AUTOMATIC ARCHITECTURE DESIGN BY LIKELIHOOD-BASED CONTEXT CLUSTERING WITH CROSSVALIDATION

Ivica Rogina
Interactive Systems Labs
University of Karlsruhe, Am Fasanengarten 5, 76131 Karlsruhe, Germany
E-mail: rogina@ira.uka.de

ABSTRACT

Most state-of-the-art speech recognizers benefit from some kind of context information in their acoustic modeling [1][2][3]. The most common approach to context clustering is a divisive method that is iteratively building decision trees [4][5]. The problem, when to stop the growing of the tree is usually solved by choosing the maximum number of resulting models that can be supported by the available training data and/or computer memory and CPU power. In this paper we propose a new algorithm, that not only offers an optimized stopping criterion, but also uses a likelihood-based distance measure that optimizes the likelihood of unseen training-data at every splitting of a decision tree node. We evaluate our algorithm on the Wall Street Journal task, and show that it outperforms an algorithm using an entropy-based distance measure.

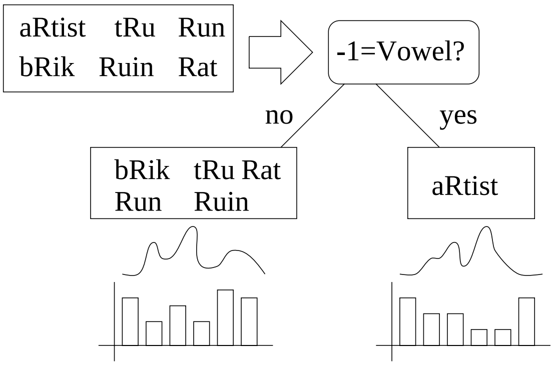
1. INTRODUCTION

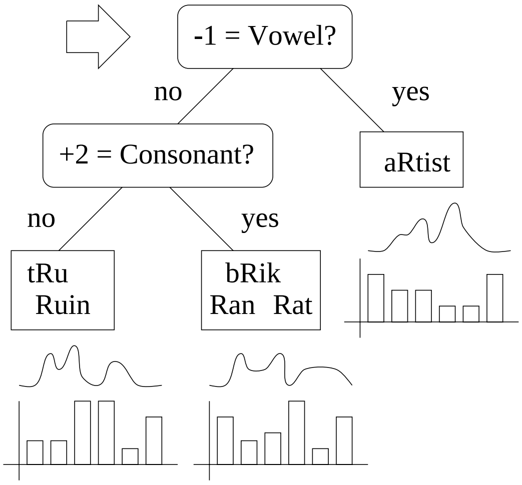
When in the near future commercially available speech recognizers will hit the market, customers will be interested in adjusting the product to their own specialized needs, to the acoustic environment, the vocabulary and the language model of their preferred scenario. The task of building a recognizer that matches their needs will be performed by engineers, not by speech scientists. Therefore it will be necessary, that many design variables of speech recognizers can be optimized automatically. Two of these design variables are the way of acoustic parameter tying and the size of the parameter space.

2. CONTEXT CLUSTERING IN JANUS

JANUS [6] uses a two stage context clustering algorithm [2] that builds context querying decision trees. The atomic elements that are clustered are subpolyphones. Before we start clustering we train a subphonetically tied semicontinuous HMM with typically three Gaussian codebooks per phoneme and a mixture weight distribution for each of the several hundred thousand subpolyphones.

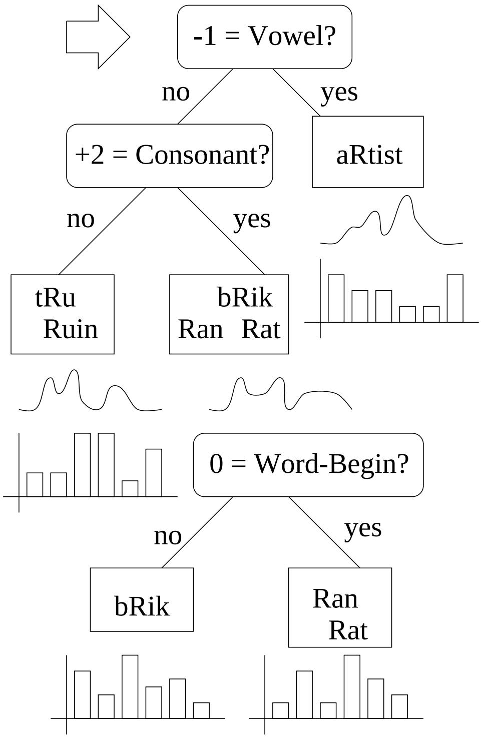
In the first stage we grow a decision tree until it reaches the number of desired leaf nodes (typically a few thousand, depending on the number of the available training data). We only allow splits such that every successor of a split node gets a minimum amount of training data. The following figure shows an example tree after one and two clustering steps for different contexts of the phone R.





After that a fully continuous Gaussian mixture model is trained for every leaf node, and a new mixture weight distribution based on the corresponding codebook is estimated for each subpolyphone.

In the second clustering phase, we continue growing the decision tree and eventually train a separate mixture weight distribution for each of the resulting leaf nodes:



3. ENTROPY-BASED DISTANCE

So far, the JANUS speech recognizer[1] and others [3][5] have used an entropy-based distance measure. When judging the benefit of splitting a decision tree node N into two nodes Q and R according to a given dividing context question, the distance $D_E(Q,R)$ of Q and R is defined by the resulting information gain in the two mixture-weight distributions γ_Q and γ_R over the common distribution γ_N:

$$D_E(Q,R) = n_Q \cdot H_Q \, + \, n_R \cdot H_R \, - \, n_N \cdot H_N$$

where the n_i are the training counts of the models, and the H_i are the entropies of the models:

$$H_i = \sum_k \gamma_i(k) \cdot \log \gamma_i(k)$$

It is obvious that with this distance measure, every split of a tree node will result in a non-negative information gain. In general, we observe positive information gains throughout the entire growing of the decision tree.

The preferred way of stopping the tree growing process is by defining a number of desired resulting leaf nodes, or by defining a minimum entropy decrease. Both ways need some kind of educated guess that meets the constraints given by the available training data and computing power. As long as training data abound, the computing power will be a limiting factor, otherwise the greatest number of models that can be trained well enough is usually chosen to get the best recognition performance. Smoothing techniques [3] are available for recognizers that use very poorly trained models, but an algorithm that stops automatically is still desirable.

4. LIKELIHOOD-BASED DISTANCE

Optimizing the information content of the acoustic parameters certainly is a well motivated way of training. But what we actually would like to achieve is to maximize the likelihood of the training data in accordance with the EM-algorithm. Here too, we could increase this likelihood up to its maximum if we just used a huge amount of models and parameters. The likelihood of some crossvalidation data would generally not increase beyond some overfitting point. Adding more parameters will then only increase the likelihood of the training data, while the likelihood of the crossvalidation data will decrease.

In the following we propose an algorithm, that computes the likelihood increase on a crossvalidation set that is achieved by splitting a particular tree node. The decision tree growing algorithm looks as follows:

1. start with a single node
2. out of all possible splits, pick the one that would give the greatest increase in likelihood if applied
3. if the best increase in likelihood is greater zero apply the split
4. while splitting is still possible goto step 2

The definition of the likelihood increase is as follows. Let the HMM emission probability for observing x when in model s be

$$p(x|s) = \sum_k \gamma_s(k) \cdot G_k(x)$$

where γ_s is the mixture weight distribution for the atomic model s, and $G_k(x)$ is the k-th Gaussian value of model s at x. G_k is assumed to be independent of the model s, i.e. all models that can be clustered together use the same codebook (SCHMM). Let's assume a tree node N represents a set S_N of atomic models. Splitting N results in two successor nodes Q and R, such that $S_Q \cup S_R = S_N$. Now, let γ_N^A, γ_Q^A, and γ_R^A be the mixture weight distributions of the nodes N, Q, and R if trained on the training subset A. Let B be the other subset of the training data. Then the likelihood of B given the models trained on A before the split is:

$$L_{N,A}(B) = \prod_{x \in B} \sum_k \gamma_N^A(k) \cdot G_k(x)$$

Thus, the likelihood after the split is

$$L_{Q,A}(B) \cdot L_{R,A}(B)$$

We now define our distance measure as $D_L(Q,R) =$

$$\frac{L_{Q,A}(B) \cdot L_{R,A}(B)}{L_{N,A}(B)} \cdot \frac{L_{Q,B}(A) \cdot L_{R,B}(A)}{L_{N,B}(A)}$$

In this definition we have used two subsets of the training data. It is also possible to use more than two subsets, say $C_1 \dots C_K$, and select each one as the crossvalidation set. Then we would use the "leaving-one-out" or "round-robin" method to compute the distance measure:

$$D_L(Q,R) \quad = \quad \prod_{j=1}^{K} \frac{L_{Q,C \setminus C_j}(C_j) \cdot L_{R,C \setminus C_j}(C_j)}{L_{N,C \setminus C_j}(C_j)}$$

where $C = \bigcup_j C_j$. If we replace the computation of the HMM emission probability by

$$p(x|s) = \gamma_s(m) \cdot G_m(x) \quad \text{where} \quad m = \operatorname*{argmax}_k G_k(x)$$

and if we compute the likelihood distance without a crossvalidation set then we get the interesting fact that:

$$D_E(Q,R) = D_L(Q,R) = \frac{L_{Q,C}(C) \cdot L_{R,C}(C)}{L_{N,C}(C)}$$

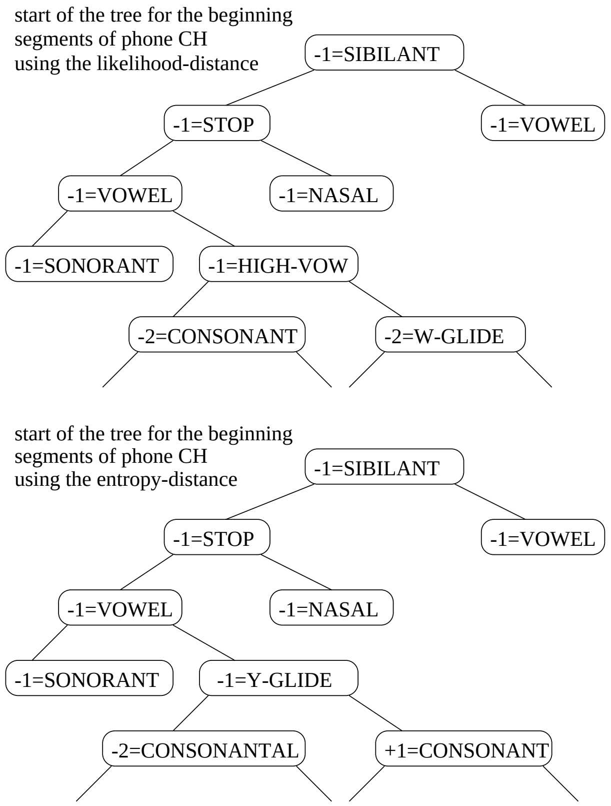
5. EXPERIMENTS

We have conducted experiments on the Wall Street Journal task and evaluated JANUS on the November 1994 evaluation set (our currently best performing system on this test set has an error rate of 7.7%). Three baseline systems were trained and the tree growing algorithm using the described entropy distance was stopped after 2000, 3000, and 5000 leaf nodes. When using the likelihood-distance with two crossvalidation sets, the algorithm stopped after 4585 leaf nodes. All tested systems did not allow splits that would create models with less than 1000 training samples. The following table summarizes the error rates of the four systems:

entropy			likelihood
2000	3000	5000	4585
14.0%	13.3%	11.6% %	10.8%

We can see that the recognition accuracy of the likelihood-clustered system is better than all of the entropy-clustered systems.

We have observed that the likelihood-clustered trees and the entropy-clustered trees usually start very similar and diverge after several splits. A typical similarity is shown in the following figure for the beginning segments of the phoneme CH.



6. CONCLUSION AND FUTURE PLANS

We have shown that with our new clustering algorithm it is possible to not only build a context decision tree optimizing the likelihood of the training data, but also to automatically determine a reasonable size of the acoustic parameter space. We have found that the top ranking questions in the decision trees are similar for both distance measures. The system that was clustered with the likelihood-distance with crossvalidation outperformed different systems clustered with the entropy distance.

Problems yet to be addressed are the question about the effect of using different numbers of crossvalidation sets. A comparison of the performance of entropy-clustered systems and likelihood-clustered systems with the same number of leaf nodes, should give us more insight into the separate benefits of the likelihood distance measure and the usage of cross-validation sets.

REFERENCES

[1]
Rogina I., Waibel A.: “The JANUS Recognizer”, ARPA Workshop on Spoken Language Technology, 1995

[2]
Finke M., Rogina I.: ”Wide Context Acoustic Modeling in Read vs. Spontaneous Speech”, Proceedings of the ICASSP 1997, Munich

[3]
Lee K-F.: ”Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System”, PhD Thesis, Carnegie Mellon University, Pittsburgh, internal report CMU-CS-88-148

[4]
S.J. Young, J.J. Odell, P.C. Woodland: ”Tree-Based State Tying for High Accuracy Acoustic Modeling”, ARPA Workshop on Human Language Technology, 1994

[5]
Hwang, M.Y.: ”Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition”, Ph.D. Thesis, Carnegie Mellon University, 1993

[6]
A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavalda, T. Zeppenfeld, P. Zhan: ”JANUS-III: Speech-to-Speech Translation in Multiple Languages”, Proceedings of the ICASSP, April 1997, Munich, vol. 1, pp. 99-102

[7]
T. Zeppenfeld, M. Finke, K. Ries, M. Westphal, A. Waibel: ”Recognition of Conversational Telephone Speech using the Janus Speech Engine”, Proceedings of the ICASSP, April 1997, Munich, vol. 3, pp. 1815-1818