Chapter 13

Speech recognition

Mikko Kurimo, Panu Somervuo, Vesa Siivola, Teemu Hirsimäki

13.1 Acoustic modeling

Acoustic modeling in automatic speech recognition (ASR) means building statistical models for subword units based on the feature vectors computed from speech. Feature representation is an important part of any pattern recognition system and ASR is no exception. It is difficult to develop any theoretically optimal feature extraction method which would minimize the recognition error. Usually the discriminative training is applied to the estimation of the model parameters and the feature representation is more or less fixed, see e.g. [1]. In practice, several feature extraction methods have been experimented and during the long history of ASR, some of them have been experimentally proved to be more beneficial than others.

In most systems the speech signal is first chunked into overlapping 20-30 ms time windows at every 10 ms and the spectral representation is computed from each frame. A commonly used feature vector consists of mel-frequency cepstral coefficients (MFCC) which are the result of the discrete cosine transform (DCT) applied to the logarithmic mel-scaled filter bank energies. Local temporal dynamics can be captured by concatenating the first and second order delta features (time differences) to the basic feature vector. Computation of delta features can be considered as a fixed transformation to the block of original feature vectors. We have experimented also other linear and nonlinear feature transformations.

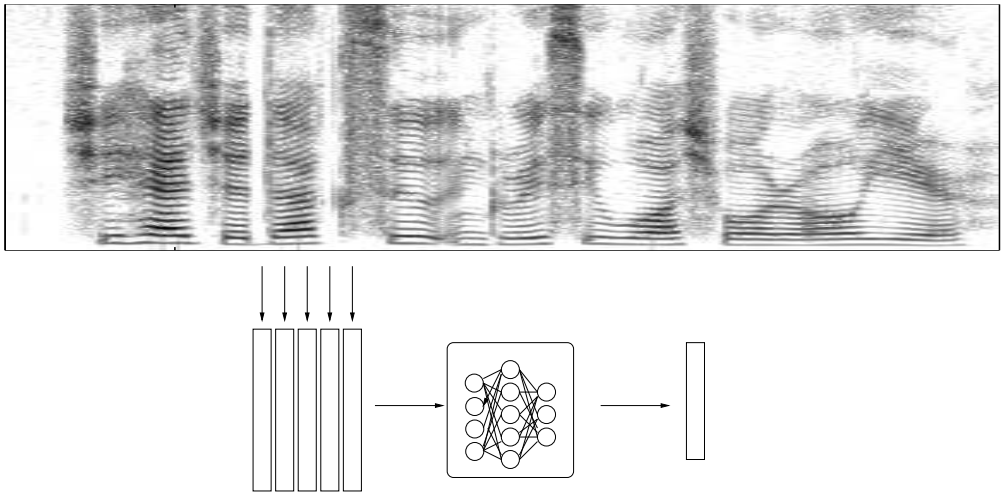

Figure 13.1: Feature transformation. One or more frames (five in this figure) original feature vectors, e.g. logarithmic mel-spectra are fed to the linear (matrix) or nonlinear (MLP network) feature transform which performs the projection of the original feature vector (or concatenation of them) to the new feature space. The output is used as a feature vector in the mixture-of-Gaussians based HMM system.

In our experiments, unsupervised transformations were based on principal component analysis (PCA) and independent component analysis (ICA) and discriminative transformations were based on linear discriminant analysis (LDA) and multilayer perceptron (MLP) networks. These transformations were experimented in TIMIT phone recognition [2] where clear improvements were gained in the recognition rate compared to the baseline MFCC feature. In another experiment [3], the acoustic models were trained using 60 hours of HUB5 training data and they were tested using OGI Numbers corpus. The combination of the PLP cepstrum and the MLP network based feature transformation stream gave the best result. The baseline word error rate was reduced from 4.1 % to 3.1 %.

Currently used speech recognizers are typically based on hidden Markov models where HMM states are modeled by Gaussian mixtures. In order to avoid the large number of parameters in the model, the covariance matrices of the Gaussians are diagonal. We have experimented the maximum likelihood linear transformation (MLLT) [4], which takes the diagonal Gaussian assumption into account when forming the transformation. The result is not the global PCA transform, since in our case the data is not modeled by a single Gaussian with a single covariance matrix but each speech unit is modeled by its own mixture of Gaussians where the diagonal covariance matrices need not be the same. Using the MLLT framework, feature transformations based on heteroscedastic linear discriminant analysis (HLDA) can also be constructed. Contrasted to the basic LDA, HLDA does not assume equal class covariance matrices. Applying these transformations to Finnish speech recognition system gave very promising results:

	monophone HMMs		triphone HMMs	
	letter error %	word error %	letter error %	word error %
baseline MFCC	11.0	44.0	4.7	24.8
MFCC+MLLT	9.0	40.2	4.5	24.1
MFCC+HLDA	8.4	37.5		

Besides speech recognition, we have also investigated methods for representing high-dimensional feature vectors. In [5] it was studied how to capture the intrinsic dimensionality of speech using fractal-dimensionality measure, multi-dimensional scaling, and hypercubical Self-Organizing Map. These results can give insights to the data being modeled and that way contribute also to the developments in speech recognition.

State-of-the-art speech recognizers are complex systems with large number of parameters. This raises the challenge how to get robust estimates and what is the optimal number of model parameters. One elegant way is to use Bayesian modeling. In [6], standard maximum likelihood (ML) estimation was compared to the variational Bayesian approach for training mixtures of Gaussians. Advantages of Bayesian approach were clear: estimation converged faster, there was no tendency of overfitting, and the likelihoods of unseen test data were better for any given number of mixture components.

References

[1] P. Somervuo. Two-level phoneme recognition based on successive use of monophone and diphone models. In *Proc. EUSIPCO*, vol. 3, pages 77–80, 2002.

[2] P. Somervuo. Experiments with linear and nonlinear feature transformations in HMM based phone recognition. In *Proc. IEEE ICASSP*, vol. 1, pages 52–55, 2003.

[3] P. Somervuo, B. Chen, and Q. Zhu. Feature transformations and combinations for improving ASR performance. In *Proc. Eurospeech*, vol. 1, pages 477–480, 2003.

[4] M. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Tr. SAP*, 7(3), pages 272–281, 1999.

[5] P. Somervuo. Speech dimensionality analysis on hypercubical self-organizing maps. *Neural Processing Letters*, 17(2), pages 125–136, 2003.

[6] P. Somervuo. Speech modeling using variational Bayesian mixture of Gaussians. In *Proc. ICSLP*, pages 1245–1248, 2002.

13.2 Language modeling

Language model unit selection for speech recognition

The traditional method to model language for speech recognition is the n-gram model. The probability of a new word is estimated based on a few previous words. For Finnish, estimating the n-gram probabilities is difficult, since there is a vast number of different word forms. For example, a single verb has theoretically thousands of inflected word forms. We have chosen to split the words to smaller units to have fewer probabilities to estimate and to cover a larger vocabulary. As subword units we have evaluated syllables and statistically found morpheme-like units [1].

For Finnish, words can be split into syllables based on a few simple hyphenation rules, except on boundary between parts of a compound word. Our algorithm implements the simple ruleset and makes infrequently mistakes on compound words. A morpheme is the smallest meaning bearing element of a word. We have used an automatic statistical method for finding morpheme-like units, called morphs (see Section 12.1).

In our evaluations, using syllables for language model units decreases the recognition word error rate 22% relative to word based model. Using morphs reduces the word error rate 44% relative to word based model. The morphs seem to be better suited for language modeling, since each morph has a distinct meaning which is useful for language modeling. For syllable and morph based models, we have another advantage: we do not need to know all of the words of Finnish language, since the words can be constructed from the smaller pieces.

To assess the language-independence of the word splitting method, we applied the same algorithm to Turkish, which is another agglutinative language[1]. To compare the performance with baseline speech recognizer, the n-gram models were trained both to these new data-driven and old rule-based morphemes and words. The data-driven morphemes achieved clearly the lowest error rates in all large-vocabulary continuous speech recognition tests made [2]. The work with Turkish data is done in a close collaboration with the University of Colorado in Boulder and the Middle East Technical University.

Focusing language models in speech recognition

The efficient language processing tools developed in the laboratory (WEBSOM) have been applied to organize language models based on the topical structure of the discourse [3]. The objective is to increase the language modeling accuracy and to obtain improved speech recognition results by automatically detecting and focusing into the best available language model for the recognition task at hand. This work is done in a close collaboration with the Natural language modeling group (see Section 12.3).

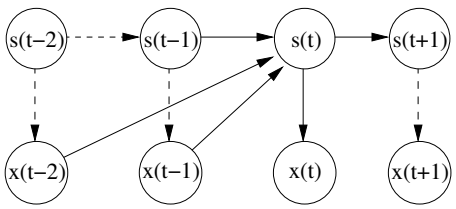

Figure 13.2: The state-space model for language modeling. $\mathbf{s}(t)$ is the current state and $\mathbf{x}(t-1)$ the previous observation.

[1] In agglutinative languges words frequently have multiple suffices concatenated one after each other.

State-space method for language modeling

The most common language model for speech recognition is the n-gram model. With back-off and smoothing, it provides a relatively robust model.However, the n-gram model cannot generalize from similar words: seeing a phrase like "Monday morning was clear" does not help modeling the phrase "Tuesday evening is cloudy" at all. This kind of generalization can be achieved by clustering similar words together and interpolating this cluster n-gram with a regular n-gram.

We have tried to achieve the generalization by mapping the words to n-dimensional feature space, so that similar words are mapped close to each other. The probability of a word is calculated as a smooth function of the features and the previous state, leading to good generalization. This kind of approach with neural networks has been shown to yield good results [4]. The mathematics of our method are based on linear state-space modeling, which is also used in famous algorithms like Kalman filtering. We have added explicit dependencies to previous observations to make the teaching of the model simpler (see Fig. 13.2).

During the first experiments, we simply tried predicting the next letter based on previously seen letters, since the learning algorithm was computationally extremely demanding [5]. We are currently working on making the algorithm suitably fast for word prediction. Figure 13.3 shows a hypothetical idealized picture of both the feature and the internal state of the model.

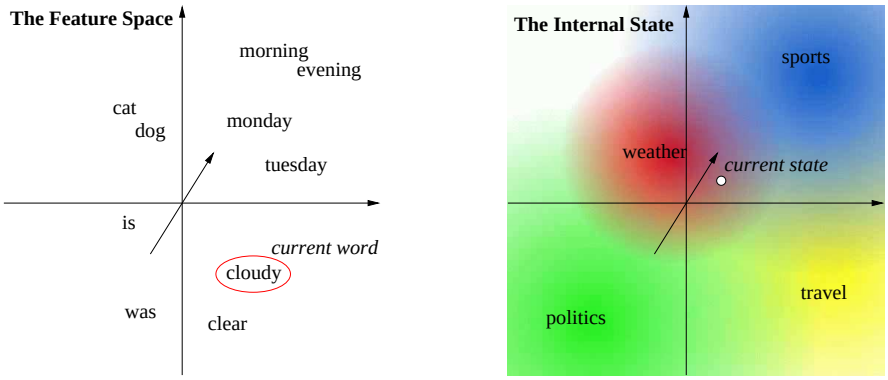

Figure 13.3: The ideal state-space language model. On left is the feature space and on right the internal state of the model.

References

[1] V. Siivola, T. Hirsimäki, M. Creutz and M. Kurimo: Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner, *Eurospeech03*, pages 2293-2296, 2003.

[2] K. Hacioglu, B. Pellom, T. Ciloglu, O. Ozturk, M. Kurimo, and M. Creutz. On Lexicon Creation for Turkish LVCSR. *Eurospeech03*, pages 1165-1168, 2003.

[3] M. Kurimo and K. Lagus. An efficiently focusing large vocabulary language model. *ICANN'02*, pages 1068–1073, Madrid, Spain, 2002.

[4] Y. Bengio, R. Ducharme and P. Vincent, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, February 2003.

[5] V. Siivola and A. Honkela: A state-space method for language modeling, *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2003

13.3 Large vocabulary decoder

The task of the decoder in a speech recognition system is to combine the probabilities and rules given by all the model component to find a word sequence that matches best with the given speech. In order to do this, the decoder should, in principle, consider all possible word sequences, and score them using the acoustic and language models. However, because the number of possible word sequences is extremely large even with small vocabularies, the decoder must concentrate the search effort on the most promising words and prune the improbable sequences in an early stage.

During the past two years, we have been actively developing a large vocabulary decoder [1,2]. Instead of using whole words as recognition units, as traditional speech recognition systems do, our decoder constructs words from smaller units, called morphs. This makes it possible to recognize very large vocabularies with a reasonable number of units, which is important in Finnish, especially. Because natural speech is continuous and does not contain clear word boundaries, the decoder has to consider a possible word boundary after every morph, and use language models to evaluate where the word boundaries are most probable. The decoder puts the combined word sequences in stacks according to their ending times, and only the best sequences are stored for each time instant. In this stack decoding approach, complex language models can be used without hindering the acoustic matching, but the dependence between acoustic models is harder to take into account.

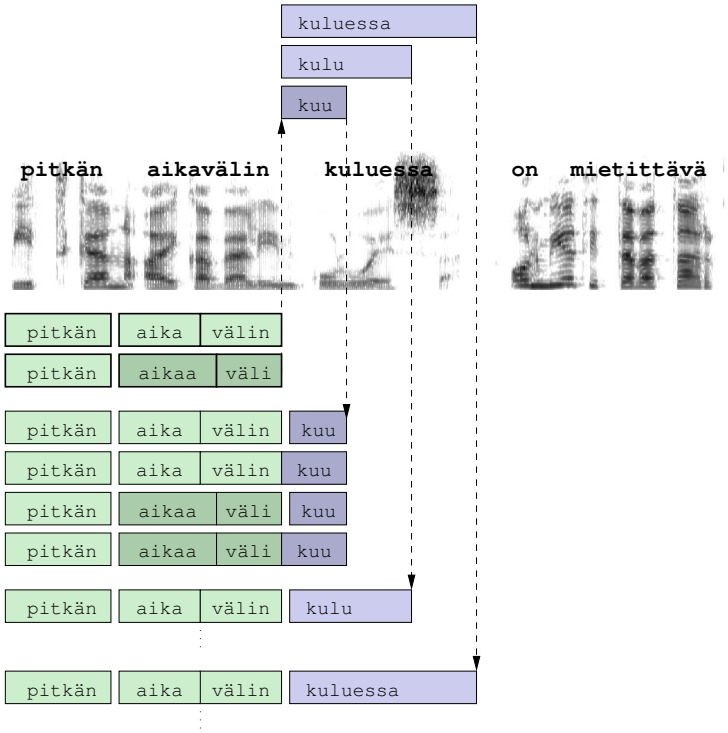

Figure 13.4: The stack decoder expands two hypotheses (bold green boxes) with three acoustically promising most morphs (blue boxes).

References

[1] T. Hirsimäki, "Decoder design for large vocabulary continuous speech recognition system," M.S. thesis, Helsinki University of Technology, Espoo, Finland, 2002.