DECISION TREE USAGE FOR INCREMENTAL PARAMETRIC SPEECH SYNTHESIS

Timo Baumann

Natural Language Systems Division, Department of Informatics
Universität Hamburg, Germany
baumann@informatik.uni-hamburg.de

ABSTRACT

Human speakers plan and deliver their utterances incrementally, piece-by-piece, and it is obvious that their choice regarding phonetic details (and the details' peculiarities) is rarely determined by globally optimal solutions. In contrast, parametric speech synthesizers use a full-utterance context when optimizing vocoding parameters and when determing HMM states. Apart from being cognitively implausible, this impedes incremental use-cases, where the future context is often at least partially unavailable. This paper investigates the 'locality' of features in parametric speech synthesis voices and takes some missing steps towards better HMM state selection and prosody modelling for incremental speech synthesis.

***Index Terms*—** Speech Synthesis, Incremental Processing, HMM Synthesis, Interactivity, Spoken Dialogue Systems

1. INTRODUCTION

Most speech synthesis software is not tailored towards interactive use, but instead expects full sentences (or utterances in dialogue) to be available when processing starts. This mode of operation is becoming more and more an impediment as novel applications that require an incremental mode of operation (where utterances are constructed in a piece-meal fashion) move into the focus of attention, such as speech-to-speech translation [1], live commentary, for example in sports domains [2], or highly responsive dialogue applications [3].

In such interactive domains, the outcome of the full utterance is yet unknown when its beginning needs to be produced. In this situation, conventional systems might deal with partial input as if it were complete, which leads to sub-optimal decisions when the assumed context of the decision-making algorithm is limited by the availability of features: for example, when determining the HMM states with decision trees that rely on utterance-global features, but have to be determined based only on a prefix of the utterance, the tree will select states that would not be chosen (and considered optimal) if the remaining words of the utterance were known. As an extreme example: the last phone of a prefix will be synthesized as if no words were to follow, despite the fact that more material is known to be added before delivery actually reaches this point.

Incremental processing for speech synthesis has so far considered the problems of HMM emission probability estimation in local contexts [4, 5], and symbolic prosody generation given limited utterance contexts [6], but these works have largely ignored the mismatch between decision tree training (with full contexts) and use (within limited contexts), and only provide the general insight that incremental synthesis is helpful in advanced use-cases. This paper aims to fill the gap by dealing with the step between linguistic symbolic TTS pre-processing and the input into vocoding parameter optimization via HMMs [7], which is most often solved by decision trees.

Decision trees (such as CART [8]) have been extensively used in speech synthesis, for example for linguistic pre-processing, for prosody (duration and f_0) modelling [9], and in parametric speech synthesis for HMM state selection [10]. In this work, we analyse the decision trees used in speech synthesis as to their suitability for incremental speech synthesis, where context is limited.

Related to the present problem of determining suitable features for incremental speech synthesis, Watts et al. analysed HMM-based synthesis using limited selection criteria [11]. While their goal was to limit features by level of linguistic abstraction (and the corresponding complexity of inferring them from text), the present goal is to examine the influence of limiting decision tree features by *lookahead* into the future. Cernak et al. [12] discuss the locality of features on HMM-based synthesis and speech coding performance, and subjectively and objectively evaluate the resulting audio, however do not describe the influence of feature availability on individual HSMM feature streams.

We detail incremental speech synthesis in Section 2 before we describe the setup of our experiments in Section 3 in which we analyse the decision trees used in two speech synthesis voices. Our detailed analysis is given in Section 4 and we analyse the performance penalty from limited contexts in Section 5. Section 6 discusses the results. We draw conclusions and outline ideas for future work in Section 7.

2. INCREMENTAL SPEECH SYNTHESIS

Incremental speech synthesis is the task of starting to produce speech given only a partial utterance specification that will

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

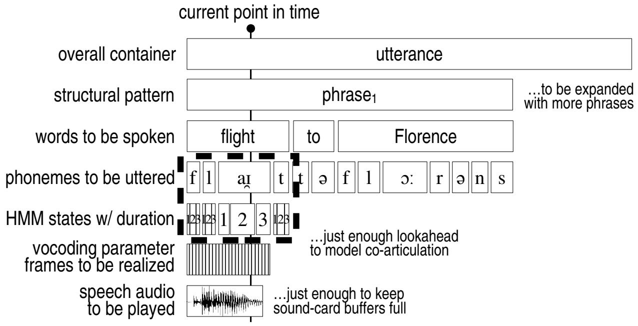
current point in time
overall container
utterance
structural pattern
phrase1
...to be expanded with more phrases
words to be spoken
flight
to
Florence
phonemes to be uttered
f l aɪ̯ t t ə f l ɔː r ə n s
HMM states w/ duration
123 123 1 2 3 123
vocoding parameter frames to be realized
...just enough lookahead to model co-articulation
speech audio to be played
...just enough to keep sound-card buffers full

Fig. 1. A processing scheme for incremental speech synthesis (adapted from [13]); the dashed rectangle marks the task at hand, generating HMM optimization input from decision trees.

later be extended [13]. Of course, incremental speech synthesis may perform at lower quality, for example because prosody contains long-range dependencies [14] which cannot fully be modelled from incomplete information. However, incremental systems should be designed in a way to take advantage of all information that is available and to degrade gracefully when confronted with missing information. This goal is in contrast to the sHTS streaming architecture for speech synthesis [5], where context is *always* limited, regardless of whether useful information is available and could be used.

Instead, we propose to implement a system in such a way that all useful information that is available is used, and substituted with plausible defaults when necessary. Furthermore, as soon as information becomes available, it should be integrated as quickly as possible. Such a process can be realized in a just-in-time incremental processing architecture such as InproTK [15]. In the architecture, levels of linguistic processing are associated to different types of incremental units [16] and future units are computed only as far as is necessary at a given moment, resulting in a processing scheme as depicted in Figure 1.

The lowest levels shown in the figure, the derivation of vocoding parameter trajectories from HMM emission optimization as well as vocoding itself have previously been shown to work incrementally (or to perform reasonably well when used incrementally) using fixed limited contexts of a few phonemes [4]. Regarding the higher levels shown in the figure, prosody processing has been shown to degrade gracefully under limited contexts [6]: the more context is available, the better the prosody generation.

The area inbetween that is tackled here, is marked by the dashed rectangle in the figure: deriving the input for HMM optimization from the symbolic levels, using decision trees.

3. EXPERIMENT SETUP

We perform our decision tree analysis using MaryTTS [17] with the BITS-1 [18] and the CMU-SLT [19] HSMM voices for German and English, respectively, as they are delivered with MaryTTS. We injected appropriate analysis code (for the analyses in Section 4) and manipulation code (for artificially limited contexts in Section 5) into the classes that perform the decision tree lookups during speech synthesis.

MaryTTS uses decision trees both for f_0 and phone duration assignments (partially based on automatically derived ToBI and stress labels), and for determining HMM state emission probabilities (mean and std dev).

HMM synthesis uses multiple, independent feature streams that are combined in the vocoding step. This means that there is more data to train HMM emission probabilities, but also that HMM states have to be selected for each feature stream. In the case of the voices investigated, there are Mel-cepstral (MCP) and aperiodicity (STRAIGHT) parameter, duration, and f_0 streams, with individual decision trees for each of the five HMM states per phoneme. For reasons of practicality, data are aggregated in the analyses below.

The analyses presented below are limited to decision trees as they are contained in these two voices of the MaryTTS distribution. The results should, however, be broadly applicable, as these reflect the data that can be observed in current day speech synthesis corpora and reflect the capabilities of state-of-the-art TTS technology.

4. NON-LOCALITY ANALYSIS

In this section we report results of our analyses of decision tree features, and their use in decision trees (statically as well as considering their use at runtime). Analyses were performed for both German and English in order to test the universality of results; where differences exist, they are pointed out explicitly.

4.1. Feature Analysis and Classification

MaryTTS uses 111 features for German, (104 for English) with the 104 features being shared among both languages, and ranging from phonetic aspects of the current phones to lexical categories and prosodic aspects of upcoming phrases.

We classified the features along two dimensions particularly relevant for incremental processing, namely:

- level of linguistic abstraction (which coincides with unit size and hence determines processing granularity), and
- temporal direction of the feature (past, current, future).

Features that combine information on two levels of abstraction are counted on the more abstract level (e. g. *number of syllables in the word* is counted as a word-level rather than a syllable-level feature). The results are presented in Table 1.

As can be seen in the table, most features relate to the phone level (and there only to the quintuple context) and fewer features relate to higher levels. As could be expected, the non-locality of features as a function of time increases with the size of linguistic units/the level of abstraction: on the phone level, the 'future' only concerns the next two phones after the current (likewise the past only concerns itself about the

Table 1. Counts of decision features, categorized along the temporal axis and by levels of linguistic abstraction (indicating granularity), for German. Feature classes are encircled.

	past	current	future
phone	20	10	19
syllable	3	8	2
word	2	7	3
phrase/accentuation	11	10	10
full sentence	—	5	—

previous two phones), which may last on the order of a few hundred milliseconds. Relevant features on the word level (e. g. *lexical category of next word*), and even more so on the phrase/accentuation level (e. g. *phrase tone of the next phrase*) may be several seconds away.

This temporal extension or *lookahead*, of course, has implications on the availability of this information in real-time synthesis. For example, the boundary tone of the current phrase may not have been decided yet (even though the phrase is already ongoing), or, the number of syllables until the next accentuated syllable will only be certain once lexicalisation (and hence the precise number of syllables) has completed.

In order to simplify the handling of the vast amounts of data (that lead to a combinatorial explosion in manual analysis), some approximation of the availability of data is helpful. For this reason, features are grouped into the following classes:

no future no lookahead into the future at all; only information in the past as well as information relating to the current phone is available; features marked as 'current' but in a higher-level representation (that typically also span the short-term future) are left out

future 1/2-phones + curr. syll one (two) phones of lookahead as well as all information on the current syllable

future syll. + curr. word information pertaining to the next syllable as well as the current word

future words + curr. phrase . . . to the next word and the current phrase

full context all context is available, as in non-incremental processing

The classes as defined above are encircled in Table 1.

In addition, more for educational purposes, we separate the 'no future' case into **past** and **curr. phone** contexts in the analyses below.

4.2. Static Analysis of Decision Trees

This subsection mainly concerns itself with the decision questions that are queried and hence analyses for the individual trees are limited.

The decision trees contain between 4 and 28 levels and contain between 132 and 1374 decision nodes, making individual analysis of all trees and decisions (and their relation to individual features) too cumbersome. Instead, summaries were generated and analysed.

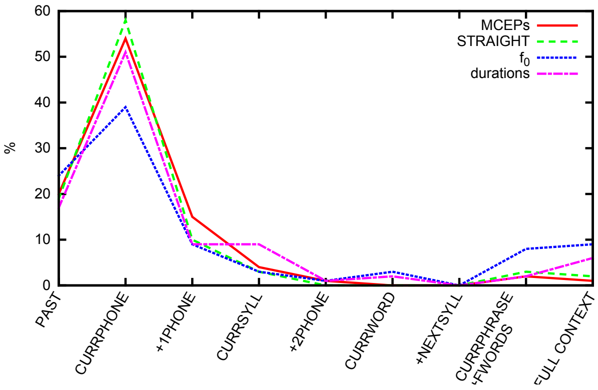

Fig. 2. Usage of per-class decision features for different types of decision trees in the German BITS-1 voice.

The vast majority of questions in the decision trees relate to the current, previous, and next phone, as well as to the 'past' category. Decision trees ask the most informing questions early on in the tree. For this reason we looked at the top 1/2/4 levels of the tree, which supports the view that the 'current phone' category contains the most informing questions. Questions relating to past, next phone, full phrase, or full sentence only become important below the first two levels of the trees.

4.3. In-vivo Analysis of Decision Tree Usage

The previous subsection's results cast a static view on the outcomes of the training process that generated the decision trees, which could potentially differ from the feature usage at runtime. Here, we profile the usage of the decision trees at runtime, by synthesizing 600 utterances from a collection of German TTS corpora [20] (for English: 300 sentences randomly sampled from the Blizzard challenge [21] test data between 2007 and 2011). We counted the individual decisions taken while traversing the decision trees, and the values of the corresponding features.

Figure 2 shows the results for the German voice, differentiated by the types of decision trees (cepstral, aperiodicity, f_0, and duration). Results for English were similar and underline the conclusion from the previous subsection, namely that the current phone, as well as the past, and the next phone are the most important feature classes.

It can also be observed that feature importance differs by decision type: cepstral and aperiodicity parameters are well described by a very limited context. In contrast, both duration and fundamental frequency (i. e. prosodic aspects) require the full-utterance context as well as features from the 'current syllable' and 'current phrase' classes. Furthermore, f_0 does not depend so much on the current phone.

5. PERFORMANCE OF INCREMENTAL DECISIONS BASED ON DEFAULT REASONING

As mentioned above, features that rely on information further out in the future utterance context will more often be unavailable in incremental speech synthesis. In this section we simulate the unavailability of features (by feature class) and measure the associated performance penalty.

Our strategy is to substitute features that are marked as missing with *default values*. Default values were gathered from the previous in-vivo analysis of decision tree querying: whenever a feature is queried, its value was recorded and the most common value (for numeric features: the arithmetic mean) is regarded as default. The chosen defaults were inspected and followed expectable and generic patterns. For this reason, we here ignored the fact that the 'training set' for finding default values is the same as the 'test set' when applying these default values to the same corpus, as reported below.

There are at least two alternatives to using default reasoning to account for missing features, which, however, do not allow to use standard 'non-incremental' voices as in our approach. Firstly, one could build custom decision trees for each of the feature classes. However, as HMM state clustering and decision tree generation are usually co-optimized, this would entail the retraining of the much more complex voice; yet such an implementation would still be limited to the envisaged feature classes and could not make use of a feature unless all features of the encompassing class are available. The second alternative is much more straight-forward and only requires a CART implementation that supports missing features. Depending on the implementation, this may also require re-training the decision trees. In addition, this mainly takes the default reasoning strategy to the corresponding node in the decision tree, and does not differ qualitatively from our approach.

As in previous work [6], we follow the maxime that (limited) incremental processing cannot systematically outperform non-incremental (full-context) processing. Hence, we evaluate the limited-context output by comparing to the full-context output. We calculate the *mean absolute error* resulting from every limited-context decision as compared to full-context decision and report z-normalized error scores (based on full-context mean and std dev). For cepstral and source parameters, which are multi-dimensional vectors, we compute the Euclidian norm from the scores z-normalized along each dimension. The results are reported in Table 2.

6. DISCUSSION

As can be seen in Table 2, the error for cepstral and source parameters decays relatively quickly, roughly halving with the addition of the 'past', 'next phone', and 'current syllable' classes. We hence infer that the lookahead required for these parameters is relatively small. As the parameters mostly relate to the immediate phonetic vicinity, this result seams reaonable.

Table 2. Z-normalized mean absolute error (MAE) of derived values for increasingly specified feature settings relative to non-incremental (full-context) processing; BITS-1 voice.

setting	f_0	dur	MCP	STR
CURRPHONE	0.77	0.70	2.72	0.89
+PAST	0.61	0.39	1.56	0.40
+1PHONE	0.53	0.27	0.64	0.20
+CURRSYLL	0.49	0.20	0.37	0.12
+2PHONE	0.48	0.17	0.25	0.10
+CURRWORD	0.45	0.13	0.21	0.10
+NEXTSYLL	0.45	0.13	0.21	0.09
+PHRASE&WORDS	0.27	0.10	0.10	0.05

In contrast, duration and especially fundamental frequency errors do not decay as quickly. Prosody has long-range dependencies which shows in our analysis. An informal listening experiment underlines that the perceptual influence of cropping features to certain feature classes has a higher influence on the perceived speech quality for aspects of prosody generation (f_0 and duration) than for cepstral and source parameters. In addition, the f_0 mean absolute error of 0.27 standard deviations in the least limited setting is still well audible, whereas the same MAE for cepstral coefficients is hardly noticeable.

We conclude that our simple default reasoning approach works fairly well for cepstral and source parameters, whereas some more elaborate method is necessary for incremental prosody modelling with small lookaheads into the future.

7. CONCLUSIONS AND FUTURE WORK

We presented the analysis of feature usage (by 'temporal distance' of feature classes) in decision trees for parametric speech synthesis, with the goal of assessing the use of decision trees in incremental speech synthesis systems. We have also shown a simple implementation based on default reasoning which shows that voice quality decisions can be taken with relatively small lookaheads, whereas prosody requires a larger lookahead or more advanced methods.

One limitation of the experiment was that feature classes were cropped away regardless of whether the feature would be available even in incremental synthesis. For example, the information that the *last* word of an utterance has been reached should result in all feature values becoming accessible. Such an implementation would of course be preferable and produce somewhat better results. We plan to fully integrate such a more advanced approach into the InproTK incremental speech synthesis component. In addition, features could also be determined based on underspecified information, e. g. based on information that is usually available in a dialogue system (like dialogue state) and we plan to elaborate this.

Acknowledgements

The author would like to thank Sven Mutzl for performing parts of the static decision tree analysis while on an internship with Prof. Kai Yu at Jiao Tong University, Shanghai, China; and Wolfgang Menzel and Arne Köhn for valuable comments. This work is supported through a grant by Daimler and Benz Foundation.

8. REFERENCES

[1] Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez, "Real-time incremental speech-to-speech translation of dialogs," in *Proceedings of NAACL-HTL 2012*, Montréal, Canada, June 2012, pp. 437–445.

[2] David L. Chen and Raymond J. Mooney, "Learning to sportscast: A test of grounded language acquisition," in *Proceedings of the 25th International Conference on Machine Learning (ICML-2008)*, Helsinki, Finland, July 2008.

[3] Timo Baumann, *Incremental Spoken Dialogue Processing: Architecture and Lower-level Components*, Ph.D. thesis, Bielefeld University, Germany, May 2013.

[4] Thierry Dutoit, Maria Astrinaki, Onur Babacan, Nicolas d'Alessandro, and Benjamin Picart, "pHTS for Max/MSP: A streaming architecture for statistical parametric speech synthesis," Tech. Rep. 1, Université de Mons, Mar. 2011.

[5] Maria Astrinaki, Nicolas d'Allessandro, Benjamin Picart, Thomas Drugman, and Thierry Dutoit, "Reactive and continuous control of HMM-based speech synthesis," in *IEEE Workshop on Spoken Language Technology (SLT 2012)*, Miami, USA, Dec. 2012.

[6] Timo Baumann and David Schlangen, "Evaluating prosodic processing for incremental speech synthesis," in *Proceedings of Interspeech*, Portland, USA, Sept. 2012, ISCA.

[7] Tomoki Toda and Keiichi Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. 90, no. 5, pp. 816–824, 2007.

[8] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, *Classification and Regression Trees*, Wadsworth, Monterey, USA, 1984.

[9] Caren Brinckmann and Jürgen Trouvain, "The role of duration models and symbolic representation for timing in synthetic speech," *International Journal of Speech Technology*, vol. 6, no. 1, pp. 21–31, 2003.

[10] Heiga Zen, Keiichi Tokuda, and Alan W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, 2009.

[11] Oliver Watts, Junichi Yamagishi, and Simon King, "The role of higher-level linguistic features in HMM-based speech synthesis," in *Proceedings of Interspeech*, Makuhari, Japan, Sept. 2010, ISCA, pp. 841–844.

[12] Milos Cernak, Petr Motlicek, and Philip N. Garner, "On the (un)importance of the contextual factors in HMM-based speech synthesis and coding," in *Proceedings of ICASSP*, 2013.

[13] Timo Baumann and David Schlangen, "INPRO_iSS: A component for just-in-time incremental speech synthesis," in *Proceedings of ACL System Demonstrations*, Jeju, Korea, July 2012.

[14] William J.M. Levelt, *Speaking: From Intention to Articulation*, MIT Press, 1989.

[15] Timo Baumann and David Schlangen, "The INPROTK 2012 release," in *Proceedings of SDCTD*, Montréal, Canada, June 2012.

[16] David Schlangen and Gabriel Skantze, "A general, abstract model of incremental dialogue processing," in *Proceedings of the EACL*, Athens, Greece, Apr. 2009, pp. 710–718.

[17] Marc Schröder and Jürgen Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 3, pp. 365–377, 2003.

[18] Tania Ellbogen, Florian Schiel, and Alexander Steffen, "The BITS speech synthesis corpus for German," in *Proceedings of LREC*, Lisbon, Portugal, May 2004.

[19] John Kominek and Alan W Black, "The CMU arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, Pittsburgh, USA, June 2004.

[20] Klaus Kohler, "Erstellung eines Textkorpus für eine phonetische Datenbank des Deutschen," in *Arbeitsberichte des Instituts für Phonetik der Universität Kiel (AIPUK)*, Klaus Kohler, Ed., vol. 26, pp. 11–39. 1992.

[21] A Black and Keiichi Tokuda, "The Blizzard Challenge 2005: Evaluating corpus-based speech synthesis on common databases," in *Proceedings of Interspeech*, Lisbon, Portugal, Sept. 2005, ISCA.