Secure Binary Embeddings of Front-end Factor Analysis for Privacy Preserving Speaker Verification

José Portêlo[12]*, Alberto Abad*[1]*, Bhiksha Raj*[3]*, Isabel Trancoso*[12]

[1] INESC-ID Lisboa, Portugal; [2] Instituto Superior Técnico, Lisboa, Portugal
[3] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
{jose.portelo,alberto.abad,isabel.trancoso}@inesc-id.pt, bhiksha@cs.cmu.edu

Abstract

Remote speaker verification services typically rely on the system to have access to the users recordings, or features derived from them, and also a model of the users voice. This conventional scheme raises several privacy concerns. In this work, we address this privacy problem in the context of a speaker verification system using a factor analysis based front-end extractor, the so-called i-vectors. Speaker verification without exposing speaker data is achieved by transforming speaker i-vectors to bit strings in a way that allows the computation of approximate distances, instead of exact ones. The key to the transformation uses a hashing scheme known as Secure Binary Embeddings. Then, a modified SVM kernel permits operating on the i-vector hashes. Experiments on sub-sets of NIST SRE 2008 showed that the secure system yielded similar results as its non-private counterpart.

Index Terms: speaker verification, secure binary embeddings, privacy

1. Introduction

With the increase of storage capability over the Internet, there has been a rapid development of a wide variety of online services ranging from banking to social networks, image and video organizers, blogs, online games, etc. All these services have in common the fact that users must first register with them and receive a unique user ID and password with which they can authenticate themselves to the system. For the password to be secure, it should be a sequence of letters and numbers that is not easily matched to any word in any dictionary, and remembering such passwords can be an onerous task. In many situations, an attractive alternative is to authenticate users by their voice instead. Voice-based interactions are often more natural, particularly in scenarios where the primary mode of interaction with the service is itself through voice or a voice-centric device such as a telephone. Indeed, voice-based speaker authentication systems are becoming increasingly popular as a result.

However, speaker authentication systems have significant privacy concerns. Current speaker authentication systems require access to a person's voice (or at least parameterized versions of it). An individual's voice, while a signature feature of a person, also carries information about their gender, nationality, etc., all of which would also be available for the services to use as they please, *e.g.* they could sell this information. Ideally, therefore, the service should not have access to such information, although it seems nearly paradoxical to require that a system not have access to many of the key voice characteristics that themselves help establish the speaker's identity. Also, the system may itself be hacked, and incoming audio or the voice prints stored on it may be stolen and used to impersonate the user elsewhere.

Therefore, in order to protect the user, both the voice and the voice prints stored on the system must be obfuscated, such that neither the system nor a hacker can extract undesired information from them or use them to impersonate the user. This aspect of speaker authentication has been largely ignored until recently. In [1] homomorphic encryption methods were employed to ensure that the system only sees encrypted data from the user, and only stores encrypted models that it cannot decrypt by itself. To successfully perform speaker authentication, it must engage in "secure multi-party protocols" requiring repeated encryption and decryption, with significant computational participation by the user. In [2] an alternate scheme based on Locality-Sensitive Hashing (LSH) [3] was proposed, that converts voice recordings to a password-like string. Authentication is performed based on perfect matches of the password-like strings derived from the voice to stored templates. Both procedures have their drawbacks – cryptographic methods pose an unacceptably high computational load, while the LSH-based method compromises system accuracy to achieve speed, although it is very secure. A more recent approach was presented in [4], where a new technique called Secure Binary Embeddings (SBE) [5] was successfully used for obtaining not only an almost negligible degradation in classification accuracy (when compared to a baseline using supervectors) but also a computational overhead similar to the one required by LSH.

In this paper we further explore the combination of speaker recognition with SBE for privacy preserving voice authentication. In particular, we aim to analyze if the previous approach in [4] scales properly across different speaker verification techniques and more challenging corpora. Thus, we propose a new privacy-preserving scheme based on factor analysis based front-end extractor, the so-called i-vectors, which is the current *de-facto* standard for speaker verification. Moreover, we carry out experimental analysis on NIST SRE 2008 data, which is considerably more challenging than the one considered so far in previous work. The remaining of this paper is structured as follows. Section 2 briefly describes current speaker verification techniques and in particular, i-vectors based schemes. In Section 3 we describe secure binary embeddings and their guarantees. Experiments illustrating the performance of the proposed secure scheme are reported in Section 4. Finally, we present our conclusions and discuss directions for future work in Section 5.

2. Speaker Verification

In conventional text-independent speaker verification systems, the user provides the system with voice samples during an en-

rollment phase and the system employs these samples to build a "model" for the user. Later, in the verification phase, new incoming speech signals are compared to this model to verify the user identity, usually generating a log-likelihood verification score.

In terms of speech parameterization, speaker verification systems are typically built upon conventional short-term cepstral features, like MFCC or PLP features. Like in other speech applications, feature vectors are usually augmented with their derivatives. As for speaker modeling, the classic approach is the well-known GMM-UBM technique, which consists of obtaining a Gaussian Mixture Model (GMM) for each target speaker based on *maximum-a-posteriori* (MAP) adaptation of the Gaussian means of a Universal Background Model (UBM) [6]. The UBM is a GMM trained with several hours of speech representing the general distribution of speech characteristics from any speaker. In addition to reducing enrollment data requirements, MAP adaptation also ensures a one-to-one correspondence between the Gaussians in the UBM and those in the model for the speaker. Given a new recording purported to be from a target speaker, the log likelihood assigned to it by the GMM for the target speaker is compared to that obtained from the UBM to determine if the speaker must be accepted or not [6].

More recently, variations to this GMM-UBM scheme have resulted in the proliferation of new successful vector-based methods, such as the GMM supervector (GSV) [7] approach and the Joint Factor Analysis (JFA) [8] or Total Variability (TV) [9] based compensation methods. On one hand, the method generally known as GSV consists of mapping each speech utterance to a high-dimensional vector space. An SVM is then used for classification of speaker vectors within this space. The mapping to the high-dimensional space is achieved by stacking all parameters (usually the means) of the adapted speaker GMM (obtained in the same way as in the GMM-UBM scheme) in a single supervector. To verify that a given test recording was indeed spoken by the speaker, the supervector derived from a new recording is classified by the SVM. On the other hand, TV modeling [9] has rapidly emerged as one of the most powerful approaches for speaker verification and has become the current *de-facto* standard. In this approach, closely related to the JFA [8], the speaker and the channel variabilities of the high-dimensional GMM supervector are jointly modeled as a single low-rank total-variability space. The low-dimensionality total variability factors extracted from a given speech segment form a vector, named i-vector, which represents the speech segment in a very compact and efficient way. Since the i-vector comprises both speaker and channel variabilities, in the i-vector framework for speaker verification some sort of channel compensation or channel modeling technique usually follows the i-vector extraction process. Regarding channel compensation, Linear Discriminant Analysis (LDA) or Within-Class Covariance Normalization (WCCN) are typically applied to compensate for channel nuisance in the i-vector space [10]. Then, the verification score can be obtained either based on a simple cosine similarity between the target speaker i-vector and the the test utterance i-vector, or by evaluating the test utterance i-vector with a previously trained SVM. In the latter, cosine kernel is usually preferred. Recently, new channel modeling techniques for i-vectors, such as Probabilistic Linear Discriminant Analysis (PLDA) [11], have been reported to overcome classical cosine-distance scoring of i-vectors with channel compensation.

In this paper we employ the i-vector approach that uses SVMs (trained on target and impostor i-vectors) for speaker modeling and scoring. Thus, we exploit the total-variability modeling as a sort of factor analysis based front-end extractor able to provide compact speaker representations of fixed length. Moreover, we do not consider any of the previously mentioned channel compensation methods, since this issue is out of the scope of the present study. Nevertheless, it would be straightforward incorporating (at least) LDA and WCCN compensation without significant alterations of the proposed scheme.

3. Secure Binary Embeddings (SBE)

A *secure binary embedding* (SBE) is a scheme for converting real-valued vectors to bit sequences using band-quantized random projections. These bit sequences, which we will refer to as *hashes*, possess an interesting property: if the Euclidean distance between two vectors is lower than a threshold, then the Hamming distance between their hashes is proportional to the Euclidean distance between the vectors; if it is higher, then the hashes provide no information about the true distance between the two vectors. This scheme relies on the concept of Universal Quantization [12], which redefines scalar quantization by forcing the quantization function to have non-contiguous quantization regions.

Given an L-dimensional vector $\mathbf{x} \in \mathbb{R}^L$, the universal quantization process converts it to an M-bit binary sequence, where the m-th bit is given by

$$q_m(\mathbf{x}) = Q\left(\frac{\langle \mathbf{x}, \mathbf{a}_m \rangle + w_m}{\Delta}\right) \tag{1}$$

Here $\langle , \rangle$ represents a dot product. $\mathbf{a}_m \in \mathbb{R}^L$ is a projection vector comprising L i.i.d. samples drawn from $\mathcal{N}(\mu = 0, \sigma^2)$, Δ is a precision parameter, and w_m is a random dither drawn from a uniform distribution over $[0, \Delta]$. $Q(\cdot)$ is a quantization function given by $Q(x) = \lfloor x \bmod 2 \rfloor$. We can represent the complete quantization into M bits compactly in vector form:

$$\mathbf{q}(\mathbf{x}) = Q\left(\mathbf{\Delta}^{-1}(\mathbf{A}\mathbf{x} + \mathbf{w})\right) \tag{2}$$

where $\mathbf{q}(\mathbf{x})$ is an M-bit binary vector, which we will refer to as the *hash* of $\mathbf{x}$. $\mathbf{A} \in \mathbb{R}^{M \times L}$ is a matrix composed of the row vectors $\mathbf{a}_m$, $\mathbf{\Delta}$ is a diagonal matrix with entries Δ, and $\mathbf{w} \in \mathbb{R}^M$ is a vector composed from the dither values w_m.

The universal 1-bit quantizer of Equation 1 maps the real line onto $1/0$ in a banded manner, where each band is Δ_m wide. Figure 1 compares conventional scalar 1-bit quantization (left panel) with the equivalent universal 1-bit quantization (right panel).

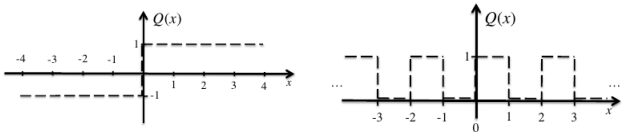

Figure 1: *1-bit quantization functions.*

The binary hash generated by the Universal Quantizer of Equation 2 has the following properties [5]: the probability that the i^{th} bits, $q_i(\mathbf{x})$ and $q_i(\mathbf{x}')$ respectively, of hashes of two vectors $\mathbf{x}$ and $\mathbf{x}'$ are identical depends only on the Euclidean distance $d = \|\mathbf{x} - \mathbf{x}'\|$ between the vectors and not on their actual values. As a consequence, the following relationship can be shown [5]: given any two vectors $\mathbf{x}$ and $\mathbf{x}'$ with a Euclidean distance d, with probability at most $e^{-2t^2 M}$ the normalized (per-bit) Hamming distance $d_H(\mathbf{q}(\mathbf{x}), \mathbf{q}(\mathbf{x}'))$ between the hashes of

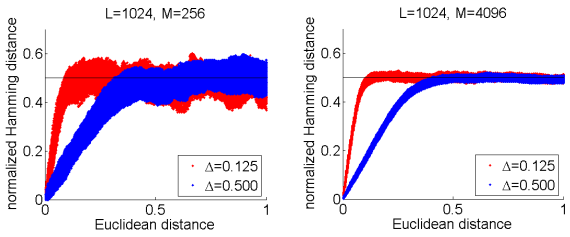

Figure 2: *SBE behavior as a function of* Δ *for two values of* M.

$\mathbf{x}$ and $\mathbf{x}'$ is bounded by:

$$\frac{1}{2}-\frac{1}{2}e^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2}-t \leq d_H(\mathbf{q}(\mathbf{x}),\mathbf{q}(\mathbf{x}')) \leq \frac{1}{2}-\frac{4}{\pi^2}e^{-\left(\frac{\pi\sigma d}{\sqrt{2}\Delta}\right)^2}+t$$

where t is the control factor. The above bound means that the Hamming distance $d_H(\mathbf{q}(\mathbf{x}),\mathbf{q}(\mathbf{x}'))$ is correlated to the Euclidean distance d between the two vectors, if d is lower than a threshold (which depends on Δ). Specifically, for small d, $E[d_H(\mathbf{q}(\mathbf{x}),\mathbf{q}(\mathbf{x}'))]$, the expected Hamming distance, can be shown to be bounded from above by $\sqrt{2\pi^{-1}}\sigma\Delta^{-1}d$, which is linear in d. However, if the distance between $\mathbf{x}$ and $\mathbf{x}'$ is higher than this threshold, $d_H(\mathbf{q}(\mathbf{x}),\mathbf{q}(\mathbf{x}'))$ is bounded by $0.5-4\pi^{-2}exp\left(-0.5\pi^2\sigma^2\Delta^{-2}d^2\right)$, which rapidly converges to 0.5 and effectively gives us no information whatsoever about the true distance between $\mathbf{x}$ and $\mathbf{x}'$.

In order to illustrate how this scheme works, we randomly generated pairs of vectors in a high-dimensional space ($L = 1024$) and plotted the normalized Hamming distance between their hashes against the Euclidean distance between them (Figure 2). The number of bits in the hash is also shown in the figures. In all cases, once the normalized distance exceeds Δ, the Hamming distance between the hashes of two vectors ceases to provide any information about the true distance between the vectors. Changing the value of the parameter Δ allows us to adjust the distance threshold until which the Hamming distance is informative. Increasing the number of bits M leads to a reduction of the variance of the Hamming distance. A converse property of the embeddings is that for all $\mathbf{x}'$ except those that lie within a small radius of any $\mathbf{x}$, $d_H(\mathbf{q}(\mathbf{x}),\mathbf{q}(\mathbf{x}'))$ provides little information about how close $\mathbf{x}'$ is to $\mathbf{x}$. It can be shown that the embedding provides information theoretic security beyond this radius, if the embedding parameters $\mathbf{A}$ and $\mathbf{w}$ are unknown to the potential eavesdropper. Any algorithm attempting to recover a signal $\mathbf{x}$ from its embedding $\mathbf{q}(\mathbf{x})$ or to infer anything about the relationship between two signals sufficiently far apart using only their embeddings will fail to do so.

4. Speaker Verification with SBE

The application of the SBE to speaker verification systems is straightforward: if the classifier could be made to operate on SBE hashes of i-vectors rather than on the i-vectors themselves, speaker verification may be performed without exposing speaker data. In this work we consider non-private reference speaker verification systems that use SVMs for speaker modeling. Then, under the privacy-preserving scheme, SVM kernels must be modified to work with Hamming distances between SBE hashes: $k(\mathbf{x},\mathbf{x}') = e^{-\gamma\cdot d_H^2(\mathbf{q}(\mathbf{x}),\mathbf{q}(\mathbf{x}'))}$. Note that for a given $\mathbf{A}$ and $\mathbf{w}$, the modified kernel closely approximates the conventional RBF for small $d(\mathbf{x},\mathbf{x}')$, but varies significantly from it at larger $d(\mathbf{x},\mathbf{x}')$. While it does not satisfy Mercer's conditions and cannot be considered a true kernel, in practice it is effective as we shall see in the experiments below.

The implementation of a *privacy-preserving* speaker verification system is now as follows: the user communicates with the server through a smartphone or computation-capable device. In the enrollment phase, the i-vectors for both the enrollment recordings and impostor recordings are computed by the user. Impostor recordings may be obtained from any public resource. The user computes SBE hashes from the i-vectors and transmits them to the server. He retains the parameters $\mathbf{A}$ and $\mathbf{w}$ employed by the SBE as his/her private keys. The system trains an SVM with the obtained SBE hashes. During verification, the user computes the SBE hash for the i-vector obtained from the test recording and transmits it to the system, which classifies it.

4.1. Experiments using i-vectors

As a proof of concept, we ran experiments on the NIST Speaker Recognition Evaluation (SRE) 2008 corpus [13]. We focused on two test sets, *8conv-short3* and *8conv-10sec*, and we analyzed the *telephone training and test* condition. The *8conv* subset consists on 8 two-channel telephone conversation excerpts for 635 target speakers, the *short3* and *10sec* subsets consist on two-channel telephone conversations containing approximately 5 minutes and 10 seconds of speech, respectively, for different speakers. In our experiments we used i-vectors based on MFCC features extracted in frames of 25ms, at the rate of 100 frames per second. For each frame we extracted 12 MFCC coefficients and the log-energy, augmenting them with the temporal differences and double-differences to result in a total of 39 features. A UBM composed by 1024 Gaussians was trained using the NIST SRE 2004, 2005 and 2006 corpora. The total variability factor matrix $\mathbf{T}$ was estimated according to [14], also using the same three corpora. The dimension of the total variability subspace was fixed to 400. 10 EM iterations were applied consisting of a first ML estimation followed with minimum divergence update. The covariance matrix was not updated in any of the EM iterations. The estimated $\mathbf{T}$ matrix is used for extraction of the total variability factors of the processing speech segments as described in [14]. Finally, SVM speaker models trained with i-vectors are obtained using the LIBSVM toolkit [15].

We are not interested in and therefore did not perform experiments using supervectors on the NIST SRE 2008 corpus. This decision was made because not only do i-vectors perform better on speaker verification tasks but they also fit better the privacy paradigm, as they are shorter and allow for the possibility of alternative and efficient scoring approaches. Figure 3 and Table 1 show the results obtained using two different kernels.

Table 1: *Speaker verification results, given in terms of EER (%age), when using i-vectors.*

	8conv-short3	*8conv-10sec*
Cosine kernel	5.1	9.4
Euclidean kernel	5.6	9.5

Since we are using SVMs for classifying the i-vectors, the choice of the selected kernels and scoring scheme must be addressed. Thus, we report results using SVMs with both cosine similarity based kernel and Euclidean distance based kernel. We need to consider both of them because although the first is known to be the most adequate for i-vectors modeling, the latter is the one to which the SBE hashes best relate. For obtaining the cosine based kernel, we first normalize the i-vectors, mapping

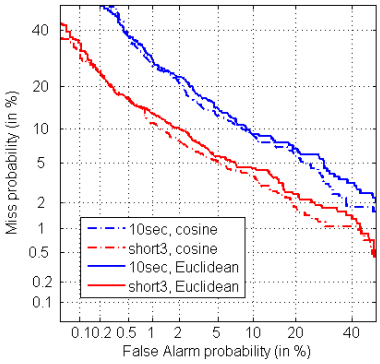

Figure 3: *Speaker verification results, given in terms of DET curves, when using i-vectors.*

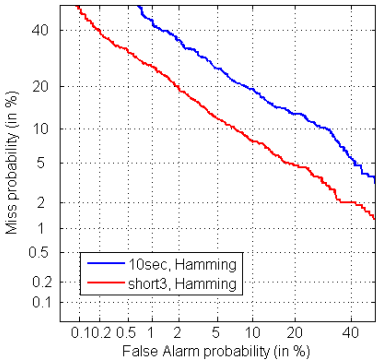

Figure 4: *Speaker verification results, given in terms of DET curves, when using SBE hashes.*

them in the unit circle, and then use a linear kernel. For obtaining the Euclidean based kernel, we use an RBF kernel. Analyzing the results, we can see that only minimal degradation is obtained when considering the Euclidean kernel instead of the cosine one, and therefore it is legitimate to consider the Euclidean distance as an adequate metric for comparing different i-vectors. Note that these baseline results are slightly worse than the ones obtained in the NIST SRE 2008 task with state-of-the-art systems [13]. There are several reasons for this. First, in this work we are only interested on exploiting the total-variability modeling simply as a sort of factor analysis based front-end extractor for providing compact speaker representations of fixed length from which SBE hashes can be obtained. Thus, the problem of channel compensation is out of the scope of the present study and neither additional channel compensation techniques nor any kind of score normalization methods have been considered. Although the effect of channel variability is partially minimized in our experimental framework, since we consider multi-session training conditions and only telephone-telephone trials, it is still responsible for a certain degradation in the results. Also, the limited amount of data used in our experimental framework for training the total-variability matrix further prevented us from achieving better baseline results. The decrease in performance is more visible in the *10sec* set, as the recordings are much shorter and therefore the lack of additional compensation techniques is more noticeable. This, however, does not invalidate any of our results using SBE in the following section, as we are primarily concerned on analyzing how using privacy-preserving SBE hashes compares to a baseline speaker verification system and *not* on improving the current state-of-the-art results for the speaker verification task.

4.2. Experiments using SBE hashes

The secure binary embeddings have two parameters that can be varied: the quantization step size Δ and the number of bits M. Although we performed many experiments using different values for these control parameters, for simplicity we only present the results we obtained for the values corresponding to a high performance trade-off between computation time, the amount of noise on the hashes (controlled by M) and the threshold above which the hashes become uninformative (controlled by Δ). These results are shown in Figure 4 and Table 2.

We report results using the Hamming distance kernel, which is the adequate one to model and score with SBE hashes. We notice that there is a degradation in the speaker verification performance achieved using our proposed privacy-preserving approach when compared with the non-private baseline. This performance drop is consistent with the results already observed in [4]. The reduced performance that emerges by shifting from a non-private speaker verification system to a privacy-preserving one is a small price to pay for keeping a user's identity secret. Nevertheless, it is worth stressing that the NIST SRE 2008 tasks addressed in this work are considerably more challenging and realistic than the one considered in [4]. It must be also taken into consideration that the proposed system is built upon a baseline verification system that could be improved based on some of the strategies described previously. Thus, a more robust non-private baseline could also contribute to improve the private counterpart. Overall, we consider these results promising and a new step forward towards the future development of robust and efficient privacy-preserving speaker authentication.

Table 2: *Speaker verification results, given in terms of EER (%age), when using SBE hashes.*

	8conv-short3	*8conv-10sec*
Hamming kernel	8.8	14.7

5. Conclusions and Future Work

In this paper we presented a new privacy-preserving speaker verification scheme based on the combination of factor analysis front-end with SBE. In spite of the expected degradation due to the protection of user's identity, experimental analysis in two challenging sub-conditions of the NIST SRE 2008 evaluation have demonstrated the feasibility of the proposed approach. In practice, improving the non-private baseline is the first step to minimize the impact of the private approach. Nevertheless, the use of i-vectors for the first time in a privacy-preserving scheme opens new perspectives for future research in this area. In particular, we plan to explore alternative modeling and scoring approaches to current SVM based on a Hamming distance based kernel for i-vector hashes. Thus, we expect to both simplify and improve our privacy-preserving speaker verification system.

6. Acknowledgements

José Portêlo and Isabel Trancoso were supported by FCT grants SFRH/BD/71349/2010, PTDC/EIA-CCO/122542/2010 and PEst-OE/EEI/LA0021/2013. Bhiksha Raj and José Portêlo were supported in part by NSF grant no. 1017256.

7. References

[1] Pathak, M., and Raj, B., "Privacy Preserving Speaker Verification using adapted GMMs", in *Proceedings Interspeech*, pp. 2405–2408, August 2011.

[2] Pathak, M., and Raj, B., "Privacy-Preserving Speaker Verification as Password Matching", in *Proceedings ICASSP*, pp. 1849–1852, March 2012.

[3] Indyk, P., and Motwani, R., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality, in "ACM Symposium on Theory of Computing", pp. 604–613, 1998.

[4] Pathak, M., Portêlo, J., Raj, B., and Trancoso, I., "Privacy-Preserving Speaker Authentication", in *Proc. Information Security Conference (ISC)*, Passau, Germany, September 2012.

[5] Boufounos, P., and Rane, S., "Secure Binary Embeddings for Privacy Preserving Nearest Neighbors", in *Proc. Workshop on Information Forensics and Security (WIFS)*, Foz do Iguaçu, Brazil, December 2011.

[6] Reynolds, D. A., Quatiery, T. F., and Dunn, R. B., "Speaker Verification using Adapted Gaussian Mixture Models", *Digital Signal Processing*, Volume 10, Issues 1-3, pp. 19–41, January 2000.

[7] Campbell, W. M., Sturim, D. E., and Reynolds, D. A., "Support Vector Machines using GMM Supervectors for Speaker Verification", *IEEE Transactions on Audio, Speech, and Language Processing*, 13(5): 308–311, 2006.

[8] Kenny, P., Boulianne, G., Ouellet, P., and Dumouchel, P., "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4): 1435–1447, 2007.

[9] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P., "Front-End Factor Analysis for Speaker Verification", *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4): 788–798, 2011.

[10] Dehak, N., Dehak, R., Kenny, P., Brümmer, N., Ouellet, P. and Dumouchel, P., "Support Vector Machines Versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification", in *Proceedings Interspeech*, pp. 1559–1562, September 2009.

[11] Burget, L., Plchot, O., Cumani, S., Glembek, O., Matejka, P., and Brümmer, N., "Discriminatively Trained Probabilistic Linear Discriminant Analysis for Speaker Verification", in *Proceedings ICASSP*, pp. 4832–4835, May 2011.

[12] Boufounos, P., "Universal Rate-Efficient Scalar Quantization", *IEEE Transactions on Information Theory*, 58(3): 1861–1872, 2012.

[13] –, "The NIST Year 2008 Speaker Recognition Evaluation", Evaluation plan available at `http://www.itl.nist.gov/iad/mig//tests/sre/2008/sre08_evalplan_release4.pdf`; Evaluation results available at `http://www.itl.nist.gov/iad/mig/tests/spk/2008/official_results/index.html`.

[14] Kenny, P., Ouellet, P., Dehak, N., Gupta, V., and Dumouchel, P., "A Study of Inter-Speaker Variability in Speaker Verification", *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5), pp. 980–988, 2008.

[15] C.-C. Chang and C.-J. Lin, "LIBSVM : A Library for Support Vector Machines", *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.