Using Bayesian Networks to find relevant context features for HMM-based speech synthesis

Using Bayesian Networks to find relevant context features for HMM-based speech synthesis

Heng Lu, Simon King

The Centre for Speech Technology Research, The University of Edinburgh, UK

hlu2@inf.ed.ac.uk, Simon.King@ed.ac.uk

Abstract

Speech units are highly context-dependent, so taking contextual features into account is essential for speech modelling. Context is employed in HMM-based Text-to-Speech speech synthesis systems via context-dependent phone models. A very wide context is taken into account, represented by a large set of contextual factors. However, most of these factors probably have no significant influence on the speech, most of the time. To discover which combinations of features should be taken into account, decision tree-based context clustering is used. But the space of context-dependent models is vast, and the number of contexts seen in the training data is only a tiny fraction of this space, so the task of the decision tree is very hard: to generalise from observations of a tiny fraction of the space to the rest of the space, whilst ignoring uninformative or redundant context features. The structure of the context feature space has not been systematically studied for speech synthesis. In this paper we discover a dependency structure by learning a Bayesian Network over the joint distribution of the features and the speech. We demonstrate that it is possible to discard the majority of context features with minimal impact on quality, measured by a perceptual test.

Index Terms: HMM-based speech synthesis, Bayesian Networks, context information

1. Introduction

Speech synthesis using HMM-like models is a well-established field. The approach taken to modelling contextually-variant speech units is a straightforward extension of the approach used in automatic speech recognition (ASR). In ASR, the context features are most commonly just the preceding and following phonemes: the modelling unit is then called a triphone. In synthesis, it is usual to add large numbers of additional context features, both phonetic and suprasegmental. However, all of this rich context information leads to an explosion of the full-context acoustic model space. In order to avoid the consequent data sparsity problem, it is usual to cluster the full-context acoustic models into a much smaller number of groups, as is also done in ASR. Widely used HMM-based speech synthesis systems, such as HTS [1] use as many as 53 context features (c.f. just 2 in ASR). Model complexity control is achieved, as in ASR, by limiting the size of the decision tree used to cluster the models: in synthesis, the MDL (Minimum Description Length) criterion is often used. However, whilst the MDL criterion itself might be based on information-theoretic principles, the decision tree is not necessarily the most effective way to discover model clusters in this exceptionally sparse space: the number of models with training examples available is a tiny fraction of all possible models. Therefore, "what is the best criterion for clustering" and "what is the optimum number of clusters" are still open questions. Going further, it is possible that clustering models is not the most effective way to control model complexity and deal with sparsity. One reason to think this is that the model clustering decision tree is learned from a very poorly trained set of unclustered models (with about 1 training example per model). In this paper, we introduce a first step away from the current approach, which, to summarise is: create a very sparse model space, control complexity by model clustering, use a decision tree to determine that clustering, and use tree size as the controlling parameter of model complexity.

In the generative model paradigm, the core problem is to model the joint distribution of all the context features and the acoustic features. The current approach naively "multiplies out" all the context features to create a vast state space with a cardinality equal to the product of the cardinalities of all the context features and the number of states per model. For the 26 context features used in this work, that is $O(10^{27})$!

It is to be hoped (and implicitly assumed in model clustering), that there are not really 10^{27} different speech units to be modelled. We must identify a subspace in which we do need models. One way to think about that is: which combinations of context features are possible; of those, which combinations actually lead to differences in the acoustics. One way to investigate that is by learning a structured model of the joint probability distribution of context features and acoustics. Contrast this to the current approach which merely "multiplies out" all the context features, creating (even if only temporarily) an unstructured and consequently very sparse model space, most of which is not needed. Structure is then re-introduced by parameter clustering.

Bayesian Networks (BNs) offer a useful framework for learning the structure in a set of variables. Each variable is a node in the network and dependencies between variables are represented by arcs between pairs of nodes, with missing arcs indicating conditional independence. In the BNs we use here, each context feature is a discrete variable in the network. The dependency structure between context features is what we wish to discover, and this can be learned automatically from data using BN structure learning algorithms. Assuming that the structure is sparse (i.e., not a fully connected graph), the joint distribution over all features (which is too large to ever learn from data directly) is factorized into the product of a number of simpler (i.e., smaller) conditional probability distributions. We have previously applied BN structure learning successfully to the problem of predicting phone duration, which also involves a relatively large number of context variables [2]. There, we used the K2 algorithm [3] to learn the structure.

In this work, we used 26 commonly-used context features. Separate Bayesian Networks are constructed for the spectral, F0

ID	Context Information	Card.
p_1	phoneme identity before previous phoneme	50
p_2	previous phoneme identity	50
p_3	current phoneme identity	50
p_4	next phoneme identity	50
p_5	phoneme after the next phoneme identity	50
p_6	position of current phoneme in syllable (forward)	7
p_7	position of current phoneme in syllable (backward)	7
b_1	whether current syllable stressed or not	2
b_2	whether current syllable accented or not	2
b_3	the number of phonemes in current syllable	7
b_4	position of current syllable in word (forward)	4
b_5	position of current syllable in word (backward)	4
b_6	position of current syllable in phrase (forward)	16
b_7	position of current syllable in phrase (backward)	16
b_{16}	name of vowel of current syllable	21
e_1	gpos (guess part-of-speech) of current word	9
e_2	the number of syllables in current word	4
e_3	position of current word in phrase (forward)	11
e_4	position of current word in phrase (backward)	11
h_1	number of syllables in current phrase	16
h_2	number of words in current phrase	11
h_3	position of current phrase in utterance (forward)	4
h_4	position of current phrase in utterance (backward)	4
j_1	number of syllables in utterance	30
j_2	number of words in utterance	17
j_3	number of phrases in utterance	4

Table 1: Context features for speech synthesis and their cardinalities (Card.)

and duration acoustic features. Various BN structure learning algorithms are tested. The structure of the learned networks are examined and used to perform feature selection. This is a simple first step, and our experiments are intended to test whether the BN structure learning is indeed discovering which the most important context features are. We then build HTS-based speech synthesis models using only the selected features. A listening test is conducted to compare these models with conventional "full context" models, which use all 26 features.

2. Bayesian Networks

2.1. Fundamentals

Let $U = \{x_1, ..., x_n\}, n > 1$ be a set of variables. A Bayesian network B over a set of variables U is a network structure B_S, which is a directed acyclic graph (DAG) over U and a set of probability tables $B_P = \{p(u|pa(u))|u \in U\}$ where $pa(u)$ is the set of parents of u in B_S. A Bayesian network represents the factorisation of the joint probability distribution $P(U) = \prod_{u \in U} p(u|pa(u))$.

In many statistical modeling problems, we wish to perform computations with the joint probability distribution (JPD) of a number of variables, for example $P(A, B, C)$. If the cardinalities of the random variables A, B, C are M, N, K respectively, then the size of the joint probability table for $P(A, B, C)$ will be $M \times N \times K$. In many interesting real-world applications, the number of discrete variable is large, and they may also have high cardinalities: this is true for the context features in speech synthesis models – se table 1. Such a large JPD causes data sparsity: it is not possible to obtain training examples corresponding to every cell in this table. The BN solution to this is to factorise the JPD; the particular factorisation is represented by the BN topology. This can be designed by hand (e.g., using expert knowledge or intuition), or automatically learned using one of a number of established BN learning algorithms.

In the case of automatic structure learning, a Bayesian Network is constructed in two steps. First, the BN structure is learned according to some score metric. Then, with the structure fixed, the conditional probability tables are learned. In this paper, we are only interested in the structure.

2.2. Score metrics

Let the entropy metric $H(B_S, D)$ of a network structure B_S and database D be defined as

$$H(B_S, D) = -N \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N} \quad (1)$$

and the number of parameters K as

$$K = \sum_{i=1}^{n} (r_i - 1) \cdot q_i \quad (2)$$

where $r_i (1 \leq i \leq n)$ is the cardinality of x_i, $q_i = \prod_{x_j \in pa(x_i)} r_j$ is the cardinality of the parents set of x_i in B_S. And we use $N_{ij} (1 \leq i \leq n, 1 \leq j \leq q_i)$ to denote the number of records in D for which $pa(x_i)$ takes its jth value. $N_{ijk} (1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i)$ denotes the number of records in D for which $pa(x_i)$ takes its jth value and x_i takes its kth value. So $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. We use N to denote the number of records in D.

- The AIC metric $Q_{AIC}(B_S, D)$ of a Bayesian network structure B_S for a database D is

$$Q_{AIC}(B_S, D) = H(B_S, D) + K \quad (3)$$

- The Minimum Description Length metric $Q_{MDL}(B_S, D)$ of a Bayesian network structure B_S for a database D is defined as

$$Q_{MDL}(B_S, D) = H(B_S, D) + \frac{K}{2} \log N \quad (4)$$

- The Bayesian metric of a Bayesian network structure B_S for a database D is

$$Q_{Bayes}(B_S, D) = P(B_S) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \quad (5)$$

where $P(B_S)$ is a prior on the network structure and $\Gamma(\cdot)$ is the gamma-function.

2.3. Structure learning algorithms

Structure learning attempts to maximise these metrics. We tried the LAGD Hill Climbing, Tree Augmented Naive Bayes (TAN) [4][5] and K2 [3] methods. LAGD Hill Climbing performs hill climbing [6] with look ahead on a limited set of best scoring steps. In TAN, a tree is formed by calculating the maximum weight spanning tree using the Chow and Liu algorithm [7]. K2 is also a hill climbing approach, adding arcs with a fixed ordering of variables. We used the Weka [8] implementation of these algorithms.

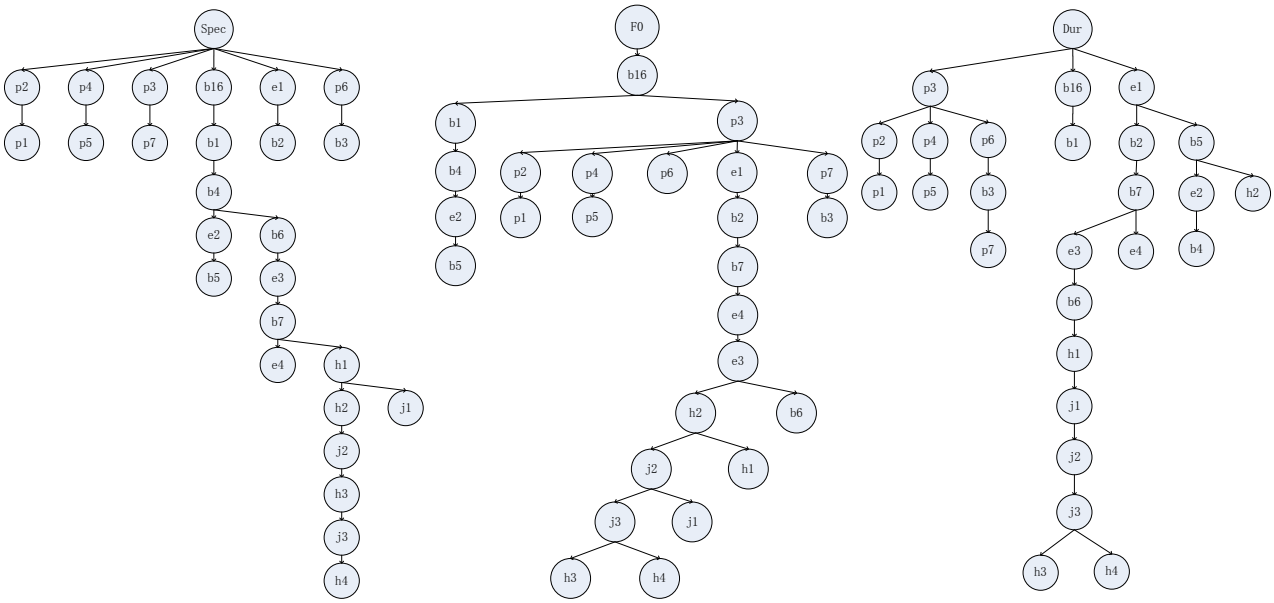

Figure 1: Bayesian Networks for (from left to right): (a) Mel-Cepstrum (LAGD), (b) F0 (TAN), (c) duration (LAGD)

3. Experiments

Since the dependency structures for spectral, F0 and duration features are likely to differ, three separate BNs were learned automatically using the structure learning algorithms introduced in Section 2.3. The three scoring metrics introduced in Section 2.2 were used to score the resulting BNs. The structure with the best score was then chosen in each case.

3.1. Database

A British English corpus with manually checked labels containing a total of 2969 utterances was used in our experiments. The speaker is male. The sampling rate is 48kHz at 16 bits and the acoustic features we used were 59th order Mel-cepstrum, log F0, and phone duration.

To learn the BN structure, we need to create a data set in which each item ('record') comprises the context features and the acoustic features. One choice that needs to be made is the temporal resolution of this dataset. We chose to use the HMM state as the basic temporal unit for BN learning in the case of spectral and F0 features: the average value of the speech features was calculated for each basic unit. For duration, we used the phone as the basic unit.

The BN structure learning algorithms we used are only available for discrete variables, so we quantised the acoustic features using LBG-based Vector Quantization (VQ) [9]. For the 59th order Mel-cepstral features, a codebook of size 512 was created. For log F0 and phone duration, codebooks with 8 and 16 classes were used. Note that the quantisation is only used for BN structure learning: the HMMs for synthesis use normal, continuously-valued features.

3.2. Context features

The 26 categories of context information most commonly used in the HTS system were used - they are listed in Table 1 along with their cardinalities. Each context feature is a discrete variable in the BN.

Learning algorithm/Scoring Metric	AIC	MDL	Bayesian
LAGD Hill Climbing	-1.149E7	-1.187E7	-1.139E7
TAN	-1.373E7	-2.127E7	-1.052E7
K2	-1.364E7	-1.419E7	-1.355E7

Table 2: Log scores for the spectrum BNs, for various structure learning algorithms

Learning algorithm/Scoring Metric	AIC	MDL	Bayesian
LAGD Hill Climbing	-1.112E7	-1.115E7	-1.111E7
TAN	-1.096E7	-1.127E7	-1.087E7

Table 3: Log score for F0 BN structure learning algorithms

Learning algorithm/Scoring Metric	AIC	MDL	Bayesian
LAGD Hill Climbing	-1.906E6	-1.924E6	-1.903E6
TAN	-2.010E6	-2.447E6	-1.815E6

Table 4: Log scores for the learned structures for modelling duration

3.3. Automatic structure learning

3.3.1. Spectrum

The best BN learned for the spectral acoustic features is shown in Fig 1(a), with the corresponding scores in Table 2. Those context features that are not connected directly to the acoustic feature are conditionally independent of it (at least, sufficiently independent to produce a good score for this structure), given the intervening context features. Another interpretation of the graph structure is that the further a context feature is from the acoustic feature, the less influence it has over the acoustics. We take this interpretation and select only those few features which are "close" to the acoustic feature. This is done by inspection of the graph structure.

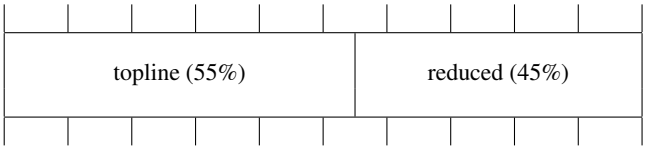

Figure 2: Results of the forced-choice subjective preference listening test.

3.3.2. F0

The best BN learned for the F0 acoustic feature is shown in Fig 1(b), with the corresponding scores in Table 3. For F0, only voiced speech was used to learn the network structure. In contrast to Fig 1(a), the F0 acoustic feature is mainly related to the vowel identity. The current phone and stress status are also closely related to F0.

3.3.3. Duration

The best BN learned for duration is shown in Fig 1(c), with the scores given in Table 4. The structure reveals that phone duration is close related to the current phoneme (p3), current vowel (b16) and part-of-speech (e1); position of current syllable in word (backward) (b5) is also closely related. [10] gives an example of within-word position and stress factor confounding. Durations of vowels turn out to be shorter in word-final syllables than in non-word-final syllables, if stressed and unstressed vowels are analysed together. But, unstressed vowels are shorter than stressed vowels and word-final syllables are five times more likely to be unstressed than stressed. So, if stressed and unstressed vowels are analysed separately, the vowel duration in final syllables (all other factors being equal) is longer than in non-final syllables, as we would expect. Therefore, one needs to take into account the vowel, stress and word-finality when predicting duration. This is consistent with the network shown in Figure 4.

3.4. Subjective listening test

A subjective preference listening test was conducted in order to test whether the structures learned do indeed predict which context features matter the most. For each acoustic feature (spectrum, F0, duration), we removed most of the context features – those least related to the acoustic feature according to the BN structure for that acoustic feature. We built two HMM synthesis systems using HTS [1]. One used all context features listed in Table 1. This is the topline. We built a second system using only those context features selected using the BNs, for spectrum (p2,p3,p4,p6,b16,e1), F0 (p2,p3,p4,p6,p7,b1,b4,b16,e1) and duration (p2,p3,p4,p6,b1,b2,b5,b16,e1). Table 5 summarises the topline and the reduced-feature systems.

25 newspaper sentences were synthesised for the listening test. 27 native speakers of English took part in a forced-choice preference test. The same sentence, synthesised by both systems, was presented as a pair, in randomised order. Sentence order was also randomised per listener. The listeners very slightly preferred the topline (all features) system at a rate of 55% to 45%. We can conclude that the quality hardly degrades, even though most context features were removed (down from 26 to just 6 or 9).

		Context features	Questions	Clusters
Mel-Cepstral	Topline	26	2211	1573
	Reduced	6	642	1470
F0	Topline	26	2211	4687
	Reduced	9	860	3251
Duration	Topline	26	2211	1013
	Reduced	9	691	890

Table 5: The number of context features used in the topline and reduced-feature systems, along with the number of question nodes in the state clustering decision tree and the number of parameter clusters.

4. Conclusions and Future work

We find that the automatic BN learning algorithms are able to discover useful structure in the data. Our first attempt to use this information is very simple: we simply perform feature selection on the large set of context features (26), reducing it to just 6 or 9 features. The next step is to make fuller use of the BN structure, and incorporate that directly into the acoustic models used for synthesis. Ultimately, one goal is to replace decision tree parameter clustering that operates in the very sparsely-populated "full-context" model space with something that avoids ever operating in this somewhat artificial model space.

5. Acknowledgements

The research leading to these results was funded from EPSRC grants EP/I031022/1 (Natural Speech Technology).

6. References

[1] "HTS," http://hts.sp.nitech.ac.jp/.

[2] O. Goubanova and S. King, "Bayesian networks for phone duration prediction," *Speech Communication*, vol. 50, pp. 301–311, 2008.

[3] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.

[4] J. Cheng and R. Greiner, "Comparing Bayesian network classifiers," *Proceedings UAI*, p. 101C107, 1999.

[5] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131C163, 1997.

[6] W.L. Buntine, "A guide to the literature on learning probabilistic networks from data," *In Proc. IEEE Trans on Knowledge and Data Engineering*, vol. 8, pp. 195C210, 1996.

[7] C.K. Chow and C.N. Liu, "Approximating discrete probability distributions with dependence trees," *In Proc. IEEE Trans. on Info. Theory*, vol. 14, pp. 426C467, 1968.

[8] "Weka," http://www.cs.waikato.ac.nz/ml/weka/.

[9] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, pp. 702–710, 1980.

[10] J.P.H. van Santen, "Assignment of segmental duration in text-to-speech synthesis," *Comput. Speech Lang*, vol. 8, pp. 95–128, 1994.