On the Use of Ensemble X-Vector Embeddings for Improved Sleepiness Detection

José Vicente Egas-López[1(✉)], Róbert Busa-Fekete[3], and Gábor Gosztolya[1,2]

[1] Institute of Informatics, University of Szeged, Szeged, Hungary
egasj@inf.u-szeged.hu
[2] MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[3] Google Research, New York, NY, USA

Abstract. The state-of-the-art in speaker recognition, called x-vectors, has been adopted in several computational paralinguistic tasks, as they were shown to extract embeddings that could be efficiently utilized as features in the subsequent classification or regression step. Nevertheless, similarly to all neural networks, x-vectors might also prove to be sensitive to several training meta-parameters such as the number of hidden layers and neurons, or the number of training epochs. In this study we experimentally demonstrate that the performance of x-vector embeddings is also affected by the random seed of the initial weight initialization step before training. We also show that, by training an ensemble learning method by repeating x-vector DNN training, we can make the utterance-level predictions more robust, leading to notable improvements in the performance on the test set. We perform our experiments on the publicly available Dusseldorf Sleepy Language Corpus, for estimating the degree of sleepiness. Improving upon our previous results, we present the highest Spearman's correlation coefficient on this dataset that was achieved by a single method.

Keywords: Human-computer interaction · Computational paralinguistics · X-vectors · Ensemble learning · Sleepiness detection

1 Introduction

Excessive daytime sleepiness (EDS) is usually considered to be caused by sleep deprivation, sleep disorders (e.g. apnea, which is the cessation of breathing), or by insomnia (the inability to fall asleep) [11]. Instant identification of the levels of sleepiness for a subject might be crucial for preventing accidents, analyzing when to recommend a break, or even minimizing the mortality risk caused by sleep deprivation. Moreover, as EDS is catalogued as a symptom rather than a condition, it could also be caused by latent neurological or psychiatric disorders [12,13,20]. In that case, the early diagnosis of EDS could be helpful for diminishing the effects of an underlying problem in a subject. Patients with sleep disorders often show symptoms of tiredness and fatigue, which, amongst other things, might affect the way they produce their speech. This modality – that

S. R. M. Prasanna et al. (Eds.): SPECOM 2022, LNAI 13721, pp. 178–187, 2022.
https://doi.org/10.1007/978-3-031-20980-2_16

is, the speech – could be a non-intrusive and economic way of controlling and monitoring the degree of sleepiness of the subjects.

A key problem in computational paralinguistic tasks is the choice of the features extracted from the audio utterances. Besides developing task-dependant attributes, a significant direction of research is to apply general-purpose attributes in paralinguistic tasks. Such an attribute set is the 'ComParE functionals', developed by Schuller et al. [19], consisting of utterance-level statisticals (e.g. mean, standard deviation, 1st and 99th percentiles) of frame-level features. Other frequently applied features are the i-vectors [2], originally developed for speaker recognition, which were applied in various paralinguistic tasks such as determining the cognitive load of the speaker [21], or estimating the speaker's age [7]. The current state-of-the-art technique for speaker recognition is the so-called *x-vector* approach [24], which employs a Deep Neural Network to map variable-length utterances to fixed-dimensional embeddings. A handful of previous studies exploited x-vector embeddings in computational paralinguistic tasks; for instance, to classify emotion from the speech of subjects [14], and to screen neuro-degenerative diseases like Alzheimer's Disease [28], and Parkinson's Disease [10]. In previous studies, x-vectors were applied for sleepiness detection as well [3,9].

Since the x-vector feature extractors are neural networks, they might prove to be sensitive to several training meta-parameters such as number of hidden layers and neurons, learning rate, and number of training epochs. Furthermore, since it is common to train a DNN from scratch by initializing the weights randomly (although, of course, from a specific distribution, following e.g. the initialization strategy of Glorot and Bengio [5] or He et al. [8]), the performance of a neural network might even be dependent on the random seed of this weight initialization step. This might also hold for x-vectors, even if they are used only for feature extraction, followed by a machine learning method (e.g. SVM). Perhaps because x-vectors are a relatively recent technique, earlier studies did not consider this stochastic behaviour as a potential source of suboptimal classification performance. In fact, we found no study at all that investigated the effect of randomness for the x-vector encodings.

Our study has two key results. Firstly, we demonstrate experimentally that paralinguistic classification is indeed adversely affected by the random noise introduced by the x-vector representation. Then, in the second step, we also demonstrate that by training an *ensemble* learning method (by repeating the x-vector DNN training process several times), we can make the utterance-level prediction process more robust, leading to notable improvements in the performance on the test set.

2 The Dusseldorf Sleepy Language Corpus

We performed our experiments on the SLEEP (Dusseldorf Sleepy Language) Corpus. It was created by the Institute of Psychophysiology, Duesseldorf, and the Institute of Safety Technology, University of Wuppertal, Germany. The corpus comprises the recordings of 915 German speakers (364 females and 551 males), from 12 to 84 years of age (mean age was 27.6 years). The subjects were asked to

read passages and to speak about specific topics, such as their last weekend or to describe a picture, which resulted in spontaneous narrative speech. It contains 5564, 5328 and 5570 utterances, training, development and test sets, respectively; all three subsets contain recordings of slightly less than six hours, leading to 17 h and 35 min of speech overall. After recording, the utterances were converted to a 16 kHz sampling rate with a quantisation of 16 bits.

The degree of sleepiness of the subjects was assessed using the Karolinska Sleepiness Scale (KSS, [22]). Each subject reported their sleepiness level on the Karolinska Sleepiness Scale (KSS): from 1 (extremely alert) to 9 (very sleepy). At the same time, two observers assigned posthoc observer KSS ratings. The average of both scores was the reference sleepiness value [18]. Later, this corpus was included in the Interspeech Computational Parainguistic Challenge (ComParE) in 2019 [18]. Studies using this corpus found that, instead of opting for classification, it is beneficial to treat this task as a regression one, and round the predictions to integer values on the scale $1, \ldots, 9$ later (see e.g. [6,18,26,27]). We will follow the same strategy in our experiments.

In the ComParE Challenge, the participants applied several techniques like attention networks and adversarial augmentation [27], end-to-end CNNs [4] and Fisher vectors [6,26]. The performance of the methods was measured with Spearman's Correlation Coefficient (CC); the results lay in the range of 0.290 and 0.373 for the development set, and between 0.325 and 0.387 on the test set. Most of the better-performing approaches were combinations of two or more methods.

3 X-Vector Embeddings

The x-vector approach is a neural network-based feature extraction method that provides fixed-dimensional embeddings for variable-length utterances. This system can be viewed as a feed-forward Deep Neural Network that computes such embeddings.

3.1 DNN Architecture

The lower, *frame-level* layers of the network have a time-delay architecture. Following the frame-level layers, the *statistics pooling* layer gets the frame-level activations of the last frame-level layer, aggregates over the input segment, and computes the mean and the standard deviation. These vectors are concatenated and used as input for the next, *segment-level* layer, which is followed by one (or possibly more) additional segment-level layers. The *x-vectors* embeddings can be extracted from any of *segment* layers [23,24]. Instead of predicting frames, the DNN is trained to predict speakers from variable-length utterances. Namely, it is trained to classify speakers present in the train set utilizing a multi-class cross entropy objective function (for more details, see [23]). Therefore, the output softmax layer has as many neurons as there are speakers in the training set. Notice that, to calculate the embeddings, this output layer is not required any more, so it can be discarded after training.

The (utterance-level) embeddings produced by this network capture information from the speakers over the whole audio-signal. These embeddings are called *x-vectors* and they can be extracted from any *segment* layer.

4 Ensemble X-Vectors

Next, we introduce ensemble learning in general, and then we describe the proposed 'Ensemble x-vectors approach' in detail.

4.1 Ensemble Learning

The basic principle of ensemble learning is to train several different, but similar machine learning models, and combine their outputs in some way. Perhaps the best-known such techniques are *bagging* and *boosting*. Bagging carries out the training of such similar models by randomly selecting *subsets* of the training data [1]. Boosting, in contrast, trains the next individual classifier model by focusing on training instances which were mis-classified by previous models (e.g. by using larger weights for these examples [17]). *Stacking*, another ensemble learning technique, is basically a two-step learning scheme, where the outputs of different classifier models (e.g. different algorithms) are combined via another machine learning method [25].

4.2 The Ensemble X-Vector Model

In this study we propose to build an ensemble model based on the x-vector feature extractors. Notice that this approach differs substantially from the above-listed ensemble approaches in the sense that those trained the classification models on the same features; the difference between the models came from a different training subset selection or from utilizing a different machine learning technique. In contrast, we seek to train our classifier or regressor models on the whole training data, and on similar (albeit different) features.

That is, in this study we propose training several x-vector neural network models on the same data, but each time applying a different random seed. By calculating the embeddings with each of them, we get a number of different representations of the same training data. Although in theory concatenating these feature vectors and training only one classifier model might lead to a more robust performance than relying on any of the individual representations, we would end up with unfeasibly huge feature vectors. Therefore we chose to train separate machine learning (e.g. SVR) models on these x-vector representations in the next step. To make the predictions more robust (and hence, make hyperparameter selection more reliable), we suggest simply averaging out the predictions scores got after evaluation in an unweighted manner. Formally, we calculate the posterior estimate provided by the ensemble model as

$$f_e(X) = \frac{1}{m}\sum_{j=1}^{m} f_j(X) = \frac{1}{m}\sum_{j=1}^{m} f_j(H^j), \tag{1}$$

where X denotes the frame-level feature sequence of the actual utterance, H^j is the utterance-level representation (i.e. embedding) of X calculated by the jth x-vector model, and the f_j value is the individual prediction provided by the jth SVR model. According to our hypothesis, the unweighted average of the predictions should improve the robustness of the combined model, provided that the predictions of the individual models are noisy. We call this approach the 'Ensemble x-vector approach'. In our experiments, the number of models in the ensemble (m) was set to 10.

5 Experimental Setup

Next, we describe our experimental setup: how our x-vectors were trained, how we trained our Support Vector Regression methods, and how we evaluated model performance.

5.1 X-Vector Training

Following the results of our previous experiments, we trained our x-vector DNN models (i.e. extractors) on the combined training and development sets of the SLEEP corpus (10892 utterances, 11 h and 39 mins). We employed the Kaldi framework [16] to do this. The *segment6* layer of the DNN is used to compute the 512-dimensional neural network embeddings (i.e. *x-vectors*). As is common in x-vector extraction, we used 23 MFCCs and log-energy as frame-level features; these were also extracted by Kaldi. Although it is standard practice to employ additive noise and reverberation both to increase training data size and to improve the noise robustness of the model, in our earlier experimental results we found that for this particular corpus this process does not assist regression performance [3]; therefore, in our actual experiments we did not employ these techniques during x-vector training.

5.2 Regression and Evaluation

Support Vector Regression (SVR) was used to estimate the degree of sleepiness of the speakers. DNN embeddings were standardized by removing the mean and scaling to unit variance before training the model; transformation parameters were set on the training set. We relied on the scikit-learn implementation [15] with a linear kernel (nu-SVR method); the C complexity parameter was set in the range 10^{-6}, ..., 10^{1}, based on the performance on the development set; we trained a new SVR model with the best complexity value on the training and development sets combined to obtain predictions for the test set. Before rounding to the nearest integer in the $1 \ldots 9$ scale, first we linearly transformed the predictions to have the same mean and standard deviation as those of the labels of the training set; transformation parameters were set on the development set. Since no parameter setting of this transformation involved the test set, and in the end all scores were integers in the range $1 \ldots 9$, the presented results are directly comparable to those found in the literature (e.g. [4,26,27]).

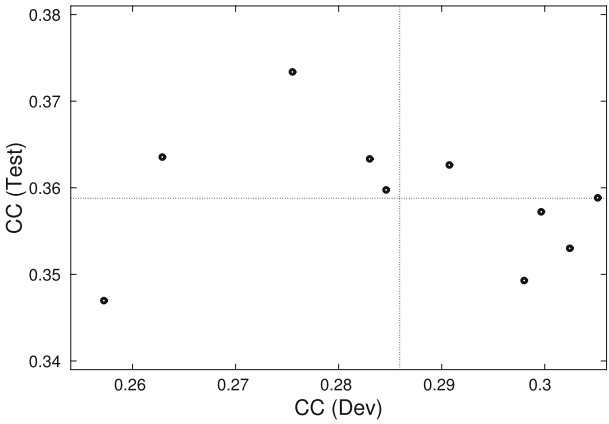

Fig. 1. Spearman's Correlation Coefficients of the individual x-vector models on the development and on the test set; the dotted lines show the average performance of the ten models.

6 Experimental Results

6.1 Model Stochasticity

First, we focused on measuring the amount of stochasticity of predictions when using x-vectors as features. That is, we experimented with training 10 individual x-vector models with the same parameters (i.e. following Sect. 5.1), only with different random seeds used during DNN weight initialization. SVR complexity C was set individually, based on development set performance (although it turned out to be 10^{-4} in all ten cases). Figure 1 shows the measured Spearman's Correlation Coefficients for the ten models for the development and for the test set. (The average CCs of the ten models are shown as dotted lines.) Note that the CC values for the test set are most likely higher than those for the development set because more data used to train the SVR models (as test predictions were obtained by training on both the training and the development sets).

We can also clearly see that the performance on the two sets is only loosely related: the model with the highest performance on the development set just achieved average scores on the test set, while the x-vector model which led to the highest CC on the test set had a CC score below average on dev. As expected, the scores also have a low correlation value (-0.130), which also indicates only a slight (practically none) connection between the performance of the models on the different sets.

Table 1 lists the exact value of some more important cases. This summarizes our previous findings: using just the x-vector model with the first random seed, which is standard practice when employing x-vectors (case 'x-vectors, single'), gave a good performance on the development set (CC = 0.300), but on the test set it scored slightly below average (CC = 0.357). The SVR model with the best performance on the development set produced an average performance on the

Table 1. Spearman's CC scores obtained for some more important x-vector-based approaches.

Regression approach	Correlation	
	Dev	Test
x-vectors, single	0.300	0.357
x-vectors, average	0.286	0.359
x-vectors, best (dev)	0.305	0.359
x-vectors, best (test)	0.276	0.373
x-vectors, ensemble (proposed)	0.298	0.370

test set (CC = 0.359), while the SVR model with the best test set performance (CC = 0.373) had a quite low (and also sub-average) Spearman's correlation coefficient on the development set (0.276).

We would like to emphasize that the differences among the ten models (that is, 0.257 . . . 0.305 and 0.347 . . . 0.373, development and test sets, respectively) are usually viewed as significant on this particular corpus. In our opinion, these experimental results indicate that the x-vectors are sensitive to random DNN weight initialization, and that this stochastic behaviour affects the subsequent, classification or regression step as well. In our opinion, these differences also justify our approach for building an ensemble of the x-vector models, as we can expect the combined model (following Sect. 4.2) to be more robust than the individual x-vector feature extractors.

6.2 Ensemble X-Vectors

Table 1 also shows the CC values for the proposed ensemble x-vector approach. We can readily see that, by training independent SVR models on the ten different x-vector representations, we obtained Spearman's Correlation Coefficient values that exceed the average CCs of the individual models, both on the development and on the test sets. On the test set the proposed method was actually better than 9 out of the 10 models, while its predictions were better than 6 models on the development set. Interestingly though, among the four individual SVR models which were able to match or surpass the performance of the ensemble on the development set, none of them could exceed even the average Correlation Coefficient value of the ten models on the test set.

By using this approach, we improved on our previous results, where we used a single x-vector model for feature extraction [3]. Therefore, the Spearman's correlation coefficient score of 0.370 achieved by the ensemble x-vector model is the highest value which was obtained via a standalone (single) method for this particular task, and it exceeds most studies which employed some kind of fusion as well (with the sole exception of [6]). Of course, the predictions of the ensemble x-vector model could have been combined with some other methods (e.g. ComParE functionals, Bag-of-Audio-Words, Fisher vectors etc.), but this was outside the scope of the current study.

7 Conclusions and Discussion

In this study we focused on the task of sleepiness detection from the speech of the subjects. To do this, we used the public Dusseldorf Sleepy Language Corpus, which contains the speech of 915 subjects, and the ratings of their sleepiness on the Karolinska Sleepiness Scale, on the range $1, \ldots, 9$. Following our previous study, we employed Support Vector Regression (SVR) and used x-vectors as features. For this, we trained custom x-vector extractor models on the training set of the Dusseldorf Sleepy Language Corpus.

X-vectors are extracted by a Deep Neural Network with a special structure, where a pooling layer allows the mapping of variable-length utterances into a fixed-dimensional feature space; the x-vector embeddings are the activations of a specific layer in the network. Of course, like all neural networks, x-vectors might prove to be sensitive to various training meta-parameters. In this study our first research question was whether they are sensitive to the *random seed* used at the weight initialization step. In particular, we were interested in the differences we might find in the performance of the classifier (or in this case, regressor) Support Vector model, which uses the x-vector embeddings directly as features.

To this end, we trained ten x-vector extractor DNNs, which just differed in their random seed, and then trained individual SVR models on each of them. In our first experiment we found that the measured Spearman's Correlation Coefficients differed significantly: they appeared in the range $0.257 \ldots 0.305$ for the development set and $0.347 \ldots 0.373$ for the test set. More importantly, the CC values for the two sets were pretty much independent (we measured a correlation coefficient of -0.130 for the two metric vectors). Then, in our second experiment we built an ensemble x-vector classifier by taking the average of the predictions of the ten SVR models. According to our experimental results, the ensemble was able to notably outperform the average performance of the individual models on the development set and, more importantly, on the test set as well.

Of course, there were several approaches which gave similar (or even higher) CC values on the development set as the proposed ensemble x-vectors did; besides other studies (e.g. [4,6,26], such an approach is our "baseline", the 'x-vectors, single' approach in Table 1). However, we would like to stress that the role of the development set is to support *model selection*; that is, to help to find a machine learning model that produces good-quality predictions on the unseen examples (which is simulated by the elements of the test set). Therefore, even if the ensemble had just produced average CC scores on the development set, we would consider it as a useful approach, because it reduces model stochasticity (and hence it improves model robustness). In our opinion, this would be advantageous even if it would only gave slight improvements on the test set (or even none at all), compared to the average classifier model. Of course, in our case the ensemble x-vector approach led to an improvement; overall, we achieved a Spearman's Correlation Coefficient of 0.370 on the test set, which is the highest value reported so far that was obtained with a standalone method on this corpus.

Acknowledgements. This research was supported by the NRDI Office of the Hungarian Ministry of Innovation and Technology (grants no. K-124413, NKFIH-1279-2/2020 and TKP2021-NVA-09), and within the framework of the Artificial Intelligence National Laboratory Program (MILAB, RRF-2.3.1-21-2022-00004). G. Gosztolya was also funded by the János Bolyai Scholarship of the Hungarian Academy of Sciences and by the Hungarian Ministry of Innovation and Technology New National Excellence Program ÚNKP-21-5-SZTE.

References

1. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
2. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2011)
3. Egas-López, J.V., Gosztolya, G.: Deep Neural Network embeddings for the estimation of the degree of sleepiness. In: Proceedings of ICASSP, Toronto, Canada, June 2021 (2021, accepted)
4. Fritsch, J., Dubagunta, S., Magimai-Doss, M.: Estimating the degree of sleepiness by integrating articulatory feature knowledge in raw waveform based CNNs. In: Proceedings of ICASSP, pp. 6534–6538 (2020)
5. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Machine Learning Research, pp. 249–256 (2010)
6. Gosztolya, G.: Using Fisher Vector and Bag-of-Audio-Words representations to identify Styrian dialects, sleepiness, baby & orca sounds. In: Proceedings of Interspeech, Graz, Austria, pp. 2413–2417, September 2019
7. Grzybowska, J., Kacprzak, S.: Speaker age classification and regression using i-vectors. In: Proceedings of Interspeech, San Francisco, CA, pp. 1402–1406, September 2016
8. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of ICCV, Santiago, Chile, pp. 1026–1034, December 2015
9. Huckvale, M., Beke, A., Ikushima, M.: Prediction of sleepiness ratings from voice by man and machine. In: Proceedings of Interspeech, Shanghai, China, pp. 4571–4575, October 2020
10. Jeancolas, L., et al.: X-vectors: new quantitative biomarkers for early Parkinson's Disease detection from speech. arXiv preprint arXiv:2007.03599 (2020)
11. Johns, M.: Daytime sleepiness, snoring, and obstructive sleep apnea: the Epworth Sleepiness Scale. Chest **103**(1), 30–36 (1993)
12. Murray, B.: A practical approach to Excessive Daytime Sleepiness: a focused review. Can. Respir. J. **2016**, 4215938 (2016)
13. Pagel, J.: Excessive daytime sleepiness. Am. Fam. Phys. **79**(5), 391–396 (2009)
14. Pappagari, R., Wang, T., Villalba, J., Chen, N., Dehak, N.: X-vectors meet emotions: a study on dependencies between emotion and speaker verification. In: Proceedings of ICASSP, Barcelona, Spain, pp. 7169–7173, May 2020
15. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
16. Povey, D., et al.: The Kaldi speech recognition toolkit. In: Proceedings of ASRU, Big Island, HI, USA, December 2011
17. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. **37**(3), 297–336 (1999)

18. Schuller, B.W., et al.: The INTERSPEECH 2019 computational paralinguistics challenge: styrian dialects, continuous sleepiness, baby sounds & orca activity. In: Proceedings of Interspeech, Graz, Austria, pp. 2378–2382, September 2019
19. Schuller, B.W., et al.: The interspeech 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. In: Proceedings of Interspeech, Lyon, France, pp. 148–152, September 2013
20. Schwartz, J.R., Roth, T., Hirshkowitz, M., Wright, K.P., Jr.: Recognition and management of excessive sleepiness in the primary care setting. Prim. Care Companion J. Clin. Psychiatry **11**(5), 197 (2009)
21. Segbroeck, M.V., et al.: Classification of cognitive load from speech using an i-vector framework. In: Proceedings of Interspeech, Singapore, pp. 751–755, September 2014
22. Shahid, A., Wilkinson, K., Marcu, S., Shapiro, C.M.: Karolinska sleepiness scale (KSS). In: Shahid, A., Wilkinson, K., Marcu, S., Shapiro, C. (eds.) STOP, THAT and One Hundred Other Sleep Scales, pp. 209–210. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-9893-4_47
23. Snyder, D., Garcia-Romero, D., Povey, D., Khudanpur, S.: Deep Neural Network embeddings for text-independent speaker verification. In: Proceedings of Interspeech, Stockholm, Sweden, pp. 999–1003, August 2017
24. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: robust DNN embeddings for speaker verification. In: Proceedings of ICASSP, Calgary, Canada, pp. 5329–5333, September 2018
25. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
26. Wu, H., Wang, W., Li, M.: The DKU-LENOVO systems for the INTERSPEECH 2019 computational paralinguistic challenge. In: Proceedings of Interspeech, Graz, Austria, pp. 2433–2437, September 2019
27. Yeh, S., et al.: Using Attention Networks and adversarial augmentation for Styrian dialect, continuous sleepiness and baby sound recognition. In: Proceedings of Interspeech, Graz, Austria, pp. 2398–2402, September 2019
28. Zargarbashi, S., Babaali, B.: A multi-modal feature embedding approach to diagnose Alzheimer's disease from spoken language. arXiv preprint arXiv:1910.00330 (2019)