, L)
2	(1000,100)	(0.5,200)	(0.05,1000)
4	(10,2)	(0.5,400)	(0.5,1000)
6	(100,20)	(0.5,400)	(1,1000)
7	(50,20)	(0.1,400)	(0.1,1000)
8	(100,5)	(0.5,400)	(0.5,1000)
9	(1000,20)	(0.5,400)	(0.1,1000)
10	(50,10)	(0.01,400)	(0.05,1000)
13	(5,10)	(0.01,200)	(0.001,1000)
18	(100,20)	(0.01,400)	(0.5,1000)
20	(1,5)	(0.01,200)	(0.1,1000)

Similar to optimized ELM, there are two parameters (i.e., λ and L) to be determined for regularized ELM. For each speaker, 13 different values of λ and 9 values of L were used to form 117 combinations of $(\lambda, L) : \lambda \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1, 3, 5, 8, 10, 50, 100, 1000\}$; $L \in \{10, 100, 200, 400, 600, 800, 1000, 2000, 3000\}$. The tuning results of class 4 are presented in Fig. 4 as an example. The optimal parameter is presented in Table 3.

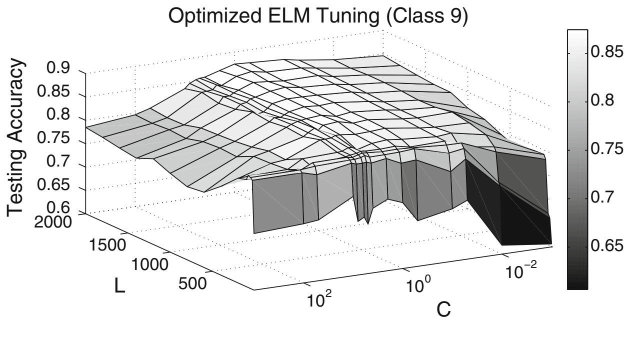

Fig. 3 Optimized ELM tuning (class 9)

The performance comparison between optimized ELM and SVM is presented in Table 4. Optimized ELM classifiers spend less or comparable time in training as compared to SVM classifier. In addition, optimized ELM classifiers outperform SVM classifiers in 7 classes out of 10. The performance of regularized ELM classifiers is shown in Table 5, and it is comparable to the ones obtained by SVM classifiers. The great advantage of regularized ELM classifiers is that it takes much less time in training for all classes.

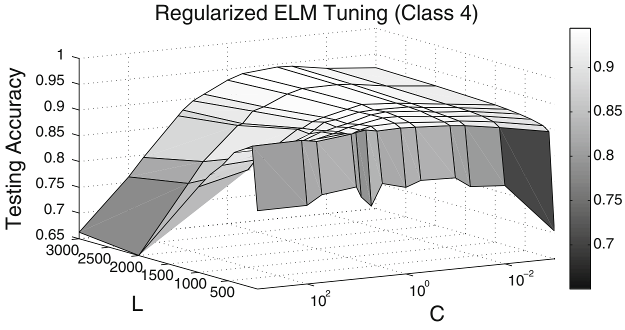

Fig. 4 Regularized ELM tuning (class 4)

3.5 Performance comparison among kernelized ELM, regularized ELM and SVM based on ROC curve

As mentioned earlier, the overall testing accuracy may depend on the composition of the testing data. Hence, in this section, we intend to compare the classifiers by true positive rate as well as false positive rate. A good classifier should provide not only high true positive rate, but also reasonably low false positive rate. In order to investigate the relationship between any single evaluation criterion and the parameter setting of the classifiers, we conduct an experiment using kernelized ELM classifier. Two parameters, λ and γ, have to be tuned for each kernelized ELM classifier. In our simulation, for each speaker, 28 different

Table 4 Performance comparison of optimized ELM and SVM

Class	Optimized ELM			SVM		
	Training time (s)	Testing rate (%)	Testing dev (%)	Training time (s)	Testing rate (%)	Testing dev (%)
2	**9.57**	85.54	2.7	9.94	**88.67**	1.46
4	**9.66**	**96.03**	0.98	10.28	94.81	1.58
6	12.18	82.93	3.23	**10.72**	**84.04**	2.74
7	**8.48**	83.92	2.34	9.20	**87.63**	1.82
8	**10.04**	**89.14**	2.42	10.94	87.98	2.59
9	11.41	**92.61**	1.63	11.43	91.59	1.49
10	**9.69**	**94.75**	1.28	9.90	92.63	1.70
13	9.32	98.82	0.40	9.54	98.10	0.43
18	9.38	**95.89**	0.85	9.39	93.31	1.27
20	8.36	**96.18**	0.61	9.00	95.68	0.53

Bold values indicate that the results obtained is much better than that obtained by the other classifier

Table 5 Performance comparison of regularized ELM and SVM

Class	Regularized ELM			SVM		
	Training time (s)	Testing rate (%)	Testing dev (%)	Training time (s)	Testing rate (%)	Testing dev (%)
2	**1.05**	85.66	2.34	9.94	**88.67**	1.46
4	**1.07**	**96.49**	0.95	10.28	94.81	1.58
6	**1.05**	84.09	3.64	10.72	84.04	2.74
7	**1.04**	85.32	2.51	9.20	**87.63**	1.82
8	**1.06**	**91.04**	1.25	10.94	87.98	2.59
9	**1.04**	**93.8**	1.15	11.43	91.59	1.49
10	**1.04**	92.63	1.61	9.90	92.63	1.70
13	**1.25**	98.16	0.59	9.54	98.10	0.43
18	**1.05**	93.34	1.04	9.39	93.31	1.27
20	**1.06**	92.76	0.96	9.00	**95.68**	0.53

Bold values indicate that the results obtained is much better than that obtained by the other classifier values of λ and 28 different values of γ were used to form 784 combinations of (λ, γ) : $\lambda \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1, 3, 5, 7, 9, 10, 30, 50, 90, 100, 300, 500, 700, 900, 1000, 3000, 5000, 7000, 9000, 10000\}$; $\gamma \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1, 3, 5, 7, 9, 10, 30, 50, 70, 90, 100, 300, 500, 700, 900, 1000, 3000, 5000, 7000, 9000, 10000\}$. The simulation results are presented in Figs. 5 and 6. According to the two figures, it is very obvious that the combination (λ, γ) that gives high positive rate will also produce high false positive rate, which is not desirable in speaker recognition system. The required combination (λ, γ) is the one that can produce high positive rate while maintain low false positive rate. However, it is quite difficult or impossible to tell the optimal combination (λ, γ) from the above two figures. Hence, it is necessary to introduce ROC curve when evaluating the performance of different models..

As explained in section 3.3, ROC curve is a very intuitive graph in which the true positive rate is plotted against the false positive rate and it is less sensitive to data skew. So the performance of different models can be compared even different amount of datasets are used. It is easy to tell the best combination of (λ, γ) from ROC curve that produces a discrete point that is nearest to point (0, 1). To draw the ROC curves, we conduct simulations that vary the parameters of the kernelized ELM classifiers, regularized ELM classifiers and SVM classifiers, and the performance of classifiers were recorded for each combination of parameter.

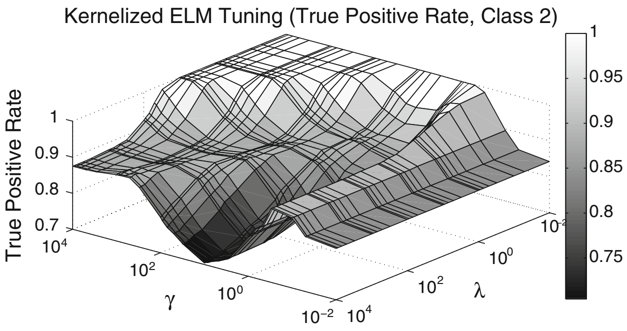

Fig. 5 Kernelized ELM tuning (True Positive Rate, class 2)

More parameter combinations have been considered in this subsection for ROC curve construction. For SVM classifiers, 33 different values of C and 21 different values of γ were used to form 693 combinations of (C, γ) for each speaker: $C \in \{0.001, 0.005, 0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5, 0.7, 0.9, 1, 3, 5, 7, 9, 10, 30, 50, 70, 90, 100, 300, 500, 700, 900, 1000, 3000, 5000, 7000, 9000, 10000\}$; $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1, 3, 5, 7, 9, 10, 30, 50, 70, 90, 100, 300, 500, 700, 900, 1000\}$. While for regularized ELM classifiers, 21 different values of λ and 17 different values of L were used to form 357 combinations of (λ, L) for each speaker: $\lambda \in \{0.0001, 0.001, 0.005, 0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5, 0.7, 0.9, 1, 3, 5, 7, 9, 10, 100, 1000\}$; $L \in \{10, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600, 2800, 3000\}$. The ROC curve of class 9 is shown in Fig. 7 as an example. The optimal parameters selected for regularized ELM classifiers and kernelized ELM classifiers are shown in Tables 6 and 7, respectively, and they are compared with the results obtained by SVM classifiers.

Observed from Tables 6 and 7, although both regularized ELM classifiers and kernelized ELM classifiers produce higher true positive rate as compared to SVM classifiers for most of the classes, the false positive rate is higher for most of the classes as well. More specifically, we compare the distance between the optimal performance point obtained by each classifier and the top left corner (0,1), and the results are presented in Table 8. The average training time spent by each classifier is shown in the same table. From Table 8, kernelized ELM classifiers perform the best in class 2, 6, 8 and 13; regularized ELM classifiers perform the best in class 7 and 20, while SVM classifiers perform the best in class 4, 9, 10 and 18. In addition, the

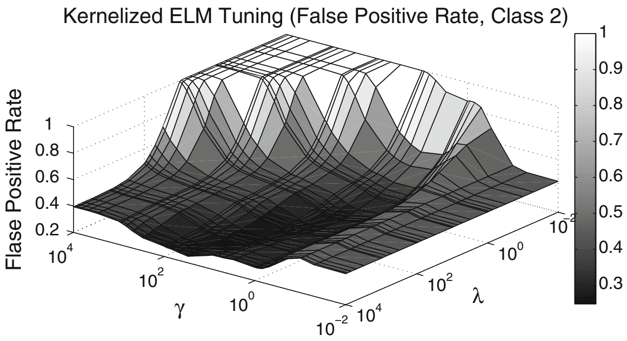

Fig. 6 Kernelized ELM tuning (False Positive Rate, class 2)

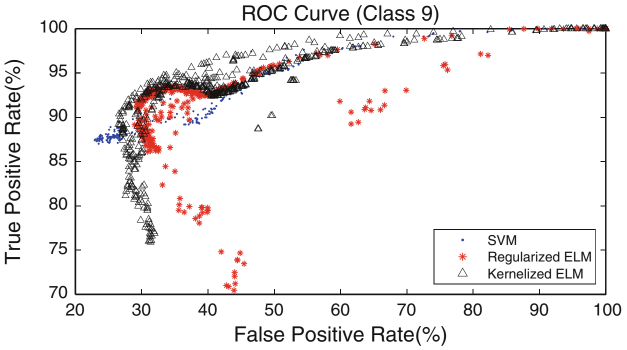

Fig. 7 ROC curve (class 9)

Table 6 Optimal parameters and performance comparison of regularized ELM and SVM

Class	Regularized ELM			SVM		
	(C, L)	True positive (%)	False positive (%)	(C, γ)	True positive (%)	False positive (%)
2	(0.9,2800)	**85.36**	28.25	(3,1)	83.83	28.57
4	(7,3000)	87.49	38.03	(3,0.7)	**91.17**	**28.37**
6	(0.5,2800)	**85.81**	25.45	(9000,10)	83.63	26.19
7	(1,2600)	84.38	**17.42**	(3000,10)	85.71	22.32
8	(0.7,2400)	**91.08**	24.94	(7,0.5)	84.62	**19.15**
9	(0.7,3000)	**91.68**	29.25	(1000,5)	87.40	**22.92**
10	(1,3000)	**89.45**	32.68	(1,0.7)	87.70	**24.21**
13	(1,3000)	93.37	20.68	(1,0.5)	93.35	20.04
18	(3,3000)	**90.57**	27.87	(3,0.7)	86.01	**17.66**
20	(0.9,2400)	90.79	15.02	(0.9,0.7)	**92.46**	15.97

Bold values indicate that the results obtained is much better than that obtained by the other classifier

Table 7 Optimal parameters and performance comparison of kernelized ELM and SVM

Class	Kernelized ELM			SVM		
	(λ, γ)	True positive (%)	False positive (%)	(C, γ)	True positive (%)	False positive (%)
2	(10,5)	84.52	26.98	(3,1)	83.83	28.57
4	(7000,70)	92.76	36.90	(3,0.7)	91.17	**28.37**
6	(3,3)	**88.10**	**25.00**	(9000,10)	83.63	26.19
7	(3,3)	83.53	22.52	(3000,10)	85.71	22.32
8	(5,3)	85.71	17.56	(7,0.5)	84.62	19.15
9	(9,1)	**90.38**	26.59	(1000,5)	87.40	**22.92**
10	(30,3)	**91.17**	31.65	(1,0.7)	87.70	**24.21**
13	(10,3)	92.46	19.44	(1,0.5)	93.35	20.04
18	(3000,100)	**91.07**	21.23	(3,0.7)	86.01	**17.66**
20	(3,0.9)	**94.35**	17.96	(0.9,0.7)	92.46	15.97

Bold values indicate that the results obtained is much better than that obtained by the other classifier

Table 8 Comparison of the distances to point (0,1) and the time consumed

Class	Distance			Training time(s)		
	K-ELM	R-ELM	SVM	K-ELM	R-ELM	SVM
2	**0.10**	0.10	0.11	**1.03**	6.95	18.66
4	0.14	0.16	**0.09**	**0.95**	8.10	67.46
6	**0.08**	0.08	0.10	**1.07**	6.97	17.12
7	0.08	**0.05**	0.07	**1.08**	5.92	13.31
8	**0.05**	0.07	0.06	**1.06**	4.99	278.67
9	0.08	0.09	**0.07**	**1.11**	8.10	290.34
10	0.11	0.12	**0.07**	**1.06**	8.10	27.02
13	**0.04**	0.05	0.04	**1.07**	8.13	130.07
18	0.05	0.09	**0.05**	**0.96**	8.11	67.94
20	0.04	**0.03**	0.03	**1.10**	4.98	15.89

Bold values indicate that the results obtained is much better than that obtained by the other classifier

K-ELM Kernelized ELM, *R-ELM* regularized ELM

time spent by regularized ELM classifiers is much less than SVM classifiers for all the classes, while kernelized ELM classifiers spend the least time in training for all the classes. Therefore, we may conclude that ELM classifiers and its variants can perform better than SVM classifiers with much less training time.

4 Conclusion

Speaker recognition has been an interesting research field for the last decades. In this paper, the extreme learning machine (ELM) and its variants are examined for speaker verification task and compared with SVM classifiers. ELMs have been proposed for generalized single hidden layer feedforward networks with a wide variety of hidden nodes. They are extremely fast in learning and perform well on many artificial and real regression and classification applications. Empirical studies in this paper have shown that ELM classifiers and its variants could perform better than SVM classifiers on the dataset we have with less training time.

References

1. Atal B (1976) Automatic recognition of speakers from their voices. In: Proceedings of the IEEE, vol 64, pp 460–475
2. Brookes M (2000) Voicebox: speech processing toolbox for matlab. World Wide Web, http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html
3. Campbell JP (1997) Speaker recognition: a tutorial. In: Proceedings of the IEEE, vol 85, pp 1437–1462
4. Canu S, Grandvalet Y, Guigue V, Rakotomamonjy A (2005) Svm and kernel methods matlab toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France
5. Doddington GR (1985) Speaker recognition-identifying people by their voices. In: Proceedings of the IEEE, vol 73, pp 1651–1664
6. Egan JP (1975) Signal detection theory and ROC-analysis. Academic Press, New York
7. Farrell KR, Mammone RJ, Assaleh KT (1994) Speaker recognition using neural networks and conventional classifiers. IEEE Trans Speech Audio Process 2(1):194–205
8. Feng L, Hansen LK (2004) A new database for speaker recognition
9. Furui S (1997) Recent advances in speaker recognition. Patt Recognit Lett 18:859–872
10. Huang GB, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70:3056–3062
11. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of international joint conference on neural networks (IJCNN'04), vol 2, Budapest, pp 985–990
12. Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
13. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
14. Huang GB, Ding XJ, Zhou HM (2010) Optimization method based extreme learning machine for classfication. Neurocomputing 74(1-3):155–163
15. Huang GB, Wang D, Lan Y (2011) Extreme learning machine: a survey. Int J Mach Learn Cybernet 2:107–122
16. Huang GB, Zhou H, Ding X, Zhang R (2011) Extreme learning machine for regression and multi-class classification. IEEE Trans Syst Man Cybernet (in press)
17. Mut O, Göktürk M (2005) Improved weighted matching for speaker recognition. In: Proceedings of World Academy of Science, Engineering and Technology, vol 5, pp 170–172
18. NOV (1997) Nova online. http://www.pbs.org/wgbh/nova/pyramid
19. Pruzansky S (1963) Pattern-matching procedure for automatic talker recognition. J Acoustical Soc Am 35(3):354–358
20. Rao CR, Mitra SK (1971) Generalized inverse of matrices and its applications. Wiley, New York
21. Rosenberg A (1976) Automatic speaker verification: a review. In: Proceedings of the IEEE, vol 64, pp 475–487
22. Swets JA, Dawes RM, Monahan J (2000) Better decisions through science. Scientific American, pp 82–87
23. Zhu QY, Huang GB (2004) Source codes of ELM algorithm. In: http://www.ntu.edu.sg/home/egbhuang/, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore