OPTIMIZING SEGMENT LABEL BOUNDARIES FOR STATISTICAL SPEECH SYNTHESIS

Alan W Black and John Kominek

Language Technologies, Institute, Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

This paper introduces a new optimization technique for moving segment labels (phone and subphonetic) to optimize statistical parametric speech synthesis models. The choice of objective measures is investigated thoroughly and listening tests show the results to significantly improve the quality of the generated speech equivalent to increasing the database size by 3 fold.

Index Terms— Speech Synthesis, Parametric Speech Synthesis, Label Boundary Optimization

1. INTRODUCTION

Building synthetic voices has become an almost automatic process with a number of toolkits available to aid the process [1] [2]. Modern techniques fall into two categories: unit selection [3] where appropriate sub-word units are selected from large databases of natural speech; and statistical parametric synthesis [4] where generative parametric models are trained from databases of natural speech. In both cases a well-labeled database is necessary to build good synthesis. Although hand labeling (or more often hand correction) can be used on a database, the human effort is expensive so fully automatic labeling techniques (previously developed) have been applied. Building on top of speech recognition techniques, HMM-based segment labeling is typically used to get initial labeling. For example [5], [6], [7] are toolkits available to automatically label the data. However hand label corrections can improve the quality of the synthesized voices. There has also been work in trying to improve these automatic labels [8]. However in most of these cases no explicit resynthesis is done to prove the label improvements help the synthesis.

In this work we present a label moving optimization technique that includes a synthesis quality measure as an inherent part of the optimization. We show the technique on a large number of databases and justify the objective measure, mean Mel-Cepstral Distortion [9], as a reasonable measure.

2. CLUSTERGEN

CLUSTERGEN [10] is a statistical parametric synthesizer that is part of the FestVox suite of voice building tools [1]. Following the work on HMM-generation synthesis in HTS [2], CLUSTERGEN offers a similar framework for development of voices. CLUSTERGEN requires a database to be labeled with HMM-state sized segments. Each phone model consists of 3 states. We use EHMM [5] in these experiments, but have also used JANUS [11] and SPHINX [6].

Once the states are labeled, feature vectors are computed for each 5ms frame. The vector includes MELCEP, delta MELCEP, F0 (interpolated through unvoiced regions), and probability of voicing.

The frames from the same segment type are clustered using CART indexed by contextual features. In addition to the standard phonetic, metrical, and phrase based features, we have found that position in phrase, phoneme, and most importantly position within the state itself has a significant improvement on quality. Thus the clusters are dependent on where in the state they are.

At synthesis time the frames are generated from the cluster tree. The frames consist of F0, MELCEP plus deltas, and voicing, and importantly variances for each of these coefficients. We then apply maximum likelihood parameter generation (MLPG) that we have adopted from HTS [2] into CLUSTERGEN. MLPG finds the optimal path through the CART-tree generated parameters, taking into account the prediction variance of each frame and the inter-frame deltas [12]. MLPG if effectively a smoothing filter, and we have found it improves quality significantly. Once the MELCEP parameters are predicted we use the MLSA [13] parameter inversion filter to generate the waveform file. Excitation of the filter is controlled by the probability-of-voicing feature that is a component of each predicted frame.

3. MOVE LABEL SEGMENT BOUNDARIES

CLUSTERGEN is a robust voice building technique that has been used on a wide range of speech databases, from standard and non-standard English in the ARCTIC databases [14], to databases collected in different languages as part of the CMU SPICE project [15]. However we are still looking for better quality synthesis, especially when dealing with small databases (less than 30 minutes of speech).

We are aware that HMM-based labeling techniques are never ideal for synthesis, and that locating better boundaries

Thanks to NSF for funding grant number 0415021, SPICE: Speech Processing Interactive Creation and Evaluation Toolkit for new Languages.

could improve models. Thus we decided to look at techniques to improve segment labeling that could improve our synthesis.

The core idea is to examine each segment boundary and consider moving it forward or back one frame if that would improve the prediction of the frame, i.e. by decreasing the distance between the predicted and original frame. The algorithm examines each label boundary in the database once before rebuilding the models and iterating again.

MLSB: Move Label Segment Boundaries
For each segment and HMM-state boundary
P_i = predicted frame coeffs before boundary
P_{i+1} = predicted frame coeffs after boundary
A_i = actual frame coeffs before boundary
A_{i+1} = actual frame coeffs after boundary

if dist(A_i,P_{i+1}) $<$ dist(A_i,P_i)
move boundary -1 frame
elif dist(A_{i+1},P_i) $<$ dist(A_{i+1},P_{i+1})
move boundary +1 frame
else
no movement
Rebuild the voice models with new labels.
Repeat until convergence criterion is reached.

This algorithm is not recommended for attempting flat-start labeling of a speech database. Rather, it is perturbation mechanism. It assumes a starting point that is reasonably good, but is not optimal.

The MLSB algorithm incorporates a shortcut to speed computation in that the distance calculation dist(A_i,P_{i+1}) compares the actual MCEP features at frame i (left of the boundary) with the predicted features at frame i+1 (right of boundary). Strictly speaking, the dist(A_i,P_i) should be computed a second time after label under examination has been (tentatively) moved. This is because moving the label boundary increases or decreases the segment length, and segment length influences the prediction of frame values. However the majority of prediction features remain unchanged. The inaccuracy of this approximation is small, and makes it possible to predict the frames of a wavefile only once per iteration.

Even this this shortcut, the algorithm is a fairly computationally expensive operation, as it requires checking each boundary in the databases. In a one-hour database there will be about 120K state boundaries. Rebuilding the models takes about 90% of the time to complete an iteration, while moveing the labels is about 10%. A 20 iteration pass of a full Arctic database (1132 utterances, which is around an hour of speech) take about 3 days on a 3GHz Xeon. A databases of only 100 uterances (5 minutes of speech) takes only 90 minutes for 10 iterations including initial labeling.

There are choices particularly regarding the distance metric, as well when to stop iterating and the maximum number of frames that a boundary may be allowed to move. We chose to optimize MCD on the grounds that this is a commonly used objective measure of voice quality. The result is a local optimization the prediction models according to measure used to evaluate the models. This is in contrast to the original embedded Baum-Welsh labeling, which optimize the likelihood of the =small HMM-based models given the data.

4. LABEL MOVEMENT USING MCD

As an initial test of the technique we used unnormalized MCD, a Euclidean distance including the C0 and C1-C24 terms, but no deltas. We applied this to the ARCTIC *rms* voice (a standard clearly articulated US English databases of about 1 hours of speech). The database was split into training and test sets, every tenth utterance was put into the test set. The MCD measure is given for the test set. The model does not include any of the test sentences, but the move labels algorithm was applied to the test sentences too. This is necessary to keep the data consistent. Excluding the test set from label movement would cause the the labeling to diverge in unknown ways.

Pass	Moves	MCD	Pass	Moves	MCD
0	0	5.253			
1	48130	5.126	6	29523	5.062
2	40896	5.088	7	28792	5.049
3	35560	5.071	8	27481	5.041
4	32816	5.064	9	26739	5.035
5	31086	5.060	10	26075	5.037

There are around 120K state boundaries per database. A boundary may only move once per iteration, but may move during more than one iteration. In the above test about 25% of the labels never move. Of those that do move 25% move only once, while only 2.5% of the labels move in all 10 iterations. Although some labels move back and forth every iteration, many move only (or almost only) in one direction.

If the process is run for many more iterations after settling down, MCD jitters about 0.02 over at least the next 40 iterations. The number of moves progressively gets smaller, implying that the process is converging on some local optimum. Yet the number of label movements per iteration remains relatively high (20%) after no significant improvement is seen in MCD. Allowing double moves at each pass does not seem to make much difference in the speed of convergence, though the MCD does drop quicker in the initial iterations.

In listening to the 9th iteration (MCD=5.035), the voice sounded better than the initial model. This encouraged us to expand listening tests to other members of our group, which are reported below in Section 7. The qualitative impression has that there are fewer inappropriately abrupt transitions. We found that the phonemes with highest MCD include the stop consonants and affricatives with a release component, such as /ch, dh, jh/. Sustainable fricatives are the easiest category to predict.

Rank	Phone	MCD Before	MCD After	Gain	Frames
1	f	4.459	4.378	0.081	1801
2	sh	4.662	4.518	0.144	742
3	s	4.668	4.563	0.105	4122
4	zh	4.762	4.825	-0.063	21
38	dh	6.053	5.805	0.248	1983
39	k	6.334	5.891	0.443	1570
40	jh	6.359	5.787	0.572	614
41	g	6.938	6.703	0.235	885

5. ALL ARCTIC DATABASES

Previously we have collected a number of single speaker well recorded phonetically balanced speech databases [14]. This set has increased to include 9 different databases. We applied our standard CLUSTERGEN build to these voices and applied MLSB to the labels. All these databases are in English though some are (fluent) non-natives. DBs *clb* and *slt* are female, all the others are male.

DB	English Dialect	Size mm:ss	MCD Base	MCD Best	Gain	Pass
ahw	German	30:50	5.254	5.019	0.235	15
awb	Scottish	55:32	4.677	4.532	0.145	10
bdl	American	50:53	5.671	5.454	0.217	10
clb	American	64:08	4.838	4.667	0.171	11
jmk	Canadian	53:47	5.400	5.208	0.192	17
ksp	Indian	59:20	5.285	5.111	0.174	17
rms	American	66:10	5.253	5.012	0.241	17
rxr	Israeli	43:38	5.298	5.120	0.178	15
slt	American	56:35	5.170	4.964	0.206	20

The average improvement is 0.195. If we use the results from [16] where a doubling of data improved MCD by 0.12, MLSB effectively improves a voice as much as increasing the data by 325%.

One database *awb* did not perform as expected. It had an increase in MCD after the first pass. Closer examination suggested that the relatively large amount of initial and final silences was the problem. We trimmed the silences on the utterances and re-ran the MLSB process, the results of which are contained in the above table. After proper trimming the abnormal behavior went away.

6. VARYING MCD

When using unnormalized 25-D mel cepstral feature vectors, the magnitude of variation is largest in the C0 term, and decreases to the C24 term, where the variation is smallest. Because C0 corresponds to log power, while C1-24 corresponds to the spectral components, it is a reasonable choice to separate out this term. We show this for the first six iterations.

Pass	Optimize C0-24 C0-24	C1-24	C0	Optimize C1-24 C0-24	C1-24	C0
1	5.162	4.472	2.125	5.260	4.496	2.332
2	5.121	4.454	2.073	5.262	4.477	2.365
3	5.112	4.462	2.041	5.280	4.469	2.405
4	5.080	4.450	1.997	5.278	4.453	2.417
5	5.078	4.451	1.993	5.286	4.453	2.432
6	5.070	4.459	1.962	5.278	4.444	2.427

In the left half of this table we see that most of improvement in C0-24 may be attributed to a decease in C0. One may conclude that MLSB primarily improves power modeling, with some improvement in spectral modeling. In the right side of the table MLSB has been altered to optimize C1-24. This gain in C1-24 is 4x better (0.52 vs. 0.13). The price paid is that C0-24 and C0 go up, and as the next section reveals, user preference goes down.

7. LISTENING TESTS

Our objective in conducting listening tests was to assess the contribution of MLSB, determine what MCD variant is best, and to place these results in context. Two useful frames of reference are a) the benefit of MLPG over a baseline system without it, and b) the addition of natural durations to improve prosodic quality. This gives six voice configurations. To get a relative ranking we conducted AB listening tests with five users. The users were presented with a randomized selection of the 113 test utterances, distributed among the 15 possible pairwise comparisons. As described in [17] solving a Bradley-Terry statistical model produces a rating for each. The table below includes 90% confidence bounds on the ratings, and the likelihood of superiority (LOS, a percentage) that the system with MLSB is superior to the others. These numbers are based on 356 AB comparisons made using identical playback equipment.

System Configuration	Rating	+/-	LOS
with natural durations	72	45	10
with MLSB	30	40	–
with MLPG	15	38	69
baseline voice (no MLx)	0	39	83
with MLSB, optimize C0	-13	58	85
with MLSB, optimize C1-24	-32	51	95

The rating separation between MLSB and MLPG is 15 points, which is exactly the advantage MLPG holds over the baseline. While more user tests are required to tighten the bounds, these results say there is a 69% likelihood that MLSB is a verifiable improvement. Combining these two techniques, a rating difference of 30 points approaches the 42 point advantage observed by the system with natural prosody (i.e. of copy synthesis). Also, we can definitively state that C0-24 is the form of MCD that should be optimized.

8. DISCUSSION AND CONCLUSIONS

One interesting question about our technique is what actually happens to the labels. We have not yet reached a definitive answer, but note that the labels do drift away from where an expert human labeler would place them. We hypothesize that in order to better model power, the segments are moving from being HMM-states to something more like di-states (from middle of one state to middle of another – cf. diphones). This does seem to happen with phrase final states, and nasals, though it is not universal.

We have noticed that some segments become very short, and are in fact errors in the labeling. This is particualrly true of silences automatically inserted during initial labeling. These progressively get shorter with MLSB. Segment deletion is currently prohibited, but should be performed under the right conditions.

Our technique for adjusting labels bears resemblance to context-dependent labeling. Yet our clusters care about not just standard tri-phone contexts, but includes stress and higher level features, plus sub-state positional features. We plan to try seeding the system with context-dependent labels. Our expectation is that the solution will converge to a similar point.

Similarities exist to discriminative training, as we are optimizing our models based on the predictive error. But there are substantial differences between ASR discriminative training and what we are doing, so it would be confusing to refer to it by that name. MLSB is also different from the discrimitive training technique of minimum generation error MGE [18].

With MLSB we are optimizing the label boundaries with respect to a speech synthesis sensitive measure, and hence see improvement on that measure. Our label-refinement technique is *metric-aware*. We believe other parametric synthesis techniques such as HTS can benefit from this approach. Unit selection synthesis may potentially benefit as well.

Our listening results help establish that MCD is a reasonable subsitute for subjective tests. An open line of research is to hunt for an even better objective measure. Even if this is not immediately forthcoming, one can identify other aspects of training that can benefit by being directly optimized against the evaluation metric.

9. REFERENCES

[1] A. Black and K. Lenzo, "FestVox: Building voices in the Festival Speech Synthesis System," `http://festvox.org/bsv/`, 2000.

[2] K. Tokuda, H. Zen, J. Yamagishi, T. Masuko, S. Sako, T. Nose, and K. Oura, "Hmm-based speech synthesis system (hts)," `http://hts.sp.nitech.ac.jp`.

[3] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP-96*, Atlanta, Georgia, 1996.

[4] A. Black, H. Zen, and K. Tokuda, "Statistical parametric synthesis," in *ICASSP2007*, Honolulu, Hawaii, 2007.

[5] K. Prahallad, A. Black, and R. Mosur, "Sub-phonetic modeling for capturing pronunciation variation in conversational speech synthesis," in *ICASSP2006*, Toulouse, France, 2006.

[6] X. Huang, F. Alleva, H.-W. Hon, K.-F. Hwang, M.-Y. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7(2), pp. 137–148, 1992.

[7] P. Woodland and S. Young, "Htk: Speech recognition toolkit," `http://htk.eng.cam.ac.uk/`, 2008.

[8] L. Wang, Y. Zhao, M. Chu, J Zhou, and Z. Cao, "Refining segmental boundaries for tts database using fine contextual-dependent boundary models," in *ICASSP 2004*, Montreal, Canada, 2004.

[9] M. Mashimo, T. Toda, K. Shikano, and N. Campbell, "Evaluation of cross-language voice conversion based on gmm and straight," in *Eurospeech 2001*, Aalborg, Denmark, 2001.

[10] A. Black, "CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling," in *Interspeech 2006*, Pittsburgh, PA., 2006.

[11] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one pass-decoder based on polymorphic linguistic content assignment," in *ASRU 2001*, Trento, Italy, 2001.

[12] K. Tokuda, T. Yoshimura, T. Masuko, and T. Kobayashi, T. nae Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis.," in *ICASSP 2000*, Istanbul, Turkey, 2000.

[13] S. Imai, "Cepstral analysis/synthesis on the Mel frquency scale," in *ICASSP-83*, Boston, MA, 1983.

[14] J. Kominek and A. Black, "The CMU ARCTIC speech databases for speech synthesis research," Tech. Rep. CMU-LTI-03-177 `festvox.org/cmu_arctic/`, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, 2003.

[15] T. Schultz, A. Black, S. Badaskar, M. Hornyak, and J. Kominek, "Spice: Web-based tools for rapid language adaptation in speech processing systems," in *Interspeech 2007*, Antwerp, Belgium, 2007.

[16] J. Kominek, T. Schultz, and A. Black, "Synthesizer voice quality on new languages calibrated with mel-cepstral distorion," in *SLTU 2008*, Hanoi, Viet Nam, 2008.

[17] A. Black, C. Bennett, J. Kominek, B. Langner, K. Prahallad, and A. Toth, "Cmu blizzard 2008: Optimally using a large database for unit selection synthesis.," in *Blizzard Challenge 2008*, Brisbane, Australia, 2008.

[18] L. Qin, Y. Wu, Z. Ling, R. Wang, and L. Dai, "Minimum generation error linear regression based model adaptation for hmm-based speech synthesis," in *ICASSP 2008*, Las Vegas, NV, 2008.