Exploring ANN Back-Ends for i-Vector Based Speaker Age Estimation

Anna Fedorova[1], *Ondrej Glembek*[2], *Tomi Kinnunen*[1], *Pavel Matějka*[2]

[1]University of Eastern Finland, Finland
[2]Brno University of Technology, Czech Republic

annafed@student.uef.fi, glembek@fit.vutbr.cz, tkinnu@cs.joensuu.fi, matejkap@fit.vutbr.cz

Abstract

We address the problem of speaker age estimation using i-vectors. We first compare different i-vector extraction setups and then focus on (shallow) artificial neural net (ANN) back-ends. We explore ANN architecture, training algorithm and ANN ensembles. The results on NIST 2008 and 2010 SRE data indicate that, after extensive parameter optimization, ANN back-end in combination with i-vectors reaches mean absolute errors (MAEs) of 5.49 (females) and 6.35 (males), which are 4.5% relative improvement in comparison to our support-vector regression (SVR) baseline. Hence, the choice of back-end did not affect the accuracy much; a suggested future direction is therefore focusing more on front-end processing.

Index Terms: age estimation, i-vector, multilayer perceptron

1. Introduction

Automatic *age estimation* of a speaker has received increased interest recently. The internet provides a wide range of possibilities for commercial use, in the context when there is no direct contact with the client; meta-information about the user, such as her language, gender or age can be helpful for delivering appropriate products and services [1, 2]. Automatic age recognition systems can be a useful tool in forensic investigation as well [3].

In the past, many methods have been studied for speaker age estimation, including, for instance, support vector machines (SVMs) [4, 5] and Gaussian mixture model (GMM) supervectors [6]. One of the most recent approaches, adopted also by us, uses *i-vector* representation of utterances [7, 8] originally devised for speaker verification [9] but later adopted to other tasks such as language [10] and accent [11] recognition. We use i-vectors as inputs to *support vector regression* (SVR) back-end, similar to [7, 8] where it was found one of the best methods.

We would like to study the applicability of *artificial neural network* (ANN) back-ends for age prediction. Recently, [12] studied such approach preliminarily, though the main focus was not in age estimation. In contrast, we focus solely on age estimation by providing detailed analyses how to configure ANNs for the task. Specifically, we study the importance of features, i-vector normalization, ANN training method, number of hidden neurons, two-vs-one hidden layer architectures and the use of multiple ANN predictors. We attempt to provide answers as which of these choices are most important and whether ANN back-end will outperform SVRs studied in detail by [7, 8].

2. Speaker Age Estimation

The problem of automatic speaker age recognition can be formulated as follows. Given a set of N training utterances $\{X_n\}$ with their age labels $\{Y_n\}$, train a system that predicts, for an unseen utterance X_{test} its age Y_{test} as accurately as possible.

2.1. Baseline Approach

We consider the approach of [8] as our baseline. It was found to be one of the most accurate approaches in the comparisons of [8] including GMM supervector based SVR and other predictors. The method is briefly described as follows. In the training phase, i-vectors for each utterance are extracted, followed by within-class covariance normalization (WCCN) [13] to supress session and channel effects. WCCN normalizes the average within-class covariance of the feature space (*e.g.* i-vector space) to identity matrix. The WCCN matrix, $\boldsymbol{B}$, is obtained from

$$\boldsymbol{B}\boldsymbol{B}^\top = \left[\frac{1}{J}\sum_{j=1}^{J}\frac{1}{N_j}\sum_{i=1}^{N_j}(\boldsymbol{w}_j^i - \bar{\boldsymbol{w}}_j)(\boldsymbol{w}_j^i - \bar{\boldsymbol{w}}_j)^\top\right]^{-1},$$

where $\boldsymbol{w}_j^i$ is the i-th feature vector of the j-th speaker, the total number of speakers is J and the number of vectors for the j-th speaker is N_j. Finally, $\bar{\boldsymbol{w}}_j$ denotes the mean feature vector for the j-th speaker. After training $\boldsymbol{B}$, all the training and future test vectors $\boldsymbol{x}$ are normalized by $\boldsymbol{x} \mapsto \boldsymbol{B}^\top\boldsymbol{x}$.

The WCCN-normalized i-vectors are then presented to support vector regression model (SVR) [14] together with their age labels to train it. SVR generalizes the idea of support vector machines (SVMs), which finds the unique hyperplane in feature space separating two classes with maximum margin. In particular, SVR finds a hyperplane for which most of the training data points would lie no further than distance ϵ from it. That is, we search for parameters $\boldsymbol{v}$ and z such that the function $f(\boldsymbol{a}) = \boldsymbol{v}^T\Phi(\boldsymbol{a}) + z$ accurately predicts the output for input $\boldsymbol{a}$. Here, $\Phi(\boldsymbol{a})$ denotes a feature mapping function defined in advance. Function f can be found by solving an optimization problem with linear constraints as detailed in [14]. In practice, we apply LS-SVMlab toolbox [1] for SVR training. In the test stage, an i-vector is extracted, WCCN transformation is applied and the normalized vector is fed to the regression model, which produces a predicted age (a scalar).

2.2. Modifications to i-Vector Front-End

The baseline approach described above was found effective for age estimation in [8]. However, the resulting i-vectors are of course strongly dependent on the acoustic features used to construct them. We consider a few revisions that will be explored in our experiments. Firstly, consider cepstral mean and variance normalization (CMVN), which normalizes each cepstral coeffi-

[1]www.esat.kuleuven.be/sista/lssvmlabs

cient c in frame t by $\hat{c}(t) = (c(t) - \mu)/\sigma$, where

$$\mu = \frac{1}{L}\sum_{n=t-\frac{L}{2}}^{t+\frac{L}{2}-1} c(n) \text{ and } \sigma^2 = \frac{1}{L}\sum_{n=t-\frac{L}{2}}^{t+\frac{L}{2}-1} (c(n)-\mu)^2$$

denote, respectively, the mean and variance of c computed over L frames. It produces features with zero-mean and unity variance over the normalization window. A special case is when L equals utterance length and variance normalization is discarded.

Another variation is achieved by replacing MFCCs by other features. We consider two such alternatives. Firstly, *shifted delta cepstral coefficients* (SDCs) [15] are computed as,

$$s_{iN+j}(t) = c_j(t + iP + d) - c_j(t + iP - d), i = 0, .., k - 1.$$

Here, $c_j, j = 1, .., M - 1$ are the base MFCCs. SDCs are defined by four parameters, the number of cepstral coefficients (M), the time difference between the frames (d), the time shift between two blocks (P) and the number of blocks (k). In language recognition, M is usually set to 12, d to 1, P and k are set to 3. Thus, SDCs are basically k blocks of delta cepstral coefficients. SDCs add contextual temporal information to the feature vectors which can be useful in terms of age estimation.

Our second alternative features are so-called *Perseus* features which are a combination of MFCCs and MMeDuSa [16] features. In contrast to MFCCs, Perseus uses gammatone filter bank instead of mel filter bank and adds a vector sub-frame energy estimations to each frame. It also uses 1/15-th root compression instead of log-compression in MFCCs.

2.3. Artificial Neural Network (ANN) Back-End

ANNs are very powerful in function approximation, classification and other tasks. Here we study their usefulness in age estimation. To this end, we use ANN in place of SVR. We consider the *multilayer perceptron* (MLP), in which one neuron is described by $\boldsymbol{y} = g(\boldsymbol{w}^T\boldsymbol{x} + b)$, $\boldsymbol{x}$ being the input vector, $\boldsymbol{w}$ the weights and b the bias. The function g is activation function, here, a hyperbolic tangent. In MLP, the neurons are combined in multiple layers connected to each other through their inputs and outputs. The first and the last layers of network are called *input* and *output* layers, while all the rest are *hidden* layers.

MLPs can be trained using a number of techniques, such as the backpropagation algorithm [17]. We consider two methods, *stochastic gradient descent* (SGD) [18] and *Broyden-Fletcher-Goldfarb-Shanno* algorithm (BFGS) [19]. SGD uses the fact that total error for the whole training set, E, is the sum of the errors for individual training cases. While a gradient descent algorithm would update weights at iteration τ based on gradient on the whole data (1), SGD updates them according to gradients of one data point at a time (2). The parameter α in both cases denotes the step size to the chosen direction of the error surface.

$$w^{(\tau+1)} = w^{\tau} - \alpha\nabla E(w^{\tau}) \quad (1)$$
$$w^{(\tau+1)} = w^{\tau} - \alpha\nabla E_n(w^{\tau}) \quad (2)$$

In practice, we divide the training data into several batches and weight updates are done according to gradient of the error in every batch.

While the SGD training algorithm utilizes only the first derivative of the error function, BFGS requires second order derivatives as well. It belongs to the class of so-called *quasi-Newton* methods. BFGS does *not* explicitly calculate the actual Hessian matrix of the second derivatives but makes an approximation, $\boldsymbol{H}$, and uses this in optimization. The weights are updated by searching for the direction p^{τ} as a solution to (3), and make a step α in this direction as $w^{(\tau+1)} = w^{\tau} + \alpha p^{\tau}$.

$$\boldsymbol{H}^{\tau}p^{\tau} = \nabla E_n(w^{\tau}) \quad (3)$$

In practice, we use a modification of BFGS known as *limited memory* BFGS [20].

An important further consideration is to prevent ANNs being overfitted to training data. There are various techniques to cope with the problem, such as *early stopping* [21], *dropout* algorithm [22] and ℓ_2- regularization [23]. We adopt this last strategy, which adds a quadratic term to the training objective to penalize for large weights.

3. Experimental Setup

The data we use for experiments is NIST speaker recognition evaluation (SRE) data from years 2008 and 2010. They contain a large number of speakers and rich metadata, including speaker's age. Speakers of two corpuses are not intersecting. We consider utterances from speakers between 20 to 70 years old as there were too few utterances for other speakers. Table 1 and Fig. 1 summarize the statistics of the selected data. For our experiments we use exactly the same evaluation setup as [7, 8]. To train the UBM and the T-matrix, we use all the available NIST SRE corpuses except NIST 2008 and NIST 2010.

Table 1: Summary of the data used for age estimation

	NIST 2008	NIST 2010
Number of speakers	1154	442
Number of utterances	3859	5583
Quality	Telephone	Telephone
Sampling rate	8.0 kHz	8.0 kHz

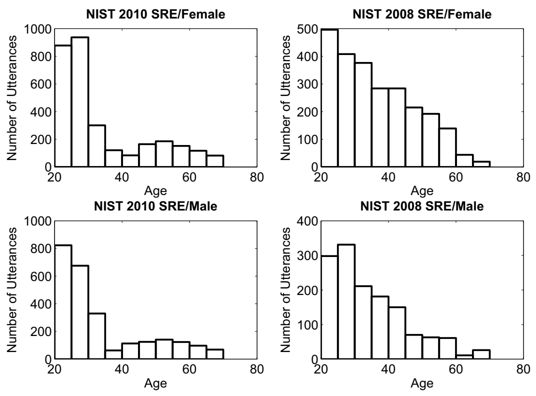

Figure 1: Age histogram of the selected speech utterances from NIST 2008 and 2010 corpuses

To gauge age estimation accuracy, we consider two objective measures utilized also in [7, 8]. The first one is *mean absolute error* (MAE), $\text{MAE} = (1/N)\sum_{n=1}^{N} |\hat{y}_n - y_n|$, where N is the number of test segments, $\hat{y}_n$ is the predicted age by a regression model and y_n is the chronological age that serves as the ground truth. Smaller MAE implies better age predictions, on average. The second measure is *Pearson's correlation coefficient* between the vectors of estimated and chronological ages:

$$\rho = \frac{1}{N-1}\sum_{n=1}^{N}\left(\frac{\hat{y}_n - \mu_{\hat{y}}}{\sigma_{\hat{y}}}\right)\left(\frac{y_n - \mu_y}{\sigma_y}\right).$$

Here, $\mu_{\hat{y}}$ and $\sigma_{\hat{y}}$ denote, respectively, the mean and standard deviation of estimated ages and μ_y and σ_y correspond to the same measures of actual ages. Higher ρ is considered better.

For the sake of consistency when comparing the results, we adopt the same experimental scheme as [7, 8]. All the data used for age estimation is divided into 15 folds so that speakers in different folds do not overlap. Then, 15 independent tests are executed so that 14 folds are used for training while the 15th fold serves as a held-out test set. The final MAE and ρ values are their averages over the 15 test folds. Each time, two gender-dependent age estimators are trained and the results are presented separately for males and females.

4. Results and Analyses

4.1. Baseline Results

Baseline experiment was first carried out as follows. We use 60-dimensional features (19 MFCCs with their energy, deltas and double-deltas) normalized using short-term CMVN [24]. Using these features, we extract 400-dimensional i-vectors to train SVR. We used grid search to find optimal values for the SVR parameters. Table 2 shows the optimized baseline along with the results reported in [8]. The results are close to each other and differences likely caused by differences in random division into 15 folds and UBM/i-vector data selections.

Table 2: MAE (in years) and ρ for baseline approach

	Male		Female	
	MAE	ρ	MAE	ρ
Our baseline	6.65	**0.73**	**5.75**	**0.80**
Baseline in [8]	**6.53**	**0.73**	5.78	0.81

4.2. Impact of Features

We next study the effect of features by using the SVR back-end and varying the acoustic front-end, see Table 3. The first three rows use exactly the same 60-dimensional MFCCs with varied options for feature normalization. The last two rows consider replacing MFCCs by SDCs or Perseus features described in Section 2.2. In both cases, the feature vectors have the same dimensionality as in the baseline approach, 60. No normalization is applied to them.

The best performance for both male and female speakers is achived with short-term CMVN. No obvious benefits are obtained using Perseus or SDC features.

4.3. Neural Networks

We now turn our attention to the MLP back-end. We use MFCC-based 400-dimensional i-vectors. For the majority of our experiments, we use MLP with a single hidden layer. Penalties of 0.1 and 0.01 for the weights of the first and second layers are used, respectively. The learning rate in all experiments is set to 0.5. These values were optimized in initial experiments utilizing on MLP with 512 neurons in the hidden layer.

Effect of the training algorithm: In the first experiment, we train a single hidden layer MLP and study the effect of the training method. Fig. 2 shows dependency of MAE and ρ on the number of hidden neurons for both the SGD [18] and the BFGS [19] methods; the results are shown for males only. For females the trends are similar. As Fig. 2 indicates, BFGS suffers from too many neurons in the hidden layer. For SGD, in turn, adding neurons helps. We did not conduct further experiments with larger number of neurons due to high computational costs. When the number of neurons is low, BFGS outperforms SGD, but for larger network,the order is reversed. Training of SGD is generally faster, too.

Table 3: MAE (in years) and ρ for SVR age estimators trained on i-vectors with various setups

	Male		Female	
I-vector Setup	MAE	ρ	MAE	ρ
no CMVN	6.77	0.70	5.85	0.79
short-term CMVN	**6.65**	**0.73**	**5.75**	**0.80**
utterance CMVN	6.73	0.72	5.78	**0.80**
Perseus features	7.01	0.69	5.99	0.79
SDC features	6.72	0.72	6.01	0.78

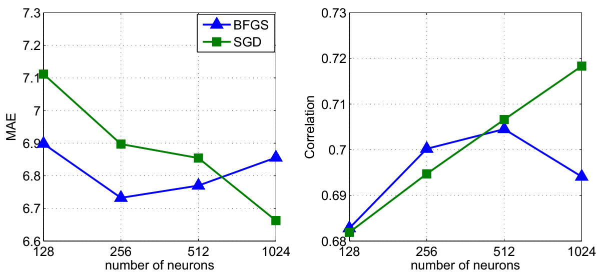

Figure 2: MAE (in years) and ρ for male speakers

Effect of WCCN: In [8], Bahari *et al.* found WCCN applied to i-vectors to improve SVR-based age estimator performance. Interestingly, [8] reported that improvement was achieved when each speaker was treated as a different class. Probably a more meaningful strategy, training WCCN using discrete age classes, actually decreases performance of age estimator. For this reason, and since our baseline results (Table 2) are similar, we study WCCN using speaker labels only for convenience. Tables 4 and 5 compare the impact of WCCN for networks of various sizes. The same single hidden layer architecture and SGD training algorithm are adopted as before. For large networks (512 and 1024 hidden units), WCCN helps and will be used in all the remaining experiments.

Table 4: MAE (in years) and ρ of the neural network age estimator with and without WCCN for female speakers

	no WCCN		WCCN	
Size of hidden layer	MAE	ρ	MAE	ρ
128	**6.19**	**0.77**	6.32	0.76
256	**5.93**	**0.78**	6.12	0.77
512	5.91	0.78	**5.72**	**0.80**
1024	5.66	0.80	**5.49**	**0.81**

Linear discriminant analysis: In the last experiment, we investigate whether *linear discriminant analysis* LDA [25] – as a session compensation and dimensionality reduction tool – can improve age prediction accuracy. As in the case of WCCN, we treat each speaker as a separate class. Table 6 shows the results

Table 5: Same as Table 4 but for males

	no WCCN		WCCN	
Size of hidden layer	MAE	ρ	MAE	ρ
128	**7.11**	**0.68**	7.25	0.67
256	**6.90**	**0.69**	7.06	0.69
512	6.85	0.71	**6.48**	**0.73**
1024	6.66	0.72	**6.35**	**0.74**

for different target size for reduced input to an MLP with 512 hidden neurons. As before, the training algorithm is SGD. In general, LDA does not affect performance much. For the sake of speed and resource consumption, it can still be beneficial.

Table 6: Effect of LDA dimensionality reduction.

	Male		Female	
Target dim.	MAE	ρ	MAE	ρ
100	6.71	0.69	**5.71**	0.79
200	6.70	0.69	5.72	0.79
300	**6.46**	**0.73**	5.73	**0.80**
No LDA	6.48	**0.73**	5.72	**0.80**

Neural networks ensembles: It is known that combining results of several predictors (e.g. neural networks) can help to improve accuracy compared to the individual predictors [26]. Therefore, here we combine several age estimation networks. One way is to train different networks by random initializations of networks that share the same architecture and another is to change the number of hidden neurons or training method. Table 7 represents these results.

Table 7: Combinations of different ANNs (averaged outputs).

	Male		Female	
System configuration	MAE	ρ	MAE	ρ
1.$n_1 = 256$, SGD	7.06	0.69	6.11	0.77
2.$n_2 = 512$, SGD	6.53	**0.73**	5.72	0.79
3.$n_3 = 1024$, SGD	**6.35**	**0.73**	**5.49**	**0.81**
4.$n_4 = 512$, BFGS	6.66	0.71	5.69	0.79
$1 + 2 + 3$	**6.42**	**0.75**	**5.56**	**0.81**
$2 + 2 + 2$	6.45	0.73	5.63	**0.81**
$2 + 4$	6.58	0.72	5.59	0.80

For all the combined networks, WCCN is applied. The first four lines correspond to single networks having different numbers of hidden neurons (n_i) with jointly varied training algorithm. The last three rows show performance of a few combinations of these base networks.In each case, we simply average outputs of the individual networks. As expected, combination of several age estimators is helpful. The best improvement achieved for combinations of networks having the same size.

Neural networks with two layers: After performing several experiments with single hidden layer, we now investigate the effect of network architecture closer by adding more nonlinear layers. Fixing all the other design choices as in the case of one hidden layer (SGD training with WCCN), we attempt to find appropriate size for each layer of the revised network. Fig. 3 shows results for networks with 1024 neurons in the first hidden layer while the size of the second layer is varied. Fig. 4 represents the opposite case, when size of the second layer is fixed to 512 neurons and the number of neurons in the first layer is varied.

Fig. 3 and Fig. 4 suggest that the best performance is achieved when the sizes of the two layers are equal. But even in this case, this alternative architecture does not outperforms the best results achieved by a single layer network. One problem is that we used parameter values optimized for single layer network. They may require re-optimization for the 2-layer structure.

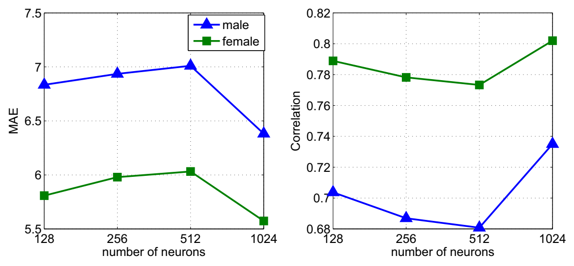

Figure 3: Performance for 2- layers network estimator of male and female speakers' age. Size of second layer is various

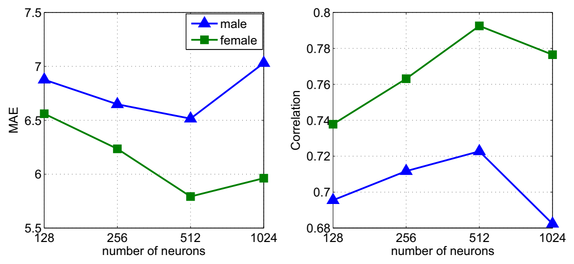

Figure 4: Performance for 2- layers network estimator of male and female speakers' age. Size of first layer is various

5. Conclusion

We studied age estimation using i-vectors. Our findings on NIST 2008 and 2010 are: (1) conventional MFCCs with short-term CMVN worked best as the features for i-vector extraction; (2) WCCN, treating speakers as classes, helped; (3) LDA did not help considerably; (4) BFGS was overfit with more than 256 hidden neurons while SGD was stable even with 1024 neurons; (5) SGD produced more accurate age predictions with faster computation; (6) only modest gain was obtained from an ensemble of ANNs over a single predictor; (7) no clear benefits were obtained with two-layer structure. Based on the last three last findings, a single network with a single hidden layer, trained with SGD, is the recommended choice. Considering the results as a whole, the best improvement from MLP over SVR was 4.5 % relative reduction in MAE. This suggests that the back-end may not have so much effect when the already-compressed i-vectors are used as input features. Future work should thus address alternative utterance parameterizations in place of i-vectors.

6. Acknowledgements

The authors would like to thank M.H.Bahari for sharing trial lists for age estimation used in his paper.

This work was partially supported from Academy of Finland (proj. no. 253120 and 283256) and BUT. The work of the first author was carried as ERASMUS exchange, during her visit to Brno University of Technology.

7. References

[1] S. E. Shepston, Z.-H. Tan, and S. H. Jensen, "Audio-based age and gender identification to enhance the recommendation of tv content," *IEEE Transactions on Consumer Electronics*, vol. 59, pp. 721–729, 2013.

[2] M. Feld, F. Burkhardt, and C. Muller, "Automatic speaker age and gender recognition in the car for tailoring dialog and mobile services," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[3] M. Li, K. J. Han, and S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech and Language*, vol. 27, pp. 151–167, 2013.

[4] D. Mahmoodi, A. Soleimani, and H. Marvi, "Age estimation based on speech features and support vector machine," in *3rd Computer Science and Electronic Engineering Conference (CEEC)*, 2011.

[5] C. van Heerden, E. Barnard, M. Davel, C. van der Walt, E. van Dyk, M. Feld, and C. Muller, "Combining regression and classification methods for improving automatic age recognition," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[6] T. Bocklet, A. Maier, and E. Noth, "Age determination of children in preschool and primary school age with gmm-based supervectors and support vector machines/regression," in *Text, Speech and Dialogue*, vol. 5246, 2008, pp. 253–260.

[7] M. H. Bahari, M. McLaren, H. V. hamme, and D. van Leeuwen, "Age estimation from telephone speech using i-vectors," in *Proceedings of Interspeech*, 2012, pp. 506–509.

[8] ——, "Speaker age estimation using i-vectors," *Engineering Applications of Artificial Intelligence*, vol. 34, pp. 99–108, 2014.

[9] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Tansactions on audio, speech, and language processing*, vol. 19, pp. 788–798, 2011.

[10] N. Dehak, P. A.Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proceedings of Interspeech*, 2011.

[11] M. H. Bahari, R. Saeidi, H. V. hamme, and D. V. Leeuwen, "Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervect," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2013, pp. 7344–7348.

[12] A. H. Poorjam, M. H. Bahari, and H. V. hamme, "Multitask speaker profiling for estimating age, height, weight and smoking habits from spontaneous telephone speech signals," in *4th International eConference on Computer and Knowledge Engineering (ICCKE)*, 2014, pp. 7–12.

[13] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Proceedings of Interspeech*, 2006.

[14] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, p. 199222, 2004.

[15] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, and R. J. Greene, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proceedings of Interspeech*, 2002.

[16] V. Mitra, M. McLaren, H. Franco, M. Graciarena, and N. Scheffer, "Modulation features for noise robust speaker identification," in *Proceedings of Interspeech*, 2013.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[18] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, 1998.

[19] S. Review, "Quasi-newton methods, motivation and theory," *SIAM Review*, vol. 19, pp. 46–89, 1977.

[20] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of Computation*, vol. 35, pp. 773–782, 1980.

[21] R. Caruana, S. Lawrence, and L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," *Advances in Neural Information Processing Systems*, vol. 13, pp. 402–408, 2001.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[23] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," *Advances in Neural Information Processing Systems*, vol. 4, pp. 950–957, 1992.

[24] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, pp. 133 –147, 1998.

[25] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis - a brief tutorial," *Advances in Neural Information Processing Systems*, vol. 13, pp. 402–408, 2001.

[26] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 993–1001, 1990.