A classifier-based target cost for unit selection speech synthesis trained on perceptual data

Volker Strom, Simon King

Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK
vstrom@inf.ed.ac.uk

Abstract

Our goal is to automatically learn a *perceptually*-optimal target cost function for a unit selection speech synthesiser. The approach we take here is to train a classifier on human perceptual judgements of synthetic speech. The output of the classifier is used to make a simple three-way distinction rather than to estimate a continuously-valued cost.

In order to collect the necessary perceptual data, we synthesised 145,137 short sentences with the usual target cost switched off, so that the search was driven by the join cost only. We then selected the 7200 sentences with the best joins and asked 60 listeners to judge them, providing their ratings for each syllable. From this, we derived a rating for each demiphone. Using as input the same context features employed in our conventional target cost function, we trained a classifier on these human perceptual ratings.

We synthesised two sets of test sentences with both our standard target cost and the new target cost based on the classifier. A/B preference tests showed that the classifier-based target cost, which was learned completely automatically from modest amounts of perceptual data, is almost as good as our carefully- and expertly-tuned standard target cost.

Index Terms: speech synthesis, unit selection, target cost

1. Introduction

In previous work [1], we investigated the target cost used in the Festival unit selection speech synthesis system, as a precursor to the current work in which we have developed a technique for automatically learning a target cost function from perceptual data.

There have been a number of attempts to learn the target cost for unit selection. In early work by Hunt and Black [2], the weights in a conventional weighted-sum-of-factors target cost function are learnt by linear regression, using the linguistic context features as input variables and the acoustic distance between natural and synthesised speech as the output variable.

Another approach to learning a target cost is to build a tree which clusters the units in the database into acoustically similar sets, by querying their linguistic properties [3, 4, 5]. This tree thus predicts acoustic qualities from only linguistic features and, by controlling the tree depth, some generalisation is achieved (i.e. not all linguistic features are queried along every possible branch of the tree). The "cl-units" method of Black and Taylor [5] does this, and uses the distance from the cluster centroid as the target cost (i.e. the cluster centroid is considered to be the ideal unit for the current linguistic context).

However, the use of natural target utterances and acoustic distance measures fails to acknowledge that there is always more than one acceptable way to render a given utterance. It is also overly restrictive: there will always be missing units in terms of context feature combinations, but a missing unit is not a problem if a perceptually equivalent unit exists elsewhere in the database, and if we know how to select it based on its linguistic features. The key challenge is to establish the relationship between the linguistic context features and *perceptual acceptability* rather than acoustic similarity to somewhat arbitrary natural target speech.

With a conventional weighted-sum-of-factors target cost function using around 10 weights, a vast amount of perceptual data would be required to learn the weights automatically [6]. Furthermore, these conventional cost functions are generally *linear* combinations of factors, which is unlikely to be appropriate: the effects of the previous segment and syllable stress (for example) probably do not combine in a linear way to contribute to a unit's acceptability in a given context.

But is it really necessary to place a fine-grained continuously-valued cost on each candidate unit? It is widely accepted that a single large error in the output speech will lead to a very poor user rating, regardless of the quality of the remainder of the utterance. Therefore, the primary goal during synthesis should be (in combination with achieving imperceptible concatenation points) to have no unacceptable units in the output, with a secondary goal of maximising the number of very good units. This goal can best be achieved by a classifier rather than a continuous function operating in a very sparsely populated high-dimensional input space.

Section 2 describes the collection of perceptual data required to train the classifiers. Section 3 describes experiments with various classifier types and parameter settings. Section 4.1 describes how four SVNs constitute the new target cost function. Section 4 reports the results of two perceptual evaluations of the new target cost function.

2. Collection of Perceptual Data

2.1. Pilot experiment

In a pilot experiment, we trained a target cost classifier for just one phone in one particular context. Our perceptual data consisted of a carrier sentence ("My name is Roger") which was synthesised using all 147,820 possible combinations of the diphones /n_ei/ and /ei_m/ found in our voice database. We selected the 92 versions having nearly perfect joins (found by shortlisting using the join cost, followed by final selection by the authors) and had them rated by listeners. The classifiers trained with these ratings worked very well for this particular phone-in-context, but (as we would expect) did not generalise well to other contexts, let alone to other phones.

2.2. Creation of Stimuli

In the main experiment, we required stimuli covering a wide variety of phones in a rich set of contexts. We decided to use short sentences so that we could ensure "nearly perfect" joins throughout – i.e., with a join cost of nearly zero (we discuss this further in section 5) – but a wide variety of context feature mismatches. We paid listeners to mark any "bad" syllable – i.e., any syllable within which a problem occurred.

We synthesised 145,137 short sentences with the target cost switched off, i.e. the search was driven by the join cost only.The sentences were chosen from the British National Corpus, were between two and four words long and did not contain numbers, abbreviations or words not covered by our lexicon.

We knew from previous experiments that a mismatch in the context features "boundary tone" or "emphasis" is always unacceptable to listeners, therefore we excluded these cases from the stimuli candidates. We selected the 7,200 sentences with the smoothest joins (33,546 syllables; 99,131 phones).

2.3. Perceptual Rating

The stimuli were presented on a web page, showing the syllabified orthography with a check box under each syllable. The listeners were able to play each stimuli as often as they wished (although they were mostly played only once) and their instructions were to "mark every syllable which you think sounds wrong" and if there was a problem at a syllable boundary, to mark the syllable thereafter, or, if in doubt, both. Some words were syllabified in a non-standard way, e.g. "clu -tter" instead of "clut-ter", so that if – in this example – something is wrong with the "t", there was no need to mark both syllables. Eight pre-marked examples were given, along with an explanation for each mark.

The advantage of a binary rating of syllables, as opposed to MOS rating of phones, is that it is much quicker and can be done by non-experts, as is does not require a phonetic transcription. The disadvantage is of course that there is not a one-to-one mapping between the perceptual acceptability of syllables and the perceptual acceptability of their constituent demiphones.

We recruited 60 undergraduate students, very few of whom were familiar with phonetics or speech synthesis. All used headphones and were native English speakers. The 7,200 stimuli were split into 12 subsets of 600. Five listeners were allocated to each subset, and each listener heard each of their 600 stimuli twice, in a random order. The rating session took each listener on average 72 minutes.

Since there were 5 listeners per stimuli and two repetitions per listener, each syllable received between 0 and 10 "bad" marks. Rather than use this directly, we converted it to a binary distinction ("good" or "bad"); for most experiments, the category boundary (threshold) was set to 5.5.

The repeated stimuli allowed us to measure the consistency of each listener: with x being the number syllables marked in only one of the duplicates and y the number syllables marked in both, the ratio x/y can be used to judge the inconsistency of the listener.

The percentage of syllables marked as bad by each individual listener is an indicator for that listener's generosity or tolerance for imperfect synthesis. Figure 1 shows the listeners' inconsistency and intolerance.

The mean$\pm$stddev of the ratio x/y is 1.4 ± 0.8; which corresponds to an F-score of 0.72 ± 0.06, assuming that for each speaker all y were true positive identifications of bad syllables, half of the x are false positives and there were an equally large number of false negatives.

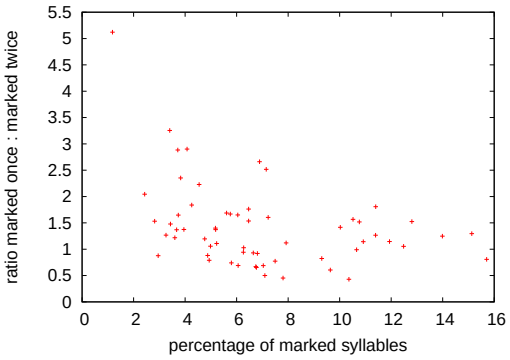

Figure 1: *Inconsistency (vertical axis – smaller values indicate more consistent listeners) and intolerance (horizontal axis – smaller values indicate more tolerant listeners) of individual listeners.*

3. Classification Experiments

3.1. Input Feature Vectors

Most of our classifiers were trained to predict the perceptual acceptability of *phones*. Thus, each feature vector comprises the set of 12 context features used in our standard target cost for the target phone, the left candidate demiphone and the right candidate demiphone.

For some of the classifiers we experimented with, the feature space must be real-valued. Therefore, we transformed the categorical features into a 1-of-n numerical encoding (n being the number of possible values for that feature). For phone identities we used a compact feature description: the first feature is "vowel or consonant". Vowels are described by 4 numerical features for frontness, height, rounding and length (which take a default value for consonants). Consonants are described by 14 binary features which fall into 3 orthogonal categories: place of articulation (7), consonant type (6) and voicing (1).

We also added features to indicate which context features mismatch between target and candidate, because this improved performance in some cases. The length of the resulting feature vector is 231.

For integration with Festival's existing unit selection module, it is necessary to place a target cost on each *demiphone*, not each phone, since the basic unit employed is the diphone. The input features then consist of the context features for target and candidate demiphone plus the added mismatch features; after converting categorical to numeric features the feature vector length is 151.

3.2. Target labels

The label of each phone or demiphone is simply taken from the syllable it belongs to. This is a pessimistic assumption: that every constituent phone or demiphone in a bad syllable is itself bad. Listener scores above 5.5 (on a scale of 0 to 10) lead to the label "bad" and scores lower than this lead to the label "good".

In early experiments using the "Wagon" tool from the Edinburgh Speech Tools Library [7] to learn Classification and Regression Trees (CARTs), we investigated the effect of using a threshold other than 5.5 but could not improve results. Higher

threshold values mean that the phones or demiphones labelled as "bad" are more likely to be consistently bad, but skews the data and reduces the already small number of "bad" examples in the training set. Attempts to account for individual listener's inconsistency and intolerance also did not improve the classifier's accuracy. Dropping the 50% least consistent listeners did give small improvements, but again at the cost of data sparsity. Subsequent experiments therefore used all the data and a threshold of 5.5.

3.3. Comparison and Tuning of Classifiers

In the main experiment, we compared three classifiers: Classification Tree (a CART implemented using Wagon), Timbl [8], which is basically an n-next-neighbourhood classifier, and Support Vector Machines (implemented using SVM Light [9]).

Settings and parameters were optimised semi-automatically, although computational cost limited the amount of tuning that was possible. Table 1 presents the best results for CART, Timbl and SVM.

accuracy (%)	CART	Timbl	SVM
phone /t/	64.1	63.0	71.5
all phones	63.9 ± 8.2	56.2 ± 3.0	66.6 ± 7.5

Table 1: *Comparison of three different classifiers. The means and standard deviations here were calculated over the results for individual phones (rather than over 9 folds).*

The best result were for the SVMs and were achieved using a polynomial kernel of order 2 and parameter $s = 0.06$. Since training time grows quadratically with data size, we trained classifiers for each individual phone in parallel and combined the test results afterwards. By splitting the training data into vowels and consonants i.e. training only two classifiers, the accuracy increased from 66.6 ± 7.5 to $71.9 \pm 1.5\%$ – see Figure 2.

3.4. Size of Training Data

Collecting perceptual data is rather slow and expensive, so it is important to consider how much data is required. Figure 2 shows the accuracies of the SVMs in two test conditions: closed-set testing on the training data ("learn==test") and open-set testing with unseen test data ("learn!=test").

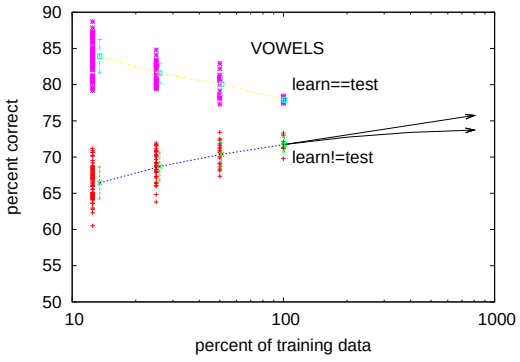

Figure 2: *Effect of training data size on classification accuracy. Arrows point to the estimated range of accuracy at 8 times the amount of training data.*

The errors bars show the means and standard deviations computed over the folds of the data (9 folds for all data, 18 for 50% of the data, 36 for 25% and 72 for 12.5%). As expected, the closed and open set accuracies converge with increasing amounts of training data. The figure suggests that even with the limited data we used, we are surprisingly close to convergence. 72 person hours of perceptual data does not seem very much, considering the vast number of possible context feature value combinations.

4. Perceptual Evaluation of the New Target Cost Classifier

After learning a classifier from the perceptual data and evaluating it using held-out data, we proceeded to implement it within the Festival speech synthesiser, to replace the standard weighted-sum-of-factors target cost. Two sets of test sentences were synthesised: the 400 Blizzard 2008 test sentences [10] and 540 short sentences (two and four words) of the same type as the training data.

4.1. Integration with Festival

As the units of speech in Festival are diphones, we needed classifiers for left and right demiphones, just as Festival's default target cost components check for both demiphones constituting a diphone. Therefore we trained SVMs with the perceptual data split four ways: left/right demiphones of vowels/consonants.

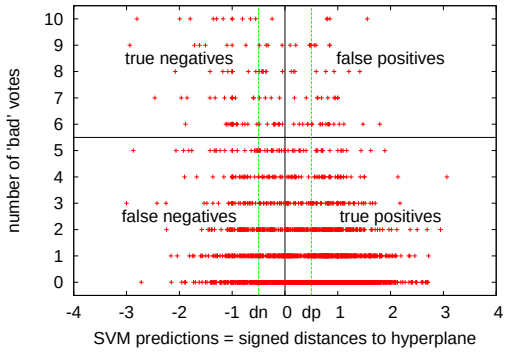

Figure 3: *SVM predictions for vowels, fold 1 of 9. The prediction consists of the signed distance to the hyperplane; the sign is the class label. Correlation between perceptual badness and SVM prediction (positiveness) is -0.340215. Later, a third class "fair" between "good" and "bad" was defined as signed distances being between dn and dp.*

For simplicity of integration with the existing Festival unit selection architecture, the output of the classifier was transformed into a continuously-valued cost.

As noted in Section 2.2, we excluded stimuli in which the context features "emphasis" or "boundary tone" mismatch, because these are always perceptually bad. If a mismatch is found in one of these features, a very large target cost of 999 ("very bad") is assigned and the SVM is not used. Otherwise, the output of the appropriate SVM is transformed to a cost of either 0 ("good") or 99 ("bad"). These costs are somewhat arbitrary but achieve our primary goal of avoiding very bad units and secondary goal of maximising the number of good units.

Although correlation between distance to the hyperplane and perceptual badness is weak, some gain may be had by using an intermediate class of "fair" between "good" and "bad"; we

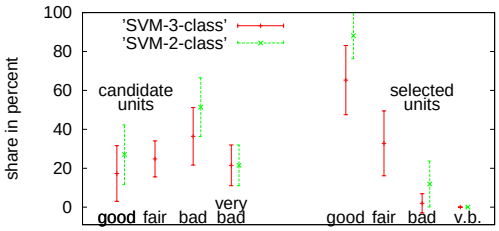

Figure 4: *Distribution of the class labels as means and standard deviations, calculated over the 600 test sentences. The signed distance to the hyperplane is translated into either 2 classes (good/bad) or 3 classes (good/fair/bad). The "very bad" class label is assigned upfront when the emphasis or boundary tone feature mismatches.*

defined this class as signed distances between $dn = -0.5$ and $dp = 0.5$ and assigned it a target cost of 9. In a pilot preference listening test, this 3-way classification was preferred over the 2-way classification. Figure 4 shows the distribution of the class labels whilst synthesising 600 sentences, for both the candidate and selected units, and for both the 2-way and 3-way classifiers.

We also investigated the effect of the target cost on the join cost distribution. Figure 5 shows that the distributions are quite similar. The largest difference is the frequency of zero costs (which are assigned when two units are consecutive in the data base) between 2-way and 3-way classification. In other words, slightly more concatenations take place when using the 3-way classification.

4.2. Listening Test

40 listeners were each presented with all 400 Blizzard 2008 test sentences in random order (average length 8.4 words; maximum 15 words) in pairs consisting of the same sentence synthesised using Festival with the default target cost and with the new target cost. Order of the pairs was randomised and the stimuli were presented via a web page interface which allowed them to listen repeatedly if they wished. All used headphones and were native English speakers. The listeners were asked to base forced preference on overall quality. No further instructions were given. On average, the listeners preferred the default target cost 53.4% of the time. The difference is statistically significant, but small.

To examine the affect of the test sentence type, we conducted a second listening test, using 540 short (2-4 words) sentences of the same type as the training sentences. Each listener was presented with half of the stimuli, repeated twice, in randomised order. The average consistency (i.e. whether a listener expressed the same preference for both repetitions of a pair) was 68.2% with a standard deviation of 6.5% and a range from 52% to 83%. The average preference for the default target cost was 54.3%, almost the same as for the previous test set. The correlation between the listener's consistency and their preference for

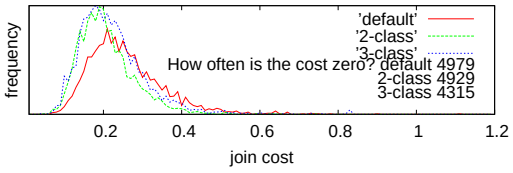

Figure 5: *Effect of the target cost on the join cost distribution. The number of joins with a cost of 0 is not plotted on the graph but instead given in the key.*

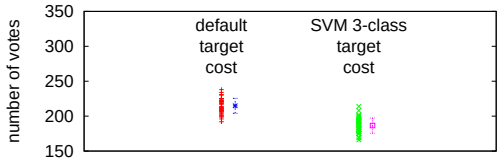

Figure 6: *The 400 Blizzard-2008 test sentences were synthesised with Festival's default and new target costs. 40 listeners were asked for their preference. The average preference for the new target cost is 46.6%.*

the default target cost is very low (0.27). We conclude that the type of test sentence has no effect.

5. Discussion

The automatically-learned target cost is nearly as good as our manually-tuned standard target cost. Considerable expertise and time is required to tune a standard target cost; our method, employing a classifier, is automatic and eliminates this effort, albeit at the expense of collecting perceptual data.

There are several advantages of using a classifier trained on perceptual data, compared to a weighted-sum-of-factors cost function. It would be straightforward to incorporate any number of additional factors into the classifier whereas adding new factors to a conventional cost function involves retuning the weights; there is also a limit on how many weights can realistically be tuned by hand. The classifier approach also requires less expertise and therefore this approach would be useful when building voices in languages where speech synthesis expertise is scarce, but where native listeners are available.

Acknowledgements: At the time this work was carried out, VS was funded by EPSRC grant EP/E031447/1 and SK held an EPSRC Advanced Research Fellowship. This work has made use of the resources provided by the Edinburgh Compute and Data Facility which is partially supported by the eDIKT initiative (http://www.edikt.org.uk).

6. References

[1] Strom V. and King S.,"Investigating Festival's Target Cost Function using Perceptual Experiments", Proc. Eurospeech, 2008.

[2] Clark R., Richmond K. and King S.,"Multisyn voices from ARCTIC data for the Blizzard challenge", Proc. Interspeech, 2007.

[2] Hunt A. and Black A., "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database", Proc. ICASSP 1996.

[3] Robert E. Donovan: Trainable Speech Synthesis, Cambridge University 1996, citeseer.ist.psu.edu/donovan96trainable.html

[4] M. Macon and A. Cronk and J. Wouters: Generalization and discrimination in tree-structured unit selection, Proc. 3rd Synthesis Workshop, 1998.

[5] Black A. and Taylor .P, "Automatically Clustering Similar Units for Unit Selection Synthesis", Proc. Eurospeech, 1997.

[6] A. Syrdal and A. Conkie, "Perceptually-Based Data-Driven Join Costs: Comparing Join Types", Proc. Interspeech, 2005

[7] The Edinburgh Speech Tools Library. http://www.cstr.ed.ac.uk/projects/speech_tools

[8] Tilburg Memory-Based Learner. http://ilk.uvt.nl/timbl

[9] SVMLight Support Vector Machine. http://svmlight.joachims.org

[10] http://festvox.org/blizzard/blizzard2008.html