Word Discovery with Beta Process Factor Analysis

Niklas Vanhainen and Giampiero Salvi

KTH, School of Computer Science and Communication,
Department for Speech, Music and Hearing, Stockholm, Sweden

niklasva@kth.se, giampi@kth.se

Abstract

We propose the application of a recently developed non-parametric Bayesian method for factor analysis to the problem of word discovery from continuous speech. The method, based on Beta Process priors, has a number of advantages compared to previously proposed methods, such as Non-negative Matrix Factorisation (NMF). Beta Process Factor Analysis (BPFA) is able to estimate the size of the basis, and therefore the number of recurring patterns, or word candidates, found in the data. We compare the results obtained with BPFA and NMF on the TIDigits database, showing that our method is capable of not only finding the correct words, but also the correct number of words. We also show that the method can infer the approximate number of words for different vocabulary sizes by testing on randomly generated sequences of words.

Index Terms: word discovery, beta process factor analysis, Bayesian nonparametric method, non-negative matrix factorisation

1. Introduction

Applications in speech technology often rely on the assumption that the basic units of speech and how these units are combined into words are known in advance. At the same time, we know that children learning to speak acquire the linguistic units in an autonomous way from exposure to their parents' spoken language. The fact that artificial systems based on speech still lack human performance suggests that it may be beneficial to investigate learning methods that are inspired by human learning.

Many attempts have been made in the past to simulate the process of learning linguistic units from speech. These methods focus on two different, but related problems: finding phoneme-like acoustic units in an unsupervised way, e.g., [1, 2, 3], and finding recurrent sequences of acoustic units to form word candidates, e.g., [4, 5, 6, 7]. Although many studies have considered learning in a multimodal context, e.g., [8, 9], we focus here exclusively on methods that attempt to discover words from continuous acoustic speech.

In [4] a computational model for word segmentation was proposed that uses transitional probabilities between atomic acoustic events in order to detect recurring patterns in speech. The method in [5] is based on a dynamic time-warping (DTW) algorithm that compares segments of speech from an MIT lecture corpus. In [1], word discovery is performed with a method called DPn-gram based on Dynamic Programming. Finally in [6] and [7], the problem is addressed with a specific solution to factor analysis called Non-negative Matrix Factorisation (NMF).

We propose an alternative solution to the method proposed in [6, 7] by implementing the factor analysis with Beta Process Factor Analysis (BPFA) [10]. BPFA has the advantage of allowing the estimation of the size of the basis of word candidates

as well as their values. We test both these properties with real speech data from the TIDigits continuous digit database [11] and with artificial data where the vocabulary size is varied. Results are compared with those obtained with the implementation of a sparse variant of Non-negative Matrix Factorisation (NMF) from [12], with the convolutive feature of the implementation disabled.

The remainder of the paper is organised as following: in Section 2 we describe our data representation, latent feature models and BPFA. In Section 3 we explain the experiment details and the data. Finally Section 4 and 5 discuss the results and conclude the paper.

2. Method

2.1. Data Representation

In order to use factor analysis to find recurring phone sequences, we need to describe sequences of phones with a fixed-length representation. As in [6], we do this by representing utterances as a phone transition matrix which we will call V . We use two different methods to construct this matrix.

In the first method we only use clean phonetic transcriptions of the utterances. In this case, a phone transition count $c(\phi, \psi)$ is calculated by simply counting the number of times phone ϕ was followed by phone ψ in each utterance. We can also construct the matrix of second order phone transitions using the same methodology, but counting how many times phone ϕ was followed by any phone and then by phone χ . This makes for a richer representation and, therefore, for more reliable word discovery in cases of larger vocabularies. However, it can also have a detrimental effect on finding words that are shorter than two phone transitions.

In the second method, we use phone lattices obtained with a speech recogniser on spoken utterances to generate our matrix. The method is similar to the method used above if only slightly more complex:

$$c_1(\phi, \psi) = \sum_{\alpha: h(\alpha)=\phi} \sum_{\beta: h(\beta)=\psi} p(\alpha)p(\beta)\Delta_{\alpha\beta}$$

Here, α and β are any two consecutive arcs in the lattice, $h(\alpha)$ is the phone identity of the arc α , $p(\alpha)$ is the posterior probability of α , and $\Delta_{\alpha\beta}$ is the inverse of the probability of the node which both arcs α and β have in common. We may also define a second order phone transition in a similar way in order to provide a more rich representation. This representation was, however, not used in the current study.

For each utterance, we obtain the fixed-length representation by concatenating the values of the first and optionally the second order transition count matrices into a vector. Doing this for all the N utterances in our training data, we obtain an $M \times N$

matrix V for further processing, where M is the number of possible transitions between phones.

2.2. Nonparametric factor analysis

Beta Process Factor Analysis (BPFA) was proposed in [10], based on recent developments of nonparametric latent feature models and specifically models based on an Indian Buffet Process (IBP) and beta process prior. A more detailed view of nonparametric feature models may be found in [13]. We will first spend some time explaining the concept of a nonparametric latent feature model and then we explain the modifications made to the BPFA method.

2.2.1. Nonparametric latent feature models

In order to understand the concept of a latent feature model, it is useful to first review what a latent class model is. A simple example of a latent class model which should be familiar is a Gaussian Mixture Model (GMM). In a GMM each sample is assumed to be generated from a single Gaussian distribution, that, in turns, is associated with a latent class. If we assume a specific number of classes, we may infer the probability that the sample belongs to a specific latent class using an algorithm such as Expectation Maximisation (EM). In many cases however, this model is insufficient. As an example, say we want to describe an orange. Using a latent class model, we may infer that an orange belongs to the class “fruit”, but what if we also want to describe it as round, having the colour orange, etc? Latent feature models solve this problem by describing each object as a collection of features and trying to infer the subset of features that the object is most likely to possess.

A latent class model such as the one above may be represented by a binary matrix Z of size $N \times K$, where K is the number of classes and N is the number of objects we wish to classify. In this matrix, if object n belongs to class k , then $Z(n, k) = 1$. Similarly, a latent feature model may be represented as a binary matrix of the same size where $Z(n, k) = 1$ if object n possesses feature k . Unlike the matrix for the latent class model, there may be multiple non-zero entries in each row, or in other words, each object may possess multiple features.

Ideally, if we don’t know the number of latent features that can describe a collection of objects, we would want an infinite number of potential features ($K \rightarrow \infty$). Because we cannot represent an infinite binary feature matrix, we need some way of placing more weight on the features that may actually be interesting for our purposes given our data. We need a prior on Z , $p(Z)$ which places weight on parameters actually used in our data.

2.2.2. Beta Process Factor Analysis

The goal of BPFA is to find the matrices Z , W and H that best describe V in the form:

$$V = (Z \circ W)H + \epsilon$$

In our case, as explained in Section 2.1, V is a $M \times N$ matrix of all the N utterances we want to investigate and the M phone transitions of each utterance. W is of size $M \times K$ and contains the basis vectors of word candidates in the form of phone transitions believed to constitute words. H is a $K \times N$ matrix which contains the factor loadings of each word candidate in each utterance, and ϵ is noise. The symbol $\circ$ indicates the element-wise matrix product, also called Hadamard product. Although K is theoretically infinite, a sufficiently large K

is enough to approximate this infinite matrix given a distribution on Z which places weight on parameters actually used in our data. Inferring these variables using traditional means is not feasible, therefore a Gibbs sampler is used. A detailed explanation of the method is available in [14, 10], which also details a variational inference method to replace the Gibbs sampler for faster inference.

BPFA was designed as a general factorization algorithm and therefore does not restrict the value of the matrices to non-negative values as does NMF. Because of the non-negative nature of our representation as additive transition probabilities, and that in our model, each utterance is composed of a strictly additive combination of basis factors, it is essential that we can restrict the factorization to non-negative factors.

BPFA uses Singular Value Decomposition (SVD) to obtain an initial estimate of the factors W and H . SVD however can result in matrices with negative elements. To remedy this, we replaced the SVD initialization with a non-negative variant of SVD proposed in [15], this method was developed specifically for initial estimates for NMF algorithms and the factors are thus restricted to non-negativity.

The second obstacle was that BPFA models the factors as a multi-variate Gaussian distribution with a mean of 0. To restrict the algorithm to non-negative values, we switched this for a *folded* Gaussian distribution obtained by taking the absolute value of a normally distributed variable.

These two modifications resulted in a non-negative BPFA method which works well for our purposes.

3. Experiments

We performed two different sets of experiments. In the first, similarly to [6], the task is to discover digit words in the TIDigits corpus [11]. In the second we generate random sentences from the robot grasping task described in [9]. In this case we test our method for varying vocabulary sizes.

In the remainder of this section, we give the details of the experimental settings and the evaluation procedures used.

3.1. TIDigits

For the first experiment we use the TIDigits corpus [11]. The corpus contains recording of continuously spoken digit sequences and is particularly suitable for the problem of discovering a small set of words. The data was parametrised with 10 ms spaced vectors of 39 Mel-frequency Cepstrum Coefficients (MFCC) including energy and first and second order derivatives. The coefficients were normalised with Cepstral Mean Subtraction (CMS). Phone level hidden Markov models (HMMs) were trained on the training part of the corpus using the Hidden Markov Model Toolkit (HTK) [16] with up to 16 Gaussian mixture components per state. The orthographic transcriptions and a dictionary of canonical pronunciations of each digit were used in the training phase.

The use of orthographic transcriptions during training of the phonetic recogniser is somewhat contrary to the intent of discovering words in an unsupervised fashion. The way we justify this training procedure is to assume that, by the time children start to learn words, they have already acquired a set of phonetic categories. We want, therefore, to decouple the problem of learning words from that of acquiring phonetic categories and leave the second problem for future studies. This assumption is, however, not completely satisfactory because the set of acoustic categories in a language is determined by how these categories are used in order to discriminate between words (see, e.g., [1]).

In order to overcome this limitation, we tested the method with acoustic categories obtained from the speech signal with Vector Quantisation (VQ) similar to what is described in [17]. Preliminary results not included in this paper show that it is still possible to discover word patterns with a certain accuracy.

The HMMs were then used to generate phone lattices on the test data. The recognition network in this case consisted of a free loop of phones. The utterances with a single isolated digit were removed from the test set in order not to make the word discovery task too easy. The remaining 6214 lattices were used to create transition matrices as described in Section 2.

3.2. Random utterances

This experiment is intended to show that BPFA can infer the number of factors from the data alone as opposed to NMF, where the number of basis vectors has to be specified in advance. We consider words from a corpus developed in order to describe robot grasping experiments [9]. The lexicon in this corpus contains a total number of 51 words. From this set we selected four subsets of size 20, 30, 40 and 50. For each of these vocabularies we generate 5000 random utterances, composed by random sequences of words selected from the vocabulary. Using the dictionary, these words are expanded into phonemes, and the phoneme sequences are used to calculate the matrix V as described in Section 2.

When we use the BPFA method, the number of words for each experiment is not specified in advance. As a reference, we performed the experiment on the same data with NMF as well. In this case we set the number of basis vectors we wish to find to the size of the vocabulary.

3.3. Evaluation Methods

In order to determine the quality of the words discovered by the methods, we use two different metrics. The first is the average Euclidean distance between the basis vector and the transition matrices generated from lattices of the corresponding isolated word in the TIDigit database.

The second is an unordered word accuracy measure. Similar to the method used to measure word activation in [6], we divide the utterances up into overlapping windows of 600 ms each, spaced 300 ms apart. For each of these windows, a transition matrix is generated as in section 2.1, and each one of these matrices is placed in a column of V as when generating the training matrix. Then an activation matrix H is estimated with a modified version of NMF where the matrix W is kept fixed

method	BPFA		NMF	
	distance	WAcc	distance	WAcc
one	13.2255	0.9508	15.4238	0.9385
two	13.9740	0.9331	14.8347	0.9331
three	11.6082	0.8684	12.3823	0.8684
four	14.1930	0.8978	14.9064	0.8978
five	12.6847	0.8621	13.3971	0.8621
six	11.3948	0.9640	12.2701	0.9670
seven	13.5886	0.8777	14.7492	0.8777
eight	13.8018	0.3006	14.2269	0.3230
nine	13.5763	0.8031	13.3857	0.8385
zero	19.1231	0.8415	14.9754	0.8908
oh	16.1977	-0.1294	-	-
average	13.7170	0.8299	14.0550	0.8396

Table 2: Quantitative evaluation of word discovery on the TIDigits corpus. The average is computed disregarding “oh”.

and contains the basis vectors discovered by the BPFA or NMF method. For all elements of H which have a value of more than 0.10, the corresponding word is assumed to be present in the utterance. The word accuracy is defined as $Wacc = \frac{C-I}{N}$, where C is the number of correctly detected words, I is the number of falsely detected words and N is the total number of words.

4. Results and Discussion

4.1. TIDigits

Results for the TIDigit corpus are presented in Tables 1 and 2. Table 1 lists the 12 basis vectors found by BPFA. In the table, we only display for each vector the 8 transitions with the highest weights. Weights are normalised by the norm of each basis vector. We also added for each basis vector the name of the digit that the vector seems to represent. As we can see, BPFA discovers all the digits from this data set. One problem of note is for the word “oh” that is uttered as the single diphthong /ou/ (ow/ in TIMIT phone codes). The word was correctly found, but is represented in the model by the transition from /ow/ to silence. This is a weakness of the representation which can only express transitions between sounds and not single phonemes. Also, two basis vectors were found for the word “nine”.

In the table we display in bold face those transitions that are compatible with each word. Note that the transitions sil $\rightarrow$ first-phone and last-phone $\rightarrow$ sil, are also present. These transitions originate from the cases where the digit was first or last in the sequence (there is usually no silence between the digits that are spoken continuously in the corpus).

In Table 2 we give quantitative results in terms of Eu-

one		two		three		four		five		six	
w $\rightarrow$ ah	0.850	t $\rightarrow$ uw	0.914	th $\rightarrow$ r	0.726	f $\rightarrow$ ao	0.712	f $\rightarrow$ ay	0.712	s $\rightarrow$ ih	0.623
ah $\rightarrow$ n	0.399	uw $\rightarrow$ sil	0.276	r $\rightarrow$ iy	0.592	ao $\rightarrow$ r	0.595	ay $\rightarrow$ v	0.615	ih $\rightarrow$ k	0.602
sil $\rightarrow$ w	0.200	sil $\rightarrow$ t	0.123	iy $\rightarrow$ sil	0.222	r $\rightarrow$ sil	0.257	v $\rightarrow$ sil	0.198	k $\rightarrow$ s	0.392
n $\rightarrow$ sil	0.174	uw $\rightarrow$ t	0.123	sil $\rightarrow$ th	0.176	r $\rightarrow$ f	0.144	sil $\rightarrow$ f	0.170	s $\rightarrow$ sil	0.203
ah $\rightarrow$ w	0.091	uw $\rightarrow$ f	0.108	ow $\rightarrow$ th	0.093	sil $\rightarrow$ f	0.134	v $\rightarrow$ f	0.115	sil $\rightarrow$ s	0.179
ow $\rightarrow$ w	0.060	uw $\rightarrow$ s	0.088	r $\rightarrow$ th	0.074	ow $\rightarrow$ f	0.092	ow $\rightarrow$ f	0.083	ow $\rightarrow$ s	0.064
n $\rightarrow$ f	0.056	ow $\rightarrow$ t	0.087	iy $\rightarrow$ f	0.062	r $\rightarrow$ s	0.079	n $\rightarrow$ f	0.080	n $\rightarrow$ s	0.060
ay $\rightarrow$ w	0.054	r $\rightarrow$ t	0.081	n $\rightarrow$ th	0.059	n $\rightarrow$ f	0.079	v $\rightarrow$ ow	0.056	r $\rightarrow$ s	0.051
seven		eight		nine		nine		oh		zero	
eh $\rightarrow$ v	0.566	ey $\rightarrow$ t	0.831	n $\rightarrow$ ay	0.714	ay $\rightarrow$ n	0.802	ow $\rightarrow$ sil	0.972	z $\rightarrow$ iy	0.103
s $\rightarrow$ eh	0.539	t $\rightarrow$ sil	0.465	sil $\rightarrow$ n	0.648	n $\rightarrow$ ay	0.457	z $\rightarrow$ iy	0.163	iy $\rightarrow$ r	0.097
v $\rightarrow$ ax	0.531	sil $\rightarrow$ ey	0.183	ay $\rightarrow$ n	0.224	n $\rightarrow$ sil	0.356	iy $\rightarrow$ ow	0.147	sil $\rightarrow$ z	0.083
ax $\rightarrow$ n	0.198	ow $\rightarrow$ ey	0.128	ay $\rightarrow$ ey	0.076	n $\rightarrow$ f	0.065	r $\rightarrow$ ow	0.046	ay $\rightarrow$ v	0.024
sil $\rightarrow$ s	0.173	ey $\rightarrow$ ey	0.078	ay $\rightarrow$ w	0.053	n $\rightarrow$ s	0.048	r $\rightarrow$ z	0.032	ow $\rightarrow$ f	0.019
n $\rightarrow$ sil	0.092	ah $\rightarrow$ ey	0.072	ay $\rightarrow$ ow	0.051	ah $\rightarrow$ ay	0.047	ow $\rightarrow$ z	0.028	ow $\rightarrow$ s	0.014
ow $\rightarrow$ s	0.063	r $\rightarrow$ ey	0.072	n $\rightarrow$ ow	0.050	ow $\rightarrow$ n	0.046	ay $\rightarrow$ z	0.026	r $\rightarrow$ w	0.009
n $\rightarrow$ s	0.058	ay $\rightarrow$ ey	0.069	ay $\rightarrow$ z	0.048	ey $\rightarrow$ n	0.038	iy $\rightarrow$ r	0.016	ow $\rightarrow$ th	0.007

Table 1: Truncated basis vectors from BPFA with $K = 300$ on TIDigit lattices with first order transitions (TIMIT phonetic codes)

Figure 1: Number of factors and unique words found on random sequences of words with different vocabulary sizes.

clidean distance and word accuracy, as explained in Section 3.3. The word representations obtained with BPFA are, in general, slightly closer to the reference than those obtained with NMF. There are, however not big differences in terms of word accuracy between the two methods.

4.2. Random Utterances

Results for the random utterance experiments are displayed in Figure 1. BPFA correctly infers that the number of factors (basis vectors) grows with the vocabulary size, although it underestimates the number of factors for the vocabulary of size 50. The resulting basis vectors were manually inspected in order to determine whether they represent actual words contained in the vocabulary. As a result, the number of unique words is also displayed in Figure 1, both for BPFA and NMF. Both methods tend to find a number of unique words that is lower than the number of actual words in the vocabulary. This may be due to the fact that the vocabulary in this corpus contains a number of morphological variations of the same words, e.g. “graps”, “grasping”, “grasped”. These variants are considered as different words in counting the vocabulary size, but are often modelled by the same basis vector, especially at larger vocabulary sizes. The methods also allocate some basis vectors for commonly repeating phone sequences that are not words. An example of this is the fact that many words end in /ɪ/, e.g., “pushing”, “rolling” and “sliding”, and many words start with /p/, e.g., “poked”, “poking” and “picked”. The cross word phone pattern /ɪ p/, although not representing any word, appears in this corpus with sufficient frequency to be assigned to a basis vector. This is not necessary a problem with the method, because larger and more diverse sets of data will cause these spurious sequences to cancel out.

5. Conclusions

In this paper we presented a new solution to the problem of discovering word-like patterns in continuous speech. We start from a representation of each utterance in terms of phone transition counts computed from phone lattices generated by a phoneme recogniser. We solve the problem of word discovery with a recently proposed Bayesian non-parametric method for factor analysis. The advantage of this method is that it can estimate the number of words from the data at the same time as it estimates the representation for each word. The results show that the method can discover all the words in the small vocabulary case, both in a corpus of spoken digits and in data generated from a robot grasping scenario. When the vocabulary size is increased, morphological variations of words with common roots are merged into the same representation, and commonly re-

curring cross-word phonetic transitions are modelled as words. This results in a slightly underestimated number of words, but, in our opinion, this is mainly due to the limited variability contained in the specific data set.

Future work will include: substituting the phoneme recogniser with acoustic categories discovered from the data, testing with richer representations (higher order transitions), and investigating if larger and more rich data sets can resolve the problem of underestimating the number of words.

6. References

- [1] G. Aimetti, R. K. Moore, and L. ten Bosch, “Discovering an optimal set of minimally contrasting acoustic speech units: A point of focus for whole-word pattern matching,” in *Proc. Interspeech*, 2010, pp. 310 – 313.
- [2] M. Huijbregts, M. McLaren, and D. van Leeuwen, “Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection,” in *Proc. Interspeech*, 2011.
- [3] P. O’Grady, “Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint,” *Neurocomputing*, vol. 72, no. 1-3, pp. 88–101, 2008.
- [4] O. Räsänen, “A computational model of word segmentation from continuous speech using transitional probabilities of atomic acoustic events,” *Cognition*, vol. 120, no. 2, pp. 149 – 176, 2011.
- [5] A. S. Park and J. R. Glass, “Unsupervised pattern discovery in speech,” *IEEE Trans. Audio, Speech and Lang. Proc.*, vol. 16, no. 1, 2008.
- [6] V. Stouten, K. Demuyne, and H. van Hamme, “Discovering phone patterns in spoken utterances by non-negative matrix factorization,” *IEEE Signal Processing Lett.*, vol. 15, pp. 131–134, 2008.
- [7] J. Driesen, L. ten Bosch, and H. van Hamme, “Adaptive non-negative matrix factorization in a computational model of language acquisition,” in *Proc. Interspeech*, 2009.
- [8] C. Yu, L. B. Smith, and A. F. Pereira, “Grounding word learning in multimodal sensorimotor interaction,” in *Proc. Annual Conf. Cog. Science Soc.*, 2008, pp. 1017–1022.
- [9] G. Salvi, L. Montesano, A. Bernardino, and J. Santos-Victor, “Language bootstrapping: Learning word meanings from perception-action association,” *IEEE Trans. Syst., Man, and Cybern., Part B: Cybern.*, 2011. [Online]. Available: <http://dx.doi.org/10.1109/TSMCB.2011.2172420>
- [10] J. Paisley and L. Carin, “Nonparametric factor analysis with beta process priors,” in *Proc. Annual Int. Conf. on Machine Learning*, ser. ICML ’09. New York, NY, USA: ACM, 2009, pp. 777–784.
- [11] R. Leonard, “A database for speaker-independent digit recognition,” in *Proc. IEEE ICASSP*, vol. 9, mar 1984, pp. 328 – 331.
- [12] D. Wang, R. Vipperla, and N. Evans, “Online pattern learning for non-negative convolutive sparse coding,” 2011.
- [13] K. T. Miller, “Bayesian nonparametric latent feature models,” Ph.D. dissertation, University of California, Berkely, 2011.
- [14] B. Chen, M. Chen, J. W. Paisley, A. K. Zaas, C. W. Woods, G. S. Ginsburg, A. Hero, J. Lucas, D. B. Dunson, and L. Carin, “Bayesian inference of the number of factors in gene-expression analysis: Application to human virus challenge studies,” *BMC Bioinformatics*, vol. 11, p. 552, 2010.
- [15] C. Boutsidis and E. Gallopoulos, “SVD based initialization: A head start for nonnegative matrix factorization,” *Pattern Recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.
- [16] “The hidden Markov model toolkit (HTK),” <http://htk.eng.cam.ac.uk>.
- [17] H. van Hamme, “HAC-models: A novel approach to continuous speech recognition,” in *Ninth Annual Conference of the International Speech Communication Association*, 2008.