Coding Methods for the NMF Approach to Speech Recognition and Vocabulary Acquisition

Meng SUN, Hugo VAN HAMME
Department of Electrical Engineering-ESAT, Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, Bus 2441, B-3001 Leuven, Belgium

ABSTRACT

This paper aims at improving the accuracy of the non-negative matrix factorization approach to word learning and recognition of spoken utterances. We propose and compare three coding methods to alleviate quantization errors involved in the vector quantization (VQ) of speech spectra: multi-codebooks, soft VQ and adaptive VQ. We evaluate on the task of spotting a vocabulary of 50 keywords in continuous speech. The error rates of multi-codebooks decreased with increasing number of codebooks, but the accuracy leveled off around 5 to 10 codebooks. Soft VQ and adaptive VQ made a better trade-off between the required memory and the accuracy. The best of the proposed methods reduce the error rate to 1.2% from the 1.9% obtained with a single codebook. The coding methods and the model framework may also prove useful for applications such as topic discovery/detection and mining of sequential patterns.

Keywords: vocabulary acquisition, vector quantization, non-negative matrix factorization, histograms of the acoustic co-occurrence, multi-stream feature

1 INTRODUCTION

A novel framework for discovering words in utterances and subsequently recognizing those words in continuous speech was proposed in [1][2]. The approach relies on non-negative matrix factorization (NMF - [3]), an information discovery method that finds additive parts in data. Every utterance is mapped to a Histogram of Acoustic Co-occurrences (HAC), a non-negative representation of speech in which the HAC of an utterance is the weighted sum of the HAC of words. Hence, in the ideal case, the HAC of words can be found with NMF as the additive parts that make up a collection of utterance-level HACs. The HAC-based NMF framework actually models how humans abstract useful patterns from speech and conceptual facts, and is a data-driven learning process that is distinct from the conventional model-driven HMM. NMF is able to discover relations in high-dimensional representations. By concatenating feature representations of multiple modalities, it can find correlations between knowledge sources and therewith establish cross-modal relations. For instance, the acoustic and visual form of "*dog*" can be related, even if embedded in a scene of other acoustic and visual objects. As pointed out in [4] [5], this is relevant to modeling how humans acquire language.

One way of forming the HAC data is by observing the co-occurrence frequencies of prototypical short-term speech spectra. Finding a close prototype for a given spectrum involves vector quantization (VQ). The VQ process forces us to make compromises on the recognition accuracy that can be obtained with the NMF approach. While we have shown in [1] that the approach can produce an accuracy that is comparable to that of discrete density HMMs, the question naturally arises of how to generate accuracies that are comparable to those of continuous density HMMs. An obvious first approach is to increase the codebook size, such that the quantization error can be reduced. However, since the HAC representation used in the NMF model is based on co-occurrence statistics, we estimate that the data requirements would scale quadratically with the codebook size, which is even worse than the linear scaling one observes with discrete density HMMs.

In this paper, we report on our continued efforts to search for alternative representations that mitigate the loss due to quantization errors. In earlier work [6], we exploited the fact that NMF can be used as the learning algorithm in a layered architecture and that it can easily cope with high-dimensional data. This allowed to represent speech as a sum of structures in the time-frequency plane. In this paper, we exploit the property that NMF can easily combine information from multiple streams [2]. We particularly look for ways to encode the spectral information with greater accuracy without increasing the complexity as much as one would by merely increasing the codebook size. Moreover, we keep in mind that our task is "recognition" and does not end at "coding": recognition requires generalization to avoid overlearning.

A first technique that we explore is to use multiple codebooks with different Voronoi regions on the same information stream, a technique which has also been applied to HMMs [7]. This way, the set of spectra that are mapped to the same quantized representation is reduced and hence quantization effects are reduced. However, each of the codebooks can remain small and the data requirements are not increased as we add more codebooks. Admittedly, such a representation is redundant and not optimal from the perspective of efficient coding. However, in this context, where the data requirements scale quadratically with codebook size and where NMF has been shown to be able to exploit information from redundant acoustic sources [2], this is not a major concern.

A second technique that is explored here is soft VQ, i.e. a spectral data vector characterized by its proximity to multiple prototypes. Proximity is measured as the posterior probability of a collection of Gaussians, much like in semi-continuous HMMs [8]. To keep the NMF problem computationally feasible, we require that the data matrix is sufficiently sparse, which translates into the requirements that each spectral vector can be characterized only by its proximity of a "limited" set of prototypes.

A third technique, adaptive VQ, is proposed to increase sparsity and hence to save memory as well as retaining the accuracy of coding with soft VQ. The number of Gaussians used to label each frame is defined adaptively based on their frame likelihoods. The frames near the centroids will use a small number of Gaussians while the ones near the boundaries will get many activations on the nearby Gaussians.

The paper is organized as follows: the frame coding method and the architecture of the NMF model are described in section 2; the experimental results are presented in section 3; the discussion and comparison are in section 4; the conclusion is in section 5.

2 FRAME CODING AND ACOUSTIC CO-OCCURRENCE IN THE NMF MODEL

The basic idea in [2] is applying batch NMF for the acoustic representation of the utterances. With the powerful ability of extracting parts from objects, NMF can find recurring word-like patterns. The diagram of making histogram of the acoustic co-occurrence is shown in Figure 1 and is now explained in more detail.

2.1 Frame Coding and HAC Model

An input utterance is processed as is common in speech recognition by hopping an analysis window (e.g. 20ms length) over the utterance by advancing it over regular *frame* shifts (e.g. 10ms) and computing a short term spectrum which is transformed to Mel Frequency Cepstral Coefficients (MFCCs), yielding a sequence of *static* (S) stream vectors, one per frame. To emphasize the dynamics in speech, the first (*velocity* - V) and second order (*acceleration* - A) difference of this sequence is computed as well. Each of these three HAC-representation streams is quantized by its own codebook. The resulting label sequence is shown in Figure 1 for the static (MFCC) stream. The number of times two codewords of a stream co-occur at a fixed time difference or *lag* is then counted over the utterance. The number of bins in the resulting HAC representation equals the square of the codebook size. When multiple utterances are available, the HAC representations are stacked in a matrix A: one column per utterance.

The VQ makes the representation of co-occurrences symbolic but with less accuracy due to the lossy compression. At first sight, we can decrease the VQ errors by enlarging the codebook size. Unfortunately, this will lead to over training: some codeword pairs may become unobserved in training while observed in the test and vice versa. By introducing the methods of multi-codebooks and soft VQ into the HAC model, we can improve the VQ resolution as well as avoiding overtraining. The adaptive VQ can further save processing time and memory.

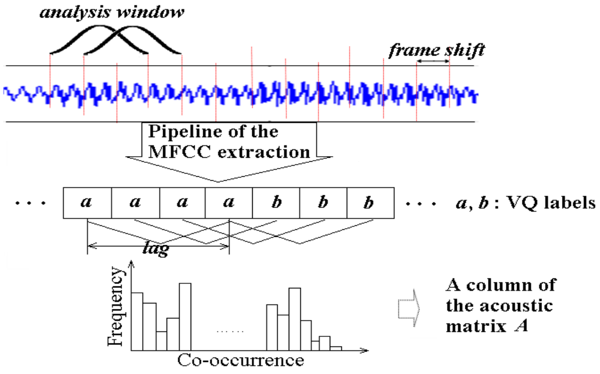

Figure 1: *Diagram of HAC. With the conventional MFCC pipeline, an utterance can be transformed into a sequence of vectors. The vectors are then compressed into a sequence of numbers (VQ labels) by vector quantization. Finally, the codeword co-occurrences are counted and stored and flattened into a column of the data matrix A.*

Multi-codebooks The above process of making a HAC representation for an utterance is identical for the *Static, Velocity,* or *Acceleration* stream, for different *lag*s, for different codebooks or even for different window lengths and frame shifts. We will call the HAC computed by a particular choice of signal analysis parameters and *lag* and applying a given codebook to quantize a stream, a **feature**. So in the above example with 3 streams all with 10ms frame shift and 20 ms window, each quantized with one codebook, but using 3 different *lag* values, there will be 9 features.

The feature matrices made from different *lags*, different streams, and different codebooks, $A_q, i = 1, 2, \ldots, Q$, can be concatenated to get the integrated acoustic feature matrix for the training or the testing set

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_Q \end{bmatrix} \tag{1}$$

where Q is the number of **features**, and each column of A represents an utterance.

All HAC representations share that the observed histogram counts are the sum of co-occurrence frequencies of the words that make up the utterance. By stacking some or all of them into a supervector, we can obtain a more accurate description of the utterance [2].

Soft VQ Soft VQ is making the membership function fuzzy using a probabilistic model for each cluster. When labeling a frame, we will not make a *hard* decision about cluster membership, but assign a membership score proportional to its likelihood.

For each stream, a codebook of T clusters was constructed with k-means clustering:

- cluster centers: $C_1, C_2, \ldots, C_T$
- covariance matrix of the cluster: $\Sigma_1, \Sigma_2, \ldots, \Sigma_T$

With a Gaussian assumption, the likelihood of a the stream's data vector x (analysis frame) on codeword C_n is,

$$p(x; n) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_n|}} \exp\{-\frac{1}{2}(x - C_n)^T \Sigma_n^{-1} (x - C_n)\} \tag{2}$$

where D is the dimension of MFCC vectors of a stream. To retain sparsity, only the top K-ranking clusters are retained:

$$p(x; n_{x,1}), p(x; n_{x,2}), \ldots, p(x; n_{x,K}) \tag{3}$$

where $n_{x,1}, n_{x,2}, \ldots, n_{x,K}$ are the K Gaussians with the highest likelihood for the frame x. The normalized scores used for computing the co-occurrence probabilities are derived as the posterior Gaussian probabilities.

$$\hat{p}(x; n_{x,k}) = \frac{p(x; n_{x,k})}{\sum_{l=1}^{K} p(x; n_{x,l})} \tag{4}$$

In the HAC-representation, the contribution to the co-occurrence of $\{n_{x,k}, n_{y,l}\}$ (where x and y are two frames separated in the time domain by the *lag* parameter) will be

$$\hat{p}(x; n_{x,k}) \hat{p}(y; n_{y,l}) \tag{5}$$

One can see that the joint probability of x, y still sums to 1.

$$\sum_k \sum_l \hat{p}(x; n_{x,k}) \hat{p}(y; n_{y,l}) = 1 \tag{6}$$

Adaptive VQ In the adaptive VQ method, we try to select K adaptively for each frame according to its scores against all clusters. Two methods are proposed in the paper.

One is to select K according to the differences of sorted scores. For frame data x, suppose the decreasing score sequence is $p(x; n_{x,1}), p(x; n_{x,2}), \ldots, p(x; n_{x,T})$. Their differences are,

$$\delta(x; t) = p(x; n_{x,t}) - p(x; n_{x,t+1}), t = 1, 2, \ldots, T - 1 \tag{7}$$

Intuitively, for each frame, we look for the *break point* K where the sorted likelihood scores of the clusters decrease abruptly from "important" to "less important" clusters:

$$K_x = min(argmax_t \delta(x; t), 10) \tag{8}$$

Then K_x codewords are used for labeling the frame. K_x is selected adaptively for each frame. 10 is used to keep the sparsity of the coding, that is we select 10 codewords for each frame at most. After normalization, the scores can be used for a probabilistic description of the co-occurrence.
Another approach is setting a threshold,

$$\eta_x = \frac{p(x; n_{x,1})}{10} \tag{9}$$

Then K is selected by the following formula.

$$K_x = min(argmin_t p(x; n_{x,t}) > \eta_x, 5) \tag{10}$$

where 5 is used to maintain the sparsity of the coding.

2.2 HAC-based NMF Model for Vocabulary Acquisition

Above we described how to make an acoustic co-occurrence matrix (Eq.(1)) using the HAC model. In principle, the HAC word representations can be found in an unsupervised manner by NMF applied to the sparse matrix A and taking Kullback-Leibler divergence as the cost function [2]. If a grounding matrix G is used as supervision, the NMF model will find the vocabulary much more accurately as explained in the next two subsections.

Training Model The grounding matrix G was used as supervision to associate speech features and patterns with speech events and evidences. For the training set, if the n-th utterance is known to contain L key words from a set of M with indices $m_1, m_2, \ldots, m_L$ $(1 <= m_i <= M)$, we can construct the $M \times N$ grounding matrix G with accumulated ones in its m_l-th row and n-th column and zero elsewhere, where $l = 1, 2, \ldots, L$ and N is the number of utterances in the training set. For details about the NMF model and the factorization algorithms, one can refer to [2] and [3]. The basic formula is as follows.

$$V = \begin{bmatrix} G \\ A \end{bmatrix} \approx \begin{bmatrix} W_g \\ W_a \end{bmatrix} H \qquad (11)$$

The learned HAC representations of the parts that all utterances are composed of are contained in W_a. The matrix W_g links the HAC representations to the word tags. The matrix H contains the word activations on the training set.

Recognition In the stage of recognition, we first compute the activation probability matrix H' of the learned parts,

$$A' = W_a H' \qquad (12)$$

where A' is the acoustic feature matrix of the utterances in the test set. Only H' needs to be estimated while W_a is the trained model. The activation matrix B of the key words is subsequently computed for the testing utterances:

$$B = W_g H' \qquad (13)$$

By thresholding these keyword activations, we can detect words in the utterances. The threshold value will trade off false alarms for missed detections. In our evaluation, we always choose the operating point where both error types have the same occurrence frequency, i.e. we report the *equal error rate* of word detection.

Table 1: *Equal Error Rates of Multi-codebooks Method*

Number of Codebooks	1	3	5	10	15
Equal Error Rates(%)	1.869	1.600	1.559	1.551	1.547
s.t.d.(%)	0.04	0.04	0.06	0.06	0.04

Table 2: *Equal Error Rates of Soft VQ with* $K = 1$

Number of Codebooks	1	2	3	5
Equal Error Rates(%)	1.646	1.501	1.372	1.393
s.t.d.(%)	0.04	0.02	0.04	0.04

Table 3: *Equal Error Rates of Soft VQ with* $K = 3$

Number of Codebooks	1	2	3	6
Equal Error Rates(%)	1.325	1.291	1.253	1.208
s.t.d.(%)	0.07	0.06	0.06	0.08

3 WORD ACQUISITION RESULTS

The experiments were made on the ACORNS-Y2-UK database [4]. It contains 50 English keywords, each occurring at least 50 times across the entire database. There are 9998 utterances in the training set and 3300 utterances in the test set, originating from 10 speakers. The aim of the ACORNS project is to learn how infants can learn words, so these words are based on the list of words infants of about 12-15 months old are reported to understand [5].

The window length for spectral analysis was 20ms and the frame shift (hopping) was 10ms. The MFCC extraction used 30 MEL-filter banks from which 12 MFCC coefficients are computed plus the frame's log-energy. The codebook sizes for streams S, V, A were 250, 250 and 100 respectively. We selected 3% of the utterances randomly to train the codebooks. The *lags* (see Section 2.1.2) were 20, 50 and 90 ms. The common factorization dimension was 75 (refer to 11), which was larger than the number of key words (50) to deal with the information of non-keywords. NMF requires an iterative algorithm which is initialized as described in [1].

To avoid the singularity of the covariance matrix of each cluster in soft VQ, principal direction bisection was used for making sure that every cluster has at least $10 \times D$ elements where $D = 13$ is the dimension of MFCC vectors of each stream.

Since the NMF algorithm is not guaranteed to find the global minimum of its cost function, we always made 5 training attempts and report the mean error rate and the standard deviation. The mean values and standard deviations are shown in Table 1 to Table 6, the error rates versus the memory required to run the NMF programs are plotted in Figure 2.

Table 4: *Equal Error Rates of Soft VQ with $K = 5$*

Number of Codebooks	1	2	3	5
Equal Error Rates(%)	1.330	1.263	1.222	1.207
s.t.d.(%)	0.08	0.07	0.05	0.03

Table 5: *Equal Error Rates of Adaptive VQ using the Differences, criterion Eq.(8)*

Number of Codebooks	1	3	5	10
Equal Error Rates(%)	1.463	1.358	1.328	1.315
s.t.d.(%)	0.06	0.05	0.04	0.04

Table 6: *Equal Error Rates of Adaptive VQ using the Threshold, criterion Eq.(10)*

Number of Codebooks	1	3	5	10
Equal Error Rates(%)	1.291	1.241	1.219	1.258
s.t.d.(%)	0.05	0.03	0.03	0.06

4 DISCUSSION

The multi-codebooks method successfully decreases the error rate with increasing number of features (Table 1). Different data was used to train the codebooks in each feature. However, the accuracy levels off around 5 to 10 codebooks. That's probably because increasing the number of codebooks suffers from poor generalization given the limited training data.

By modeling each cluster (codebook) as a full-diagonal Gaussian and making soft assignment of MFCCs on the codebooks, the performance of our model was further improved as in Table 2, Table 3 and Table 4.

Adaptive VQ can keep the good performance of soft VQ (Table 5 and Table 6) while using a lower memory by pruning the labels with small scores for each frame.

Figure 2 shows the change of error rates with respect to the required memory. Here, more features means larger A matrix and hence more memory. Compared to the baseline in Figure 2, where we merely increase the codebook size, we do succeed in decreasing the error rates significantly with the proposed methods. In Figure 2, we can also find that applying soft VQ and adaptive VQ is a better compromise of required memory versus accuracy. $K = 3$ is a good choice for the number of codewords to be retained for each frame. We also checked the average number of labels applied for each frame in the two adaptive VQ techniques. For the first one, the average number is 1.2, for the second one, it is 1.8, which are in line with the memory plot in Figure 2.

Notice that the criteria of labeling frames is different between hard VQ (Table 1) and soft VQ with $K = 1$ (Table 2). The first one uses Euclidean distances, while the second one is tantamount to Mahanalobis distance.

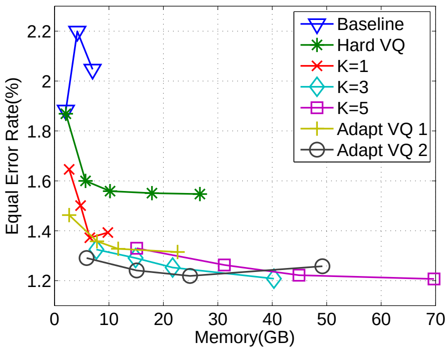

Figure 2: *Comparison of the Coding methods. The baseline is the results with increasing codebook size (S250,V250,A100; S350,V350,A300; S500,V500,A400). K is the number of selected labels for each frame in Soft VQ. Adapt VQ 1 selects labels with the differences of the sorted scores for each frame. Adapt VQ 2 takes the 1/10 of the largest score of the frame as the threshold. The number of Codebooks increases from left to right of the figure.*

5 CONCLUSION

Recurring acoustic co-occurrence relationships can be *learned* by non-negative matrix factorization (NMF). But when applying the method directly to traditional MFCC speech representations, the procedure involves vector quantization, which leads to accuracy loss. The proposed methods improved the accuracy of the NMF-approach to word learning and recognition of spoken utterances. The coding methods can overcome the disadvantages to some extent using multi-codebooks (multi-stream data fusion) or soft VQ and adaptive soft VQ. According to the evaluation on the task of spotting a vocabulary of 50 keywords in continuous speech, the performance can be improved by 35% (relative) with respect to the original coding using one codebook and hard VQ. So with the newly introduced methods, the information loss due to hard VQ can be alleviated. The ratio between the "gain" (the error rates) and the "pain" (the required memory) was improved as well.

However, the performance levels off as we scale up the present approaches to more complex models. We can see that the error rates will not decrease so much when using more than 5 codebooks (Figure 2). In future research, we will combine these methods with feature selection to address the scaling problem. NMF further has the ability to cope with the high dimensional representations that arise naturally when considering long-span time-frequency features.

The coding methods and the model framework may also prove useful for applications such as topic discovery and detection in large speech database. The modeling of co-occurrences can be used to mining of sequential patterns when integrated with NMF and Kullback-Leibler divergence.

6 ACKNOWLEDGEMENT

This research was funded by the European Commission under contract FP6-034362 (ACORNS) and K.U.Leuven research grant OT/09/028 (VASI).

7 REFERENCES

[1] H.,Van hamme "HAC-models: A Novel Approach to Continuous Speech Recognition", **Proceedings of the Interspeech 2008**, Brisbane, Australia, pp.2554-2557, 2008.

[2] H.Van hamme, "Integration of Asynchronous Knowledge Sources in A Novel Speech Recognition Framework", **Proceedings of the workshop on Speech Analysis and Processing for Knowledge Discovery**, Aalborg, Denmark, paper 038, 2008.

[3] D.Lee,H.Seung, "Learning the Parts of Objects with Nonnegative Matrix Factorization", **Nature**, Vol.401, 1999.

[4] T.Altosaar, L.ten Bosch, G.Aimetti, C.Koniaris, K.Demuynck, H.van den Heuvel,"A Speech Corpus for Modeling Language Acquisition: CAREGIVER", **Proceedings of the International Conference on Language Resources and Evaluation**, Malta, 2010.

[5] L.Boves,L.ten Bosch,R.Moore, "ACORNS–Towards Computational Modeling of Communication and Recognition Skills", **Proceedings of the 6th IEEE International Conference on Cognitive Informatics**, Vol.00, pp.349-356, 2007.

[6] M.Van Segbroeck, H.Van hamme, "Unsupervised Learning of Time-Frequency Patches as A Noise-Robust Representation of Speech", **Speech Communication**, Vol.51, No.11, pp.1124-1138, November 2009.

[7] K.F.Lee, H.W.Hon, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition Using HMM", **International Conference on Acoustics, Speech, and Signal Processing**, Vol.1, pp.123-126, April, 1988.

[8] X.D.Huang, W.H.Hon, K.F.Lee, "Large Vocabulary Speaker Independent Continuous Speech Recognition With Semi-Continuous Hidden Markov Models", **Proceedings of the workshop on Speech and Natural Language**, Cape Cod, Massachusetts, pp.15-18, 1989.