Independent Components for Acoustic Modeling

Jan Trmal, Jan Vaněk, Luděk Müller, Jan Zelinka

Department of Cybernetics,
University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
{jtrmal, vanekyj, muller, zelinka}@kky.zcu.cz

Abstract

In the paper, we present a comparative study of several methods used nowadays in the field of feature and information extraction. We compared several Independent Component Analysis (ICA) algorithms together with the commonly used Principal Component Analysis (PCA) algorithm in two real-world tasks. The first task was a Voice Activity Detection (VAD), the second is Speaker Verification and Recognition (SVR). The VAD system as well as the SVR system benefited from the ICA decompositions. Moreover, a brief comparison of the information extraction ability is described.

1. Introduction

In a lot of today's applications, a GMM models with diagonal covariance matrices are used. To ensure good performance, various methods of decorrelation are used. In this paper, we try to show, that there are better choices than ordinary PCA or DCT (discrete cosinus transform).

Suppose we have a set $\mathbf{x}$ of k realizations of m-dimensional random variable $\mathbf{X}$, $\mathbf{x} = \{\vec{x}_1, \ldots, \vec{x}_k\}$. Our task is to induce a linear transformation of the random variable $\mathbf{X}$ into n-dimensional random variable $\mathbf{Y}$ given the (training) set $\mathbf{x}$. Every linear transformation consisting of rotation, scaling and reflection of the input data can be described by a $n \times m$ matrix $\mathbf{W}$:

$$\vec{y} = \mathbf{W}\vec{x} \tag{1}$$

so without loss of generality we will talk about the linear transformation $\mathbf{W}$. The resulting random variable $\mathbf{Y}$ is demanded to have some special properties. The properties can be imposed by our a-priori knowledge on the given problem task and, additionally, by the requirements of mathematical methods supposed to operate on the realizations of the random variable.

The following two construction methods of the matrix $\mathbf{W}$ construction are based on the assumption that each element of vector $\vec{x}$ is some linear combination of the elements of the "original" (source) random variable $\vec{y}$, i.e.:

$$\vec{x} = \mathbf{A}\vec{y} \tag{2}$$

Therefore our task is to determine the unmixing matrix $\mathbf{W}$ such as

$$\vec{\hat{y}} = \mathbf{W}\vec{x} \tag{3}$$

where ideally $\vec{\hat{y}} = \vec{y}$ or it is a "good" approximation in general.

2. Principal Component Analysis

The Principal Component Analysis is the classic technique for statistical data analysis and information extraction. Given the set $\mathbf{x}$ we want to find a matrix $\mathbf{W}$, such as the resulting random variable will retain as much information as possible while the dimension of random variable is reduced. The redundancy is measured by the correlations between elements of $\vec{x}$.

The PCA does not impose any limitation on probability density function, only first and second-order statistics must be possible to estimate from the training data (in case where we do not know these statistics a-priori). Suppose we have the correlation matrix

$$\mathbf{C} = \mathrm{E}\{\mathbf{X}\mathbf{X}^{\mathrm{T}}\}$$

then the PCA finds such a matrix W for which the off-diagonal cross-correlations are minimized.

The PCA algorithm consists of several steps. In the first step, the training data are centered (the mean of data is subtracted). Secondly, such a matrix is found which rotates the coordinate system in such a way, which ensures that the correlation between elements of y is the lowest possible one. The contrast function can be written in the form

$$J^{\mathrm{PCA}} = \mathrm{trace}\{\mathbf{C}\} - \sum_{j=1}^{m} w_j^{\mathrm{T}} \mathbf{C}\, w_j \tag{4}$$

where w_j is the j-th row of the matrix $\mathbf{W}$. It is a well-known fact that the solution of the PCA problem is given by terms of eigenvectors and eigenvalues. In the last step, a transformation matrix $\mathbf{W}$ is created from eigenvectors whose eigenvalues are the biggest ones. For more detail, see [11].

3. Independent Component Analysis

As the name suggests, Independent Component Analysis (ICA) methods assume statistical independence among the elements of $\vec{y}$. The functions evaluating degree of independence take higher statistical moments into account, not only the second moments, as PCA does. There exists a wide range of criterion functions – common measures are Entropy, Kullback-Leibler Divergence and Negentropy and their approximations.

3.1. FastICA algorithm

FastICA is based on a fixed-point iterative scheme for finding the maximum of the contrast function. The contrast function is based on approximation of negentropy. Negentropy is defined in the following way

$$J(\mathbf{y}) = \mathrm{H}(\nu) - \mathrm{H}(\mathbf{y}) \tag{5}$$

where the ν is a random variable with (multivariate) normal distribution with the same mean and variance as the variable $\mathbf{y}$ has. Usually, the variance is constrained to unity. The $\mathrm{H}(\vartheta)$ is entropy

of a random variable ϑ. It can be proved, that negentropy is a statistically optimal measure of non-gaussianity.

The evaluation of the negentropy is a very difficult task in general, since the estimation of probability density function must be carried out. Therefore, an approximation of the contrast function is used. Moreover, only scalar case is taken into account – using these approximations we obtain an algorithm for extracting of only one component. However, this is not a significant problem (see later). The approximation has a form

$$J(y) = [\mathrm{E}\{G(y\} - \mathrm{E}\{G(\nu)\}]^2 \tag{6}$$

where $G(\cdot)$ is practically any non-quadratic function. By virtue of choosing function $G(\cdot)$ one can obtain an approximative negentropy with different properties.

By choosing the $G(\cdot)$ function as

$$G(y) = y^4 \tag{7}$$

the kurtosis-based approximation is obtained, i.e. the non-gaussianity is measured by means of kurtosis. However, this approximation is very sensitive to outliers. There exist different and more robust approximations

$$G(y) = \frac{1}{a} \log \cosh ay \tag{8}$$

or

$$G(y) = -\exp\left(-\frac{y^2}{2}\right) \tag{9}$$

where $1 \leq a \leq 2$; often the a is chosen so that $a = 1$.

As already mentioned, by maximizing the scalar value of the contrast function $J(y)$ we can obtain only one (the most non-gaussian) component. However, there exist two different schemes for obtaining m components using the fact, that the weight vectors $\vec{w}_i$ (rows of matrix $\mathbf{W}$) are orthogonal. In general, we apply the algorithm several times and use some orthogonalization process to ensure the orthogonality of extracted components. The *deflationary* scheme extracts the m most independent components in a sequential way. The *symmetric* scheme extract some m independent components without preferring one to another. For more detail on FastICA see [1], for the extraction schemes see [2].

3.2. The Cumulant Based ICA (CuBICA)

This method is based on a description of the data by cumulants. The cumulants-based description is more detailed than the description given by the first four statistical moments (mean, variance, skewness and kurtosis). In fact, cumulants of a given order form a tensor and the diagonal elements of this tensor are moments of the given order. We will assume the order of three and four – i.e. skewness and kurtosis based cumulants. The off-diagonal elements (cross-cumulants) characterize the statistical dependencies between components. If and only if all components are statistically independent, the cross-cumulants will be zero.

Thus, the ICA task can be reformulated by the means of finding some linear transformation matrix diagonalizing the cumulant tensors up to the given order. However in general there does not exists such a diagonal matrix, which would diagonalize the third-order and the fourth-order cumulants simultaneously, so some choice of a different optimization criterion must be made.

The CuBICA algorithm is based on the following criterion

$$\begin{aligned} J(\mathbf{y}) = &\frac{1}{3!} \sum_{\alpha\beta\gamma \neq \alpha\alpha\alpha} \left(C^{(\mathbf{y})}_{\alpha\beta\gamma}\right)^2 \\ &+ \frac{1}{4!} \sum_{\alpha\beta\gamma\delta \neq \alpha\alpha\alpha\alpha} \left(C^{(\mathbf{y})}_{\alpha\beta\gamma\delta}\right)^2 \end{aligned} \tag{10}$$

where the factors $\frac{1}{3!}$ and $\frac{1}{4!}$ arise from an expansion of the Kullback-Leibler divergence in $\mathbf{y}$ i.e. in $\mathbf{Wx}$. This criterion can be significantly simplified using the fact that the square sum over all elements of a cumulant tensor is preserved under any orthogonal transformation $\mathbf{W}$ and due to the multilinearity of the cumulants [4].

The optimization process is based on Givens rotation (rotation around the origin within the plane of two selected components). Thus, the optimization process is transformed to sequential finding the rotation angle which maximizes the independence between two components for all possible component pairs. The final contrast function for optimizing the independence of two components has the form

$$J(\Phi, \mathbf{y}) = A_0 + A_4 \cos(4\Phi + \Phi_4) + A_8 \cos(8\Phi + \Phi_8) \tag{11}$$

where the constants A_0, A_4, Φ_4 and Φ_8 depend only on the cumulants of $\mathbf{y}$ before rotation. Equations of these constants can be found in [3], [4]. In the same paper a modification which does not take into account the third moment is described. The related contrast function can be obtained by setting the third-order cumulants to zero in equations for A_0, A_4, Φ_4 and Φ_8.

4. Speech Data and Experiment Setup

4.1. Speech Corpora Used in Our Experiments

The first experiment was a speaker verification task. We used a special setup of the well-known TIMIT corpus [8]. The TIMIT corpus contains 16 kHz 16 bit sampled recordings from 630 speakers. The recordings are divided into the training and testing parts. The training part contains five sentences for every speaker. The test part consists of three sentences. Each training session was verified with target speaker session and 30 impostor sessions which were randomly chosen. It means $630 + 630 \cdot 30 = 19530$ trials were performed. The verification system performance was measured by the equal error rate (EER).

The second experiment was a simple VAD (voice activity detection). In this experiment we used the Czech high-quality speech corpus [5]. The Czech high-quality speech corpus is a read-speech database consisting of the speech of 100 speakers. Each speaker reads 40 sentences identical for all speakers. The database of text prompts from which the sentences were selected was obtained in an electronic form from the web pages of Czech newspaper publishers [6]. Special consideration was given to the selection of sentences to obtain a representative distribution of frequent triphone sequences (reflecting their relative occurrences in natural speech). Speech corpus was digitized at 44.1 kHz with the resolution of 16 bit per sample. We used MFCC the parametrization method. The dimension of feature vector was 36.

4.2. Speaker Verification System Description

The used SVR system was a text-independent system based on optimized MFCC features and diagonal gaussian mixtures models.

DIM	base	PCA	CuBICA4	CuBICA34	FastICA($_{\text{gauss}}$)	FastICA($_{\text{kurt}}$)	FastICA($_{\text{tanh}}$)
40	0.63 %	0.60 %	0.38 %	0.39 %	0.41 %	0.48 %	0.45 %
37	–	1.08 %	–	–	0.45 %	0.72 %	0.50 %
35	–	1.31 %	–	–	0.32 %	0.81 %	0.61 %
33	–	1.44 %	–	–	0.56 %	1.17 %	0.48 %
30	–	2.22 %	–	–	0.77 %	1.75 %	1.15 %

Table 1: Comparison of EER in the speaker recognition task. CuBICA4 is CuBICA diagonalizing only the fourth moment, CuBICA34 diagonalizes the third as well as the fourth moment. FastICA($_{\text{gauss}}$) uses nonlinearity $G(\cdot)$ from eq. 9, FastICA($_{\text{kurt}}$) uses nonlinearity $G(\cdot)$ from eq. 7, FastICA($_{\text{tanh}}$) uses nonlinearity $G(\cdot)$ from eq. 8

.

The MFCC features were augmented by their delta coefficients. The energy (zeroth) coefficient is discarded so the final dimension of the feature vector was 40. Time sequences of cepstral coefficients were smoothed by a Blackman window yielding better noise robustness. Only the speech segments were chosen by robust voice activity detector. The system is described in more detail in [9].

The acoustic model was a Gaussian mixture model (GMM) and it was trained by combination of the distance based (DB) and the expectation-maximization (EM) algorithm. The DB algorithm has been introduced in [10] for robust GMM training with a little amount of data. We used this clustering algorithm for creating target number of initial gaussian mixtures. This step is very fast in comparison to the classical EM algorithm with iterative addition of the mixtures. Subsequently the EM algorithm was employed on the initial DB-GMM model to estimate more accurate parameters of the final GMM. For the speakers models (λ_{Sp}), 32 mixtures were used. The universal background model (λ_{UBM}) score normalization technique was applied in the verification experiment. The background model was trained in the same way as the speaker models from the data marked as non-speech by the VAD system. The number of mixtures for the background model was 128.

Each trial in the experiment was evaluated according to the expression:

$$L(\hat{\mathbf{Y}}) = \log p(\hat{\mathbf{Y}}|\lambda_{Sp}) - \log p(\hat{\mathbf{Y}}|\lambda_{UBM}) \qquad (12)$$

where $\hat{\mathbf{Y}}$ are the preprocessed testing data, λ_{Sp} and λ_{UBM} are GMM models. $L(\hat{\mathbf{Y}})$ is the resulted score vector. The final verification probability $\mathcal{R} \in < 0, 1 >$ was computed as $\mathcal{R} = \frac{N_p}{N_{tot}}$, where N_p is the number of the data vectors which score is greater then threshold (in our case zero) and N_{tot} is the total number of the tested data vectors. The Detection Error Tradeoff (DET) curve and the EER could be computed after processing all trials in the experiment phase.

4.3. Voice Activity Detection System Description

The VAD system was based on a simple HMM-based acoustic model. The acoustic model comprised of only two models: the model of silence and the model of speech. Each model permitted generation of exactly one segment of acoustic signal. The output probability density function assigned to each state was approximated by Gaussian Mixture Model (GMM) with diagonal covariance matrices. In our experiment we gradually increased the number of mixtures by one and saved and tested the particular models.

We used an n-gram based pseudo-language model. The model was based on language having only two words: "silence" and "speech". In our experiments we used the zerogram based pseudo-language model and the bigram based pseudo-language model. The bigram model is more complicated and uses four transition probabilities. All the transition probabilities were computed from the training part of the data. The system is described in [7]

5. Experimental Results

In both of the two tests we used the original matlab routines created by the authors of the methods (for FastICA package, see `http://www.cis.hut.fi/projects/ica/fastica/`, for CuBICA see `http://itb.biologie.hu-berlin.de/~blaschke/code.html`).

5.1. Voice Activity Detection Task

In the first experiment, we tested the benefits of ICA transformation and the pseudo-language model influence on HMM classifier performance in VAD task. Firstly, the type of G function was chosen. We chose $G(u) = \frac{1}{4}u^4$. Using this function, we obtained a kurtosis-based approximation of negentropy.

The complete training data from the corpus described above were used to estimate the unmixing matrix $\mathbf{W}$. The resulting matrix was used for preprocessing both the training as well as the testing data in the following way: firstly, for every speech record we take all it's observation vectors and form an $m \times T$ matrix $\mathbf{X}$, where m is the dimension of the observation vector (36 in our case) and T is the length of a given record (the number of observed vectors).

The decomposition was performed in the following way: suppose that $\mathbf{x}_t$ is the t-th column of the matrix $\mathbf{X}$, and μ is the mean vector of the m observation vector $\mathbf{X}$, then the estimated source components are

$$\forall t \in \{0, 1, \ldots, T\} : \hat{\mathbf{y}}_t = \mathbf{W}\left(\mathbf{x}_t - \mu\right) + \mathbf{W}\mu \qquad (13)$$

We briefly tested other mentioned ICA packages, but their performance was very similar to the above mentioned. For the HMM based silence detector the classifier trained on ICA transformed data outperformed the HMM classifier trained on "plain" data. Moreover, the pseudo-language modeling was also found to be beneficial.

5.2. Speaker Verification and Recognition

For each speaker in the training set one unmixing matrix $\mathbf{W}$ was calculated and a GMM model was trained on the independent components. The same process was performed for the background data.

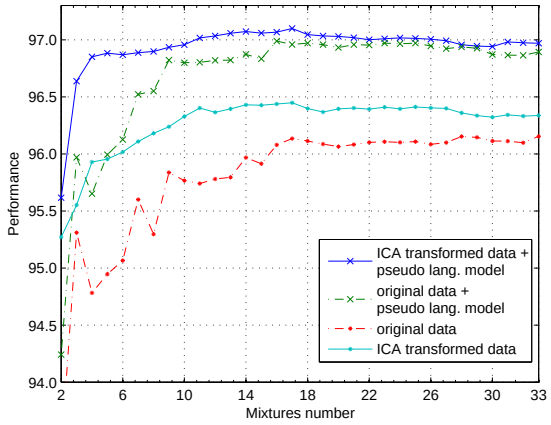

Figure 1: Comparison of the influence of ICA on the performance of HMM-based VAD system

The testing phase is straightforward: for each speaker we unmix the input data (feature vectors $\mathbf{x}$) using the unmixing matrix of the given speaker. Then the log-likelihood $\log p(\hat{\mathbf{Y}}|\lambda_{Sp})$ is performed.

Since the unmixing matrix is different for every speaker, the calculated $\log p'(\hat{\mathbf{Y}}|\lambda_{Sp})$ are not directly comparable. Therefore the probability $\log p(\hat{\mathbf{Y}}|\lambda_{Sp})$ of every speaker and of the background model is normalized by the corresponding unmixing matrix $\mathbf{W}$ to obtain the log-likelihood used in the SVR engine:

$$\log p(\hat{\mathbf{Y}}|\lambda_{Sp}) = \log p'(\hat{\mathbf{Y}}|\lambda_{Sp}) + 2\log\left[\det(\mathbf{W}\mathbf{W}^{\mathrm{T}})\right] \quad (14)$$

where the $2\log\left[\det(\mathbf{W}\mathbf{W}^{\mathrm{T}})\right]$ is a correction factor transforming the obtained log-likelihood into the original data space, the $p'(\cdot)$ is the probability in the given space of independent components and the $p(\cdot)$ is the probability used in the SVR engine.

Moreover, if the components extraction method allows to choose only the n most "important" components, then we performed an additional measuring the the capability of relevant information extraction. Basically, we tested the FastICA algorithm using the deflationary reduction scheme and the classical PCA. However, even in this case, the correction factor always contained the full unmixing matrix corresponding to the given speaker.

The results can be found in Table1. The result show that all systems exploiting ICA algorithms outperformed the baseline SVR system and even the SVR system using the PCA. Talking only about the ICA algorithms, the CuBICA performs better than the "competing" FastICA algorithm. The CubICA4 yields slightly better results, but the difference is statistically insignificant. In the FastICA family, the algorithm using the approximation 9 performed better than the others.

The results of dimensionality reduction tests imply that the FastICA algorithms has good ability to extract relevant information. In some cases, we were able to reduce the number of dimensions from 40 to 30 while keeping the performance comparable to the baseline (40 dimensions).

6. Conclusion

In the paper we demonstrated the possibility to improve the performance of existing speech methods by employing ICA in the preprocessing stage. To take this advantage only a very little work is needed and the computational demands are essentially the same as in the case of PCA. Furthermore, also experiments with extraction of informative features (components) were performed. The test results imply that the ICA algorithms can perform better than the simple PCA. However, due to the diversity of the tested methods, some further tests should be always performed to choose the most suitable ICA algorithm.

7. Acknowledgments

Support for this work was provided by the Grant Agency of Academy of Sciences of the Czech Republic, project No. 1ET101470416.

8. References

[1] Hyvärinen, A: Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. In: IEEE Trans. on Neural Networks. 10(3):626-634, 1999

[2] Hyvärinen, A; Karhunen, J.; Oja, E: Independent Component Analysis. John Willey & Sons, Inc., New York, USA 2001; ISBN 0-471-40540-X

[3] Blaschke, T.: Independent Component Analysis and Slow Feature Analysis: Relations and Combinations. Dissertation thesis; Faculty of Mathematics and Natural Sciences I, Humbold University, Berlin, 2004.

[4] Blaschke, T.; Wiskott, L.: CuBICA: Independent Component Analysis by Simultaneous Third- and Fourth-Order Cumulant Diagonalization. IEEE Transactions on Signal Processing, 52(5):1250-1256, 2004.

[5] Müller, L.; Psutka, J.: Building robust PLP-based acoustic module for ASR applications. In SPECOM 2005 proceedings. Moscow : Moscow State Linguistic University, 2005. s. 761-764. ISBN 5-7452-0110-X.

[6] Radová V., Psutka J.: UWB_S01 Corpus: A Czech Read-Speech Corpus, Proceedings of the 6th International Conference on Spoken Language Processing ICSLP2000, Beijing 2000, China. Volume IV., pp.732-735.

[7] Trmal, J.; Zelinka, J.; Psutka, J. V.; Müller: Comparison between GMM and decision graphs based silence/speech detection method; In SPECOM 2006 proceedings.

[8] Linguistic Data Consortium: TIMIT Acoustic-Phonetic Continuous Speech Corpus, http://www.ldc.upenn.edu, LDC Catalog No.: LDC93S1.

[9] Vaněk, J., Padrta, A.: Introduction of Improved UWB Speaker Verification System, Proc. of Text Speech and Dialogue 2005, Karlovy Vary, Czech Republic, pp. 364-370.

[10] R. D. Zilca, Y. Bistricz: Distance-Based Gaussian Mixture Model for Speaker Recognition over the Telephone, in Proc. ISCLP 2000, Beijing, China, 2000, pp. 1001-1003.

[11] Jackson, J. E.: A User's Guide to Principal Components. John Willey & Sons, Inc., New York, USA 2003; ISBN 0-471-47134-8