Variational Autoencoders with implicit priors for short-duration text-independent speaker verification

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, we exploited different strategies to provide prior knowledge to commonly used generative modeling approaches aiming to obtain speaker-dependent low dimensional representations from short-duration segments of speech data, making use of available information of speaker identities. Namely, convolutional variational autoencoders are employed, and statistics of its learned posterior distribution are used as low dimensional representations of fixed length short-duration utterances. In order to enforce speaker dependency in the latent layer, we introduced a variation of the commonly used prior within the variational autoencoders framework, i.e. the model is simultaneously trained for reconstruction of inputs along with a discriminative task performed on top of latent layers outputs. The effectiveness of both triplet loss minimization and speaker recognition are evaluated as implicit priors on the challenging cross-language NIST SRE 2016 setting and compared against fully supervised and unsupervised baselines.

1 Introduction

Variational autoencoders (VAEs) (1; 2) have been introduced as an effective framework within the context of generative models that support tractable approximate inference (3), leveraging neural networks both for generative modeling as well as for approximate inference, usually employing a non-informative prior. However, follow-up works have shown that too simplistic of a prior will in general lead to also simplistic posteriors which might not encode relevant information about the inputs.

Attempts to overcome the above mentioned issue include adversarial autoencoders, proposed originally in (4), which employ an adversarial game on top of latent variables. The discriminator tries to distinguish samples from the posterior and prior distributions, while the encoder tries to produce samples that are indistinguishable from the prior. Moreover, stochastic variational methods (5; 6) appeared as an alternative in which informative data-dependent priors can be used. Sampling methods are employed to estimate gradients of the variational gap, such that any prior from which one can sample can be used. In both of the described cases, the only requirement for a prior is the possibility of efficiently sampling from it.

Even though aforementioned adversarial autoencoders and stochastic variational methods allow the use of non-trivial priors, designing prior distributions which yield desired properties on the finally learned posterior is a challenging task in itself. In this work, rather than explicitly matching posterior and prior distributions, we evaluate the effectiveness of enforcing relevant properties on the posterior distribution by introducing auxiliary discriminative tasks at train time, making use of available labels. By doing so, we argue prior knowledge is introduced implicitly, since desired properties are directly enforced into the posterior distribution.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

The remainder of this paper is organized as follows: Section 2 includes a brief description of the VAE framework along with a brief definition of the speaker verification problem, which we employ as a test-bed for the proposed approach. Section 3 details the strategy we proposed in order to enforce desired properties within the VAE's learned posterior. In Section 4 we evaluate our method, and finally draw conclusions along with future directions in Section 5.

2 Background: Variational Autoencoders and Speaker Verification

Consider $p(X, Z)$, where X is the observed data and Z is the latent representation. The posterior distribution $p(Z|X)$ can be approximated within the family of distributions $q(Z|X, \lambda)$, parametrized by λ. The so-called variational gap has to be minimized in order to give the maximum likelihood estimate of λ. The variational gap is defined as the Kullback-Leibler divergence between the approximate $q(Z|X, \lambda)$ and the true posterior over Z, $p(Z|X)$, written as $\text{KL}(q(Z|X, \lambda)||p(Z|X))$.

A common approach to minimize $\text{KL}(q_\lambda(Z|X)||p(Z|X))$ with respect to λ is to define the Evidence Lower Bound (ELBO) given by:

$$\text{ELBO}(\lambda) = \log(p(X)) - \text{KL}(q(Z|X, \lambda)||p(Z|X)), \tag{1}$$

whose terms can be rearranged, and ELBO can be simplified to:

$$\text{ELBO}(\lambda) = \mathbb{E}_q[\log p(X|Z)] - \text{KL}(\log q(Z|X, \lambda)||p(Z)). \tag{2}$$

Two main components present in above equation are the inference model $q(Z|X, \lambda)$ and the generative model $p(X|Z)$. VAEs parametrize both distributions using neural networks in an encoder/decoder setup. The encoder takes samples from X and outputs the parameters λ of the latent variable model $q_\theta(Z|X)$. The decoder receives samples from Z as input and returns reconstructed data samples from $p_\phi(X|Z)$. Parameters θ and ϕ are the weights and biases of the neural networks which are selected to minimize the negative ELBO using stochastic gradient descent. The negative of the ELBO yields the following loss function used for training the neural networks:

$$l(\theta, \phi) = -\mathbb{E}_{q_\theta(z|x)}[\log p_\phi(X|Z)] + \text{KL}(\log q_\theta(Z|X)||p(Z)). \tag{3}$$

First term in above equation is equivalent to maximum likelihood estimation, thus it is in general substituted by a reconstruction loss, while the second term can be seen as a regularizer, which tries to ensure that the approximation follows the prior distribution as much as possible.

The posterior $q_\theta(Z|X)$ is in general assumed to be an uncorrelated Gaussian. In order to train the VAE using stochastic gradient descent, the reparametrization trick (7; 8) is employed allowing gradients computation through the sampling process between encoder and decoder. Hence, the outputs of the encoder network are the statistics of $q_\theta(Z|X)$ and Z - input for the decoder - is ultimately obtained by $Z = \mu(X) + \sigma(X) \cdot \epsilon$, where $\mu(X)$ and $\sigma(X)$ are the encoder's outputs given X, while ϵ is sampled from $\mathcal{N}(0, I)$.

Speaker verification consists of accepting or rejecting a claimed identity by comparing two spoken utterances, the first of these utterances being used for enrollment (produced by the speaker with the target identity) and the second utterance is obtained from the verified speaker (9).

Under the text-independent setting, speaker verification is performed on top of unconstrained spoken phrases of arbitrary length. The added phonetic variability in this scenario represents an extra adverse factor when compared to the session and speaker variabilities, present in the text-dependent case (10). Classical approaches for Automatic Speaker Verification split the problem into two distinct phases: (i) compute low dimensional speaker representations; (ii) perform binary classification on top of pre-computed representation of enrollment and test utterances.

3 Proposed Model

Unlike the ELBO-based loss definition in Equation 3, we evaluate the use of an auxiliary task on top of the posterior statistics $\mu(X)$, with the aim at enforcing a multi-modal posterior with modes depending on given class labels. Our training loss is thus defined by:

$$l(\theta, \phi) = (1-\beta)||X - X'||_2^2 + \beta D(\mu(X), y), \tag{4}$$

where the first term, the mean squared error between the input X and its reconstructed pair X', is the same as in the standard VAE setting, while the second term, $D(\mu(X), y)$, is some discriminative loss which plays the role of the KL term in Equation 3, considering given class labels y. $\beta \in [0, 1]$ is a tunable hyperparameter. $\mu(X)$ is employed as a low-dimensional embedding of inputs for the discriminative auxiliary task. Two distinct choices of $D(\mu(X), y)$ are evaluated here:

1. A soft triplet loss defined by softplus($||d_+ - d_-||_2$), where d_+ and d_- correspond to a distance measure between pairs of embeddings. d is chosen as $d(\mu(X_1), \mu(X_2)) = 1 - \frac{\mu(X_1) \cdot \mu(X_2)}{||\mu(X_1)||_2 ||\mu(X_2)||_2}$, and the second term is the cosine of the smallest angle between $\mu(X_1)$ and $\mu(X_2)$.
2. The sum of triplet loss with a multi-class classification loss, i.e. $\mu(X)$ is linearly projected into an output layer and cross-entropy loss is measured using available labels.

We evaluate the described setting on the speaker verification task. *RMSProp* is employed for optimization with α set to 0.99. The global learning rate starts at 0.001 and is halved once triplet loss, measured on a validation set held out of training, plateaus for 30 epochs. Training is carried out in a single Titan X NVIDIA GPU, with minibatches of size 64. Minibatches are contructed such that two random segments of different utterances belonging to the same speaker are sampled to form same class pairs (positive), and a random sample from a different speaker is selected to compose the different classes pair (negative). β was at 0.8 for all experiments.

4 Experimental Setup and Results

Evaluation is performed on top of the cross-language NIST SRE 2016 setting (11). Test data in Tagalog and Cantonese are available, while train data is in English. Embeddings obtained with a standard VAE, along with our two proposed strategies using two distinct $D(\mu(X), y)$ previously described choices are compared with x-vectors, a fully-supervised approach shown to outperform i-vectors (12) in the full-recording setting (13). Train data is composed of: *Switchboard*-2, phases 1, 2, and 3, along with *NIST SRE*s from 2004 to 2010 combined with *Mixer* 6, which sums up to approximately 7000 speakers, out of which we remove all the recordings of 50 speakers to be used as validation set. Training is performed on top of 40-dimensional log-mel filter banks. Only the SRE portion is used for training probabilistic linear discriminant analysis (PLDA) (14), which was used as a backend at evaluation phase. Since our model requires fixed size inputs, speech segments of 256 frames were randomly selected from each recording at train time. We augment the described train dataset following the approach in (13), i.e. with additive background noise from the MUSAN corpus and reverberation by convolving room impulse responses (RIR) with original audio data (MUSAN and RIR are available at `www.openslr.org`). We removed silence frames from data using a simple energy-based voice activity detector.

For enrollment, test, and unlabelled (used for PLDA adaptation) data, embeddings of each recording are obtained from 256 frames windows without overlap, and then averaged, such that each test utterance is represented by a single fixed dimensional representation, even though models only have access to short-duration segments.

PLDA was employed as backend after dimensionality reduction of embeddings from 256 to 128, using linear discriminant analysis. PLDA is trained on embeddings from the *SRE* partition of training data, which are computed following the same approach as described for test data for the case of our proposed models, while using the full-recordings in the case of x-vectors. Results in terms of Equal Error Rate (EER) are shown in Table 1 for embeddings obtained from VAEs trained both in a standard fashion, and our proposed approaches.

Table 1: EER obtained for embeddings averaged over short short-duration segments.

	PLDA			*Adapted PLDA*		
	Cantonese	Tagalog	Pooled	Cantonese	Tagalog	Pooled
X-vector	30.91	31.32	31.04	**14.41**	**20.98**	**17.62**
VAE	31.55	32.13	31.83	31.10	32.24	31.66
VAE+Triplet loss	21.81	27.80	24.79	19.89	25.50	22.76
VAE+Cross-entropy	**21.46**	**27.05**	**24.28**	16.50	23.00	20.02

As expected, including speaker identities relevantly increases the discriminability of learned representations when compared to a fully-unsupervised VAE, in both Tagalog and Cantonese evaluations. We further notice that performing speaker recognition on top of statistics of the posterior is more effective than the metric learning approach of triplet loss minimization alone.

In order to overcome the relevant domain shift between train and test data due to different spoken languages, the model adaptation scheme introduced in (15) is utilized for PLDA. To do so, embeddings of unlabelled data in Cantonese and Tagalog are clustered, and clusters are used as speaker identities, which are then employed for training a second PLDA model. The final model is obtained by simply averaging the second order statistics of the two trained models.

Results, as reported in the right section of Table 1, correspond to the evaluation using the adapted PLDA model. Interestingly, one can notice that the higher the *level of supervision* employed on embeddings model training, the higher is the performance gain when adaptation is used. By *level of supervision* we mean how relevant class labels (speaker identities in the studied case) are at train time. Standard VAE makes no use of class labels, while triplet loss employs such information for triplets construction only. Even in the case in which our VAE is trained with cross-entropy minimization, semi-supervised settings can be used, leveraging available unlabelled data, which is not the case for x-vectors, for instance, whose training is performed in a fully-supervised fashion. We thus argue that an increasing level of supervision induces domain-dependent representations, and this is the reason adaptation yields a huge improvement in such cases.

We further evaluate the discriminability of the representations corresponding to the statistics of posterior distributions approximated by VAEs trained in a standard fashion and making use of available speaker identities by plotting 2-dimensional t-SNE embeddings of $\mu(X)$, computed for 10 speakers held out of training. Figures 1, 2, 3 are ordered in increasing *level of supervision*, which once more supports the claim that making use of class labels to perform discriminative tasks on top of statistics of the posterior is an effective strategy to enforce desired properties.

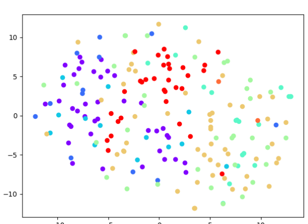

Figure 1: Embeddings obtained from a standard VAE posterior.

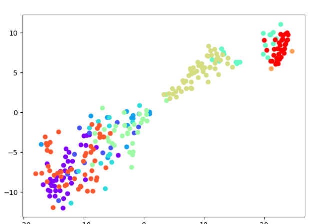

Figure 2: Embeddings obtained from a VAE trained with triplet loss minimization.

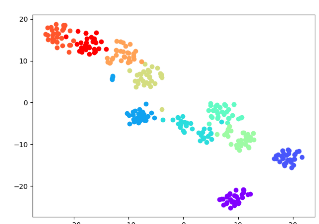

Figure 3: Embeddings from a VAE trained with cross-entropy minimization

5 Conclusion

In this work, we proposed to exchange the divergence term within the variational autoencoders training loss by some discriminative cost, leveraging available class labels. We thus argue such an approach is equivalent to implicitly defining prior distributions, directly inducing desired properties in the learned posterior distribution. Evaluation is performed on the challenging cross-language NIST SRE 2016 evaluation setting, for which we show embeddings obtained by such an approach are speaker-dependent, as enforced by discriminative tasks performed at train time. Future directions include the evaluation of this framework on the semi-supervised setting, employing unlabelled data for training of the generative model, along with labelled data.

References

[1] Diederik P Kingma and Max Welling, "Stochastic gradient vb and the variational auto-encoder," in *Second International Conference on Learning Representations, ICLR*, 2014.

[2] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[3] Alex Lamb, Vincent Dumoulin, and Aaron Courville, "Discriminative regularization for generative models," *arXiv preprint arXiv:1602.03220*, 2016.

[4] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[5] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[6] Rajesh Ranganath, Sean Gerrish, and David Blei, "Black box variational inference," in *Artificial Intelligence and Statistics*, 2014, pp. 814–822.

[7] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] Danilo Jimenez Rezende and Shakir Mohamed, "Variational inference with normalizing flows," *arXiv preprint arXiv:1505.05770*, 2015.

[9] Wei Li, Tianfan Fu, Hanxu You, Jie Zhu, and Ning Chen, "Feature sparsity analysis for i-vector based speaker verification," *Speech Communication*, vol. 80, pp. 60–70, 2016.

[10] Tomi Kinnunen and Haizhou Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[11] "NIST speaker recognition evaluation 2016," 2016, https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation2016/.

[12] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[14] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[15] Daniel Garcia-Romero, Alan McCree, Stephen Shum, Niko Brummer, and Carlos Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2014.

Appendix A - Model architecture

Architectures employed for encoder and decoder are detailed in Tables 2 and 3. Batch normalization is used after all convolution layers. Inputs present dimensionality $[40, 256]$, corresponding to 40 filter banks and 256 frames.

Table 2: Encoder architecture

Layer	**Outputs**	**Kernel size**	**Stride**	**Dilation**	**Activation**
Convolution	19, 84, 128	5, 5	2, 3	1, 2	ELU
Convolution	9, 40, 256	5, 5	2, 2	1, 2	ELU
Convolution	4, 40, 512	5, 5	2, 1	1, 1	ELU
Convolution	1, 40, 1024	5, 5	2, 1	1, 1	ELU
Average Pooling	1, 1, 1024	1, 40	1, 1	-	-
Dense	1024	-	-	-	ELU
Dense	256, 256	-	-	-	ELU, -

Table 3: Decoder architecture

Layer	**Outputs**	**Kernel size**	**Stride**	**Dilation**	**Activation**
Dense	800	-	-	-	ELU
Transpose Convolution	7, 14, 128	3, 4	1, 2	1, 3	ELU
Transpose Convolution	11, 29, 128	3, 4	2, 2	1, 2	ELU
Transpose Convolution	19, 59, 256	3, 4	2, 2	1, 2	ELU
Transpose Convolution	18, 118, 128	4, 6	1, 2	1, 1	ELU
Transpose Convolution	38, 246, 32	4, 12	2, 2	1, 1	ELU
Transpose Convolution	40, 256, 1	5, 13	1, 1	1, 1	-