Code Breaking for Automatic Speech Recognition

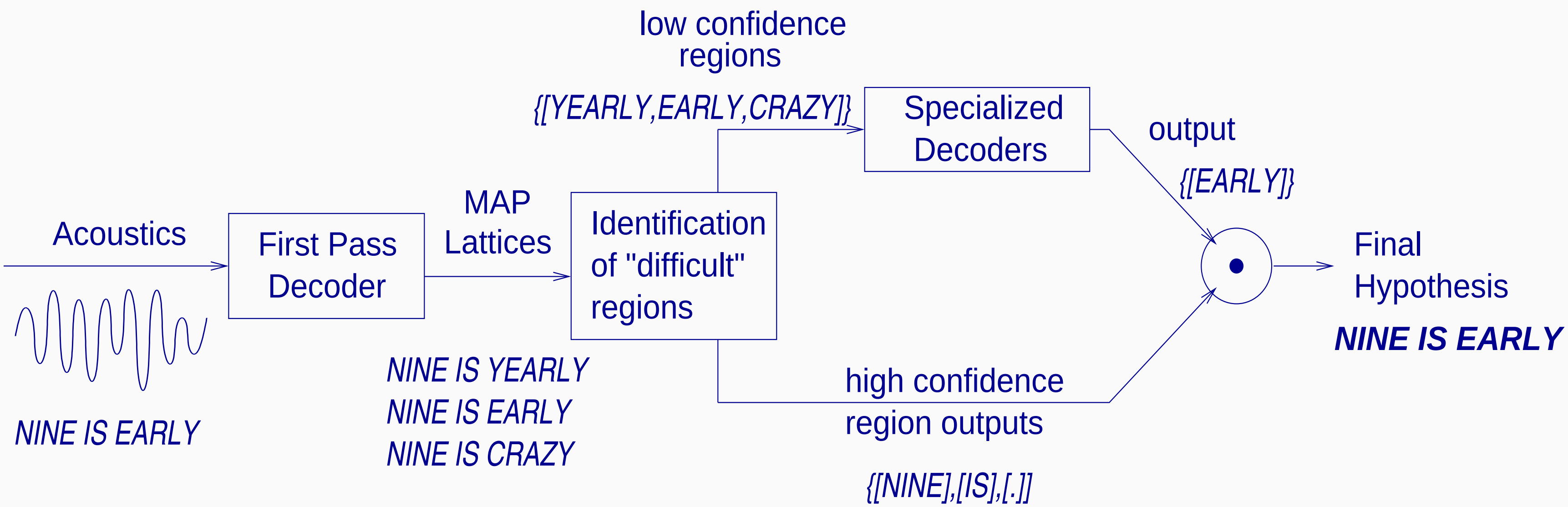
A Dissertation Defense

Veera Venkataramani
Advisor: Prof. William J. Byrne
Committee: Prof. Gert Cauwenberghs,
Prof. Gerard. G. L. Meyer &
Prof. Frederick Jelinek.

Department of Electrical and Computer Engineering,
Center for Language and Speech Processing,
The Johns Hopkins University,
March 25, 2005.

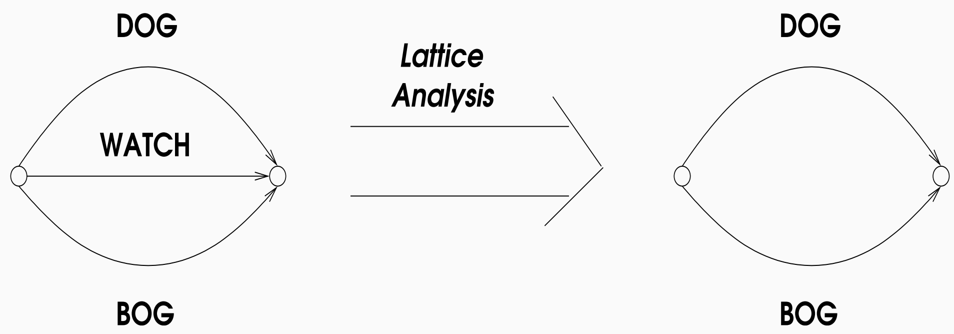
Code Breaking for ASR

- A divide-and-conquer approach.
- Attempt to find and fix weaknesses of a baseline speech recognizer.
- It involves:
  - An initial decoding pass to produce a search space of hypotheses.
  - Identification of “difficult” regions in the hypothesis space.
  - Resolving these confusions with specialized models.



Motivation

- We will improve upon the performance of a state-of-the-art HMM system.
- Framework for trying out novel ASR techniques without losing the benefits of HMMs.
- Allows the use of simple and powerful classifiers that would otherwise have not been appropriate, *e.g.*, Support Vector Machines.
- Different word recognition problems require different types of decoders.



New Framework

We propose using

- HMMs as our first-pass system
- Lattice cutting techniques as a means to identify regions of confusion.
- Both HMMs and Support Vector Machines (SVMs) as specialized models to resolve the remaining confusion.

Related Prior Work:

- Speech Recognition as Code Breaking [F. Jelinek, '95]
- ACID-HNN [J. Fritsch *et al*, '96]
- Consensus Decoding [L. Mangu et. al, '99, G. Evermann *et al*, '01]
- Corrective Training [L. Bahl, *et al*, '93]
- Boosting [Schapire *et al*, '95]
- Confusion Sets [Fine *et al*, '01]

Outline

- Statistical Speech Recognition
- Identification of Confusions
- SVMs for Continuous Speech Recognition
- Validation on a Small Vocabulary task
- Feasibility for Large Vocabulary tasks
- Conclusions and Future Work

Statistical Speech Recognition

- Goal: Determine the word string $\hat{W}$ that was spoken based on acoustics A.
- Maximum A Posteriori (MAP) Recognizer formulation:

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|A). \tag{1}$$

- Applying Bayes Rule,

$$P(W|A) = \frac{P(A|W)P(W)}{P(A)}.$$

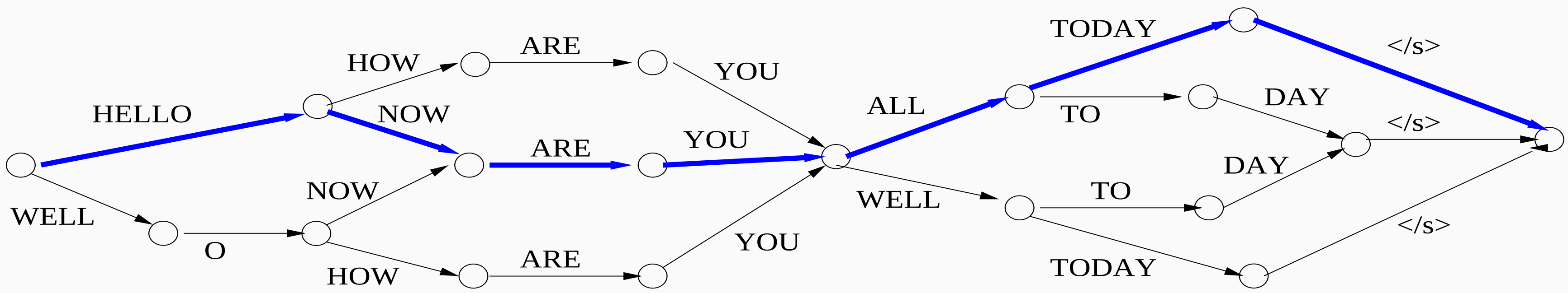
- Since the search in Eqn. 1 is independent of A, we have

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(A|W)P(W).$$

$P(A|W)$ is estimated using an *acoustic model*, usually an HMM. $P(W)$ is estimated using a *language model*.

Notations

- Evaluation Criterion: Word Error Rate (WER)= string-edit distance between hypothesis and the truth
- Lattice: A compact representation of most likely hypotheses, with associated acoustic segments.



- Lattice Word Error Rate=the WER of the lattice hypothesis with lowest WER.

< > - +

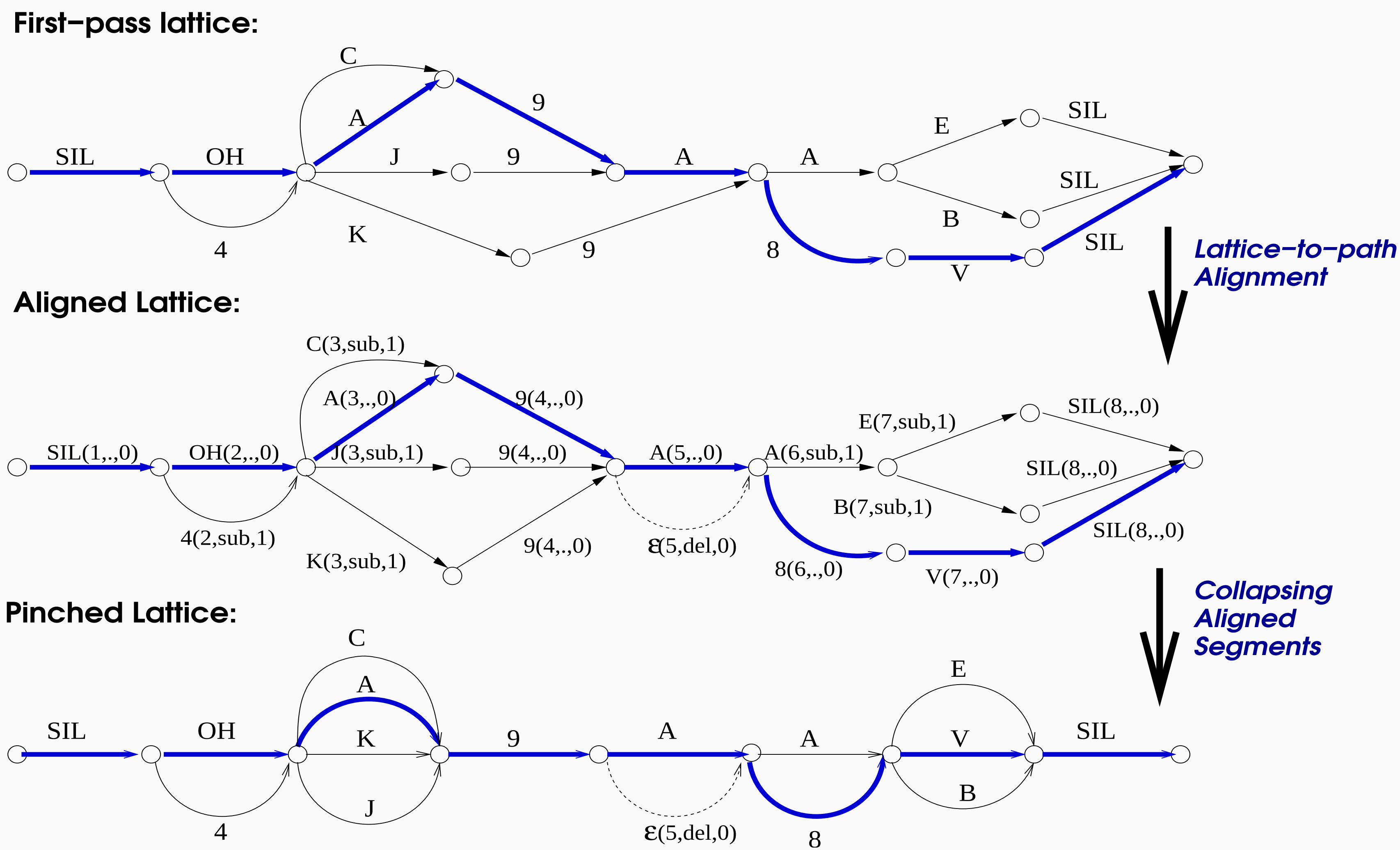
Outline

- Statistical Speech Recognition
- Identification of Confusions
- SVMs for Continuous Speech Recognition
- Validation on a Small Vocabulary task
- Feasibility for Large Vocabulary tasks
- Conclusions and Future Work

< > - +

Lattice Cutting [V. Goel *et al*, ’04]

Identifying ASR sub-problems in an unsupervised manner:



< > - +

Key Aspects of Lattice Cutting

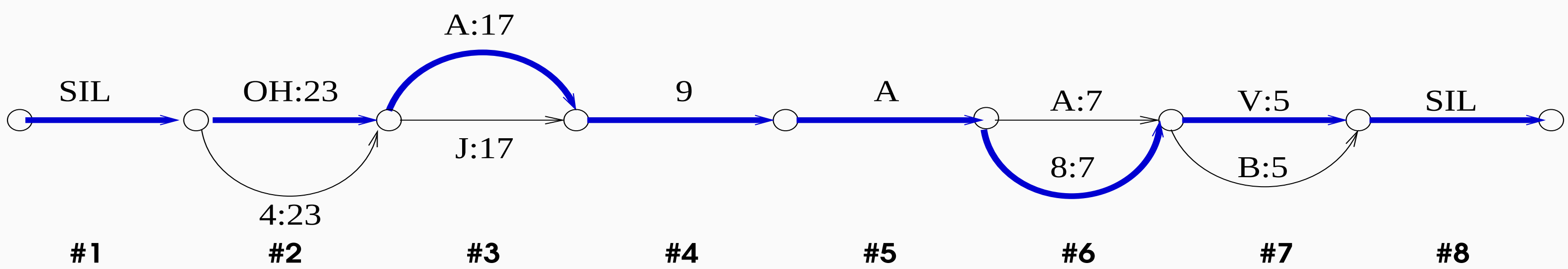
- Lattice Error Rate preserved throughout the process.
- Posteriors estimates on the collapsed segments can be obtained.
- Regions of high and low confidence.

In summary:

- Reduces ASR to a sequence of independent, smaller decision problems.
- Isolates and characterizes smaller decision problems as regions of high and low confidence, consistently and reliably.
- Consistency: identifies regions of similar confusion in both train and test data [Doumpiotis *et. al*, 03].
- Reliability: low posterior probability estimate on the MAP path usually implies a recognition error.

Pruning to obtain binary segment sets

Pinched and pruned lattices:



- Starting form the path with lowest posterior, paths are successively pruned to obtain binary confusions.
- eplsion paths are discarded

Confusion-pair specific decoder for the ith segment ($\mathcal{W}_i = \{w_{-1}, w_{+1}\}$),

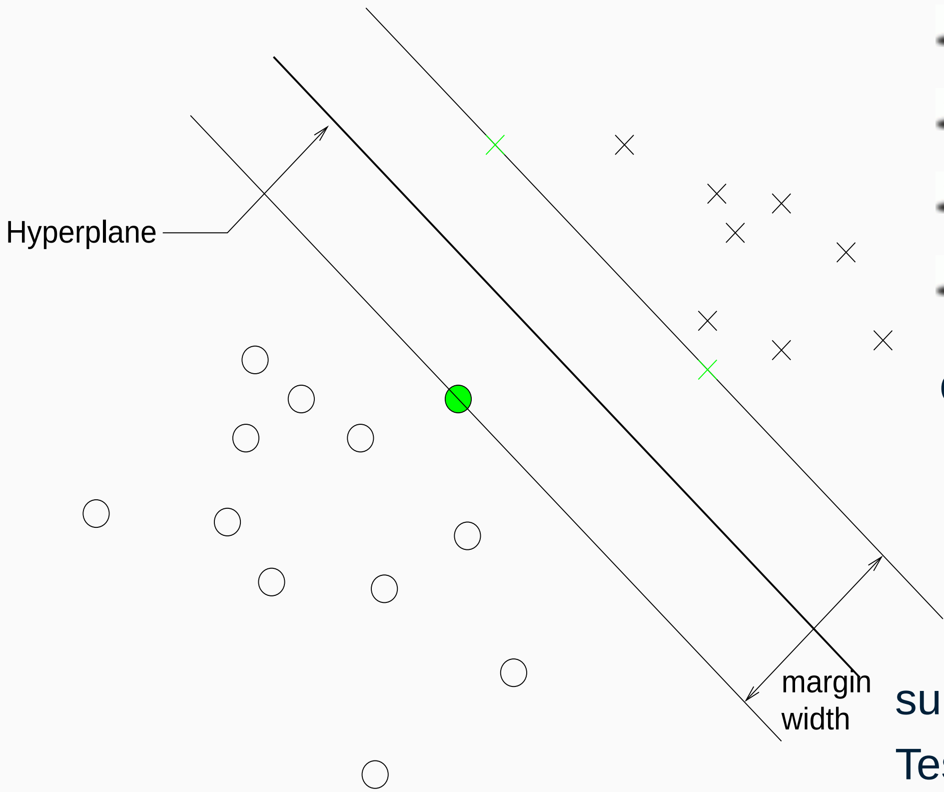
$$\hat{W}_i = \operatorname*{argmax}_{w_j \in \{w_{-1}, w_{+1}\}} p(w_j | \mathbf{O}; \theta)$$

Note that acoustics need *not* be segmented.

Outline

- Statistical Speech Recognition
- Identification of Confusions
- SVMs for Continuous Speech Recognition
- Validation on a Small Vocabulary task
- Feasibility for Large Vocabulary tasks
- Conclusions and Future Work

SVMs



- Inherently binary classifier
- Maximum margin hyperplane
- Linearly non-separable data
- Kernels

Cost function:

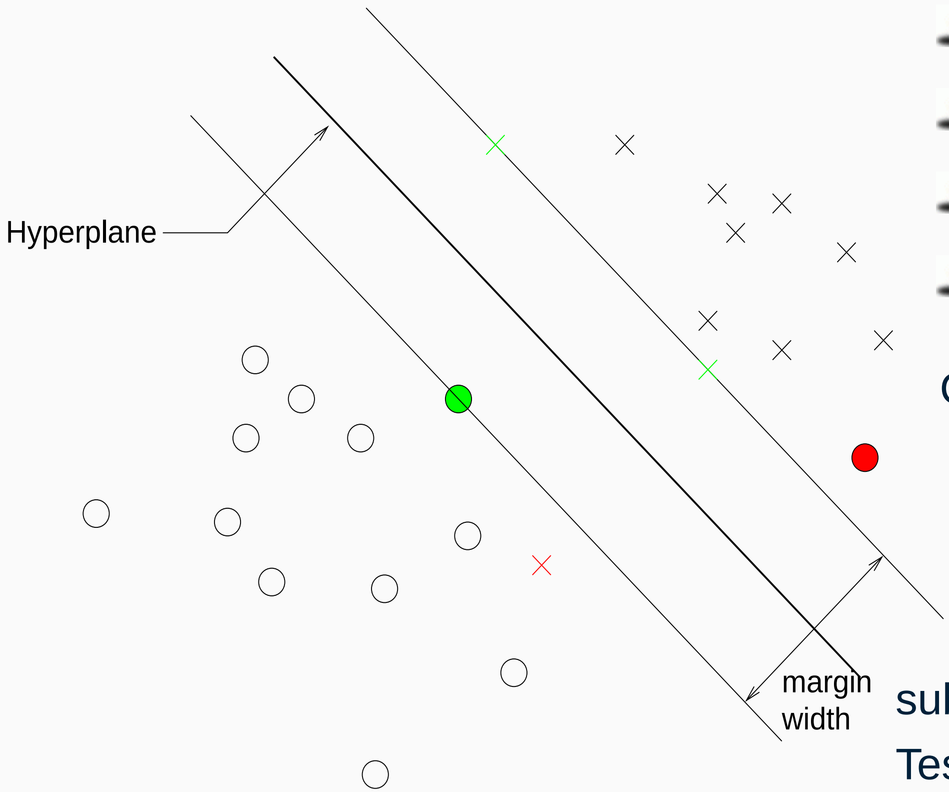
$$\frac{1}{2}\sum_{i,j}\alpha_i y_i \mathbf{K}(\mathbf{x}_i,\mathbf{x}_j) y_j \alpha_j - \sum_i \alpha_i$$

subject to $\sum_i y_i\alpha_i = 0.$

Testing: $y = \mathrm{sgn}(\sum_i y_i \alpha_i \mathbf{K}(\mathbf{x},\mathbf{x}_i)) + \mathbf{b}$

< > - +

SVMs



- Inherently binary classifier
- Maximum margin hyperplane
- Linearly non-separable data
- Kernels

Cost function:

$$\frac{1}{2}\sum_{i,j}\alpha_i y_i \mathbf{K}(\mathbf{x}_i,\mathbf{x}_j) y_j \alpha_j - \sum_i \alpha_i$$

subject to $\sum_i y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C.$

Testing: $y = \mathrm{sgn}(\sum_i y_i \alpha_i \mathbf{K}(\mathbf{x},\mathbf{x}_i)) + \mathbf{b}$

C = SVM trade-off parameter

SVMs for Continuous Speech Recognition

Lattice cutting and pruning circumvents most problems.

- Sequence Classification task.
- Multi-class task.
- Variable length observations.

Need to map variable length utterances into fixed dimension vectors.

Likelihood-ratio Score-Space [Smith *et. al* '01, Jaakkola *et. al* '99]:

$$\begin{aligned}
\varphi_\theta(\mathbf{O}) &= \begin{bmatrix} 1 \\ \nabla_\theta \end{bmatrix} \ln\left(\frac{p(\mathbf{O}|\theta_{-1})}{p(\mathbf{O}|\theta_{+1})}\right) \\
&= \begin{bmatrix} \ln \frac{p(\mathbf{O}|\theta_{-1})}{p(\mathbf{O}|\theta_{+1})} \\ \nabla_{\theta_{-1}} \ln p(\mathbf{O}|\theta_{-1}) \\ -\nabla_{\theta_{+1}} \ln p(\mathbf{O}|\theta_{+1}) \end{bmatrix}
\end{aligned}$$

where $\mathbf{O}$ is a T-length observation sequence, θ_i are the parameters of the ith HMM and $\theta = [\theta_{-1}^\top \theta_{+1}^\top]^\top$.

Mean Score-Spaces

- We are deriving these fixed dimension vectors from HMMs themselves.
- Each component of a score is the sensitivity of the log-likelihood-ratio of the observed sequence to a parameter of the generative model.

Mean Score-Space:

The gradient w.r.to $\mu_{i,s,j}$, the mean of the Gaussian observation density of the jth component of the sth state of the ith HMM is given by,

$$\nabla_{\mu_{i,s,j}} \ln p(\mathbf{O}|\theta_i) = \sum_{t=1}^{T} \gamma_{i,s,j}(t) \left[(o_t - \mu_{i,s,j})^\top \Sigma_{i,s,j}^{-1} \right]^\top ,$$

where $\gamma_{i,s,j}$ is the posterior occupation probability of component (i, s, j) and $\Sigma_{i,s,j}$ is the variance.

Note that the observation sequence $\mathbf{O}$ is *not* segmented.

< > - +

Score-Space Normalization

Mean/Variance Normalization [Smith *et. al*]:

$$\bar{\varphi}_\theta(\mathbf{O}) = \hat{\Sigma}_{sc}^{-1/2}[\varphi_\theta(\mathbf{O}) - \hat{\mu}_{sc}],$$

where $\hat{\Sigma}_{sc} = \int \varphi_\theta(\mathbf{O})'\varphi_\theta(\mathbf{O})P(\mathbf{O}|\theta)d\mathbf{O}$ and $\hat{\mu}_{sc} = \int \varphi_\theta(\mathbf{O})P(\mathbf{O}|\theta)d\mathbf{O}$.

- $\hat{\mu}_{sc}$ and $\hat{\Sigma}_{sc}$ are *not* HMM parameters.
- $\hat{\mu}_{sc}$ and $\hat{\Sigma}_{sc}$ are approximated over the training data.

$$\hat{\Sigma}_{sc} = \frac{1}{N-1}\sum(\varphi_\theta(\mathbf{O}) - \hat{\mu}_{sc})^\top(\varphi_\theta(\mathbf{O}) - \hat{\mu}_{sc})$$

$$\hat{\mu}_{sc} = \frac{1}{N}\sum \varphi_\theta(\mathbf{O})$$

and N is the number of training samples for the SVM.

- Diagonal approximation for Σ_{sc}.

Sequence length normalization (for the utterance length T):

$$\bar{\varphi}_\theta^T(\mathbf{O}) = \frac{1}{T}\bar{\varphi}_\theta(\mathbf{O})$$

Previous Work: SVMs for Speech Tasks

A sample of the previous work:

- Ganapathiraju *et al.*.
  - Forced every sequence to have same length.
- Smith *et al.*
  - Used Score-Spaces for handling Variable length observations.
  - Only isolated binary classification.
- Chakrabartty *et al.* developed Forward Decoding Kernel Machines and the $gini$SVM.
  - Mainly motivated for producing sparse SVM solutions.
  - We used $gini$SVMs in our experiments.
- Fine *et al.* used Score-Spaces for Speaker Identification.

Outline

Small Vocabulary Experiments

OGI AlphaDigits Corpus:

- Vocabulary of 37 words (26 letters and 11 numbers)
- Training set $\approx$ 50K utterances, each utterance having 6 words.
- Test set has 3112 utterances, also having 6 words each.
- Word loop grammar (any word can follow any word).

Baseline HMM System:

- Each word is modeled by a left-to-right 20 state HMM, 12 mixtures per state.
- 39 dimensional feature vectors, at a 10msec period.
- WER of MMI-HMM systems is around 9%.

SVM Training

- Cut Train and Test set lattices.
- 50 most frequently observed confusion pairs $e.g.$, [B,V], [TWO,U].
  - $\approx$ 120,000 instances in the training set.
  - $\approx$ 8,000 instances in the test set.
- Lattice Word Error Rate increased from 1.7% to 4.1%.
- Log-likelihood ratio scores were generated.
- Global SVM trade-off parameter (C) set at 1.0 for all confusion pairs.
- Used $tanh$ kernels.

Results

WERs for HMM and SVM systems:

Training Criterion	HMM	SVM	System Combination
ML	10.7	8.6	8.2
MMI	9.1	8.1	7.7

Classifier Combination:

- Error patterns are uncorrelated between HMM and SVM based systems.
- For HMM and SVM systems at 8% WER the difference was 4%.
- Ideal for system combination.

$$p_+(w_i) = \frac{p_h(w_i) + p_s(w_i)}{2}$$

$p_h(w_i)$ is the HMM posterior estimate obtained from the pinched lattice
$p_s(w_i)$ is the SVM posterior estimate

Outline

- Statistical Speech Recognition
- Identification of Confusions
- Posterior Distributions from $Gini$SVMs
- Validation on a Small Vocabulary task
- Feasibility on a Large Vocabulary task
  - Identify small number of sub-problems and show performance improvements in these sub-problems.
  - Requires huge test sets to validate, $i.e.$, to obtain statistically significant improvements.
  - Improvements will be modest by design!
- Conclusions and Future Work

System Description

MALACH spontaneous Czech conversational domain:

Train:

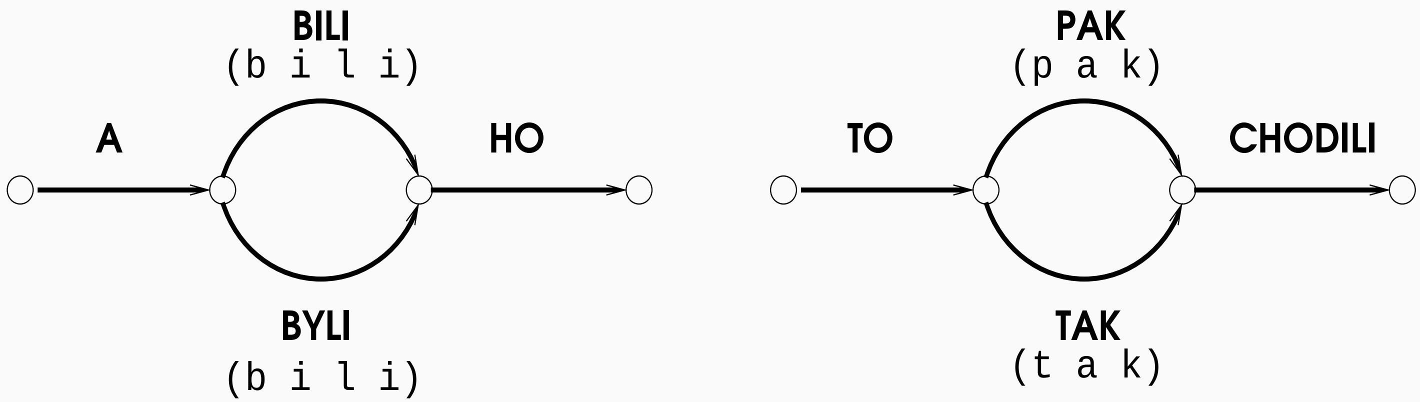
- 65 hours of acoustic training data
- 39 dimensional MFCCs, delta and acceleration coefficients
- HMMs trained HTK style
- Speaker independent continuous mixture density, tied state, cross-word, gender-independent, triphone HMMs.
- 80K Vocabulary; Bigram LM interpolated with out-of-domain data.
- Lattices generated using the AT&T decoder.
- Lattice-based MMIE was performed.

Test:

- Test set is 8400 utterances ($\approx$ 25 hours) from 10 heldout speakers
- Unsupervised MLLR transforms were estimated on a 1000 utterance subset.
- WER of MAP is 45.6%WER. Lattice Error Rate (LER) is 13.5%.

Challenges faced

- Sparsity - LER with frequently occuring confusion pairs is practically the WER.
- Language Models. Homonym confusion pairs: Words with different semantics but with similar phonetic sequences.



Possible to train specialized language models.

- Identifying segment sets where MAP is erroneous.
- Identifying segment sets containing truth.

Posteriors of the MAP path can indicate if erroneous.
Study lattice cutting as we prune paths based on their posteriors.

Studying Segment Set Pruning

Towards studying the ability of lattice pinching in
(a) identifying regions where the MAP hypothesis is in error and,
(b) identifying the correct alternative.

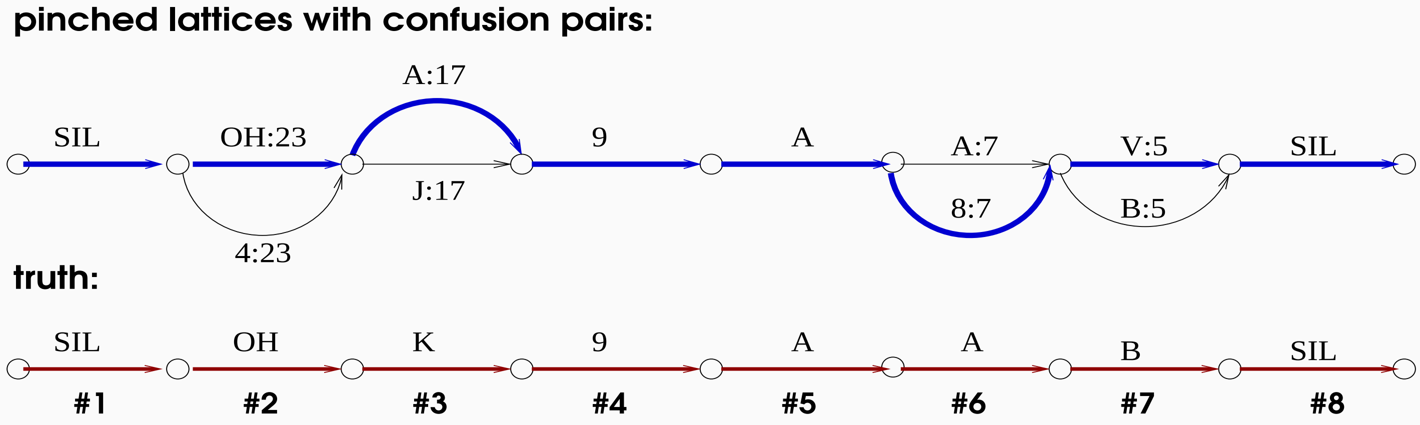
Effect of pruning links based on their posteriors:

Pruning Threshold	LER	Avg. # Hyps./ Segment Set
0.00	27.3	11.65
0.05	35.3	2.82
0.10	37.9	2.35
0.20	41.1	2.06
0.30	43.2	2.00
0.40	44.7	2.00
0.50	45.6	-

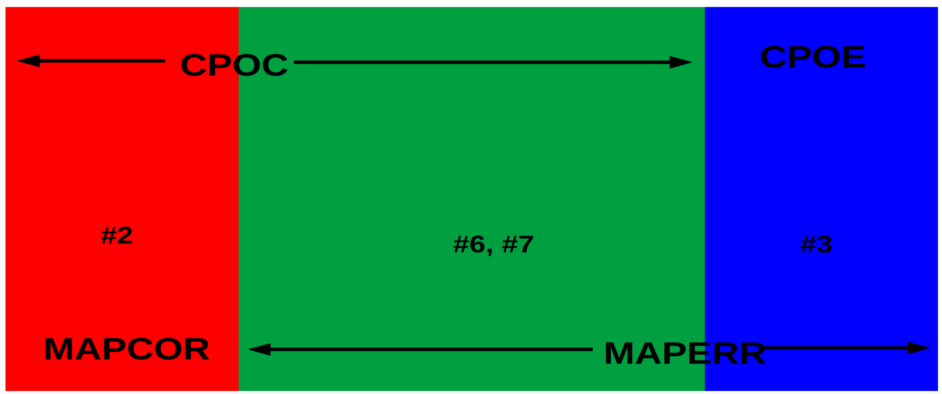
Pruning paths based on their posteriors removes more incorrect paths than correct ones.

Focus only on binary confusion problems that occur at least 100 times in the test set.

Characterization of Segment Sets

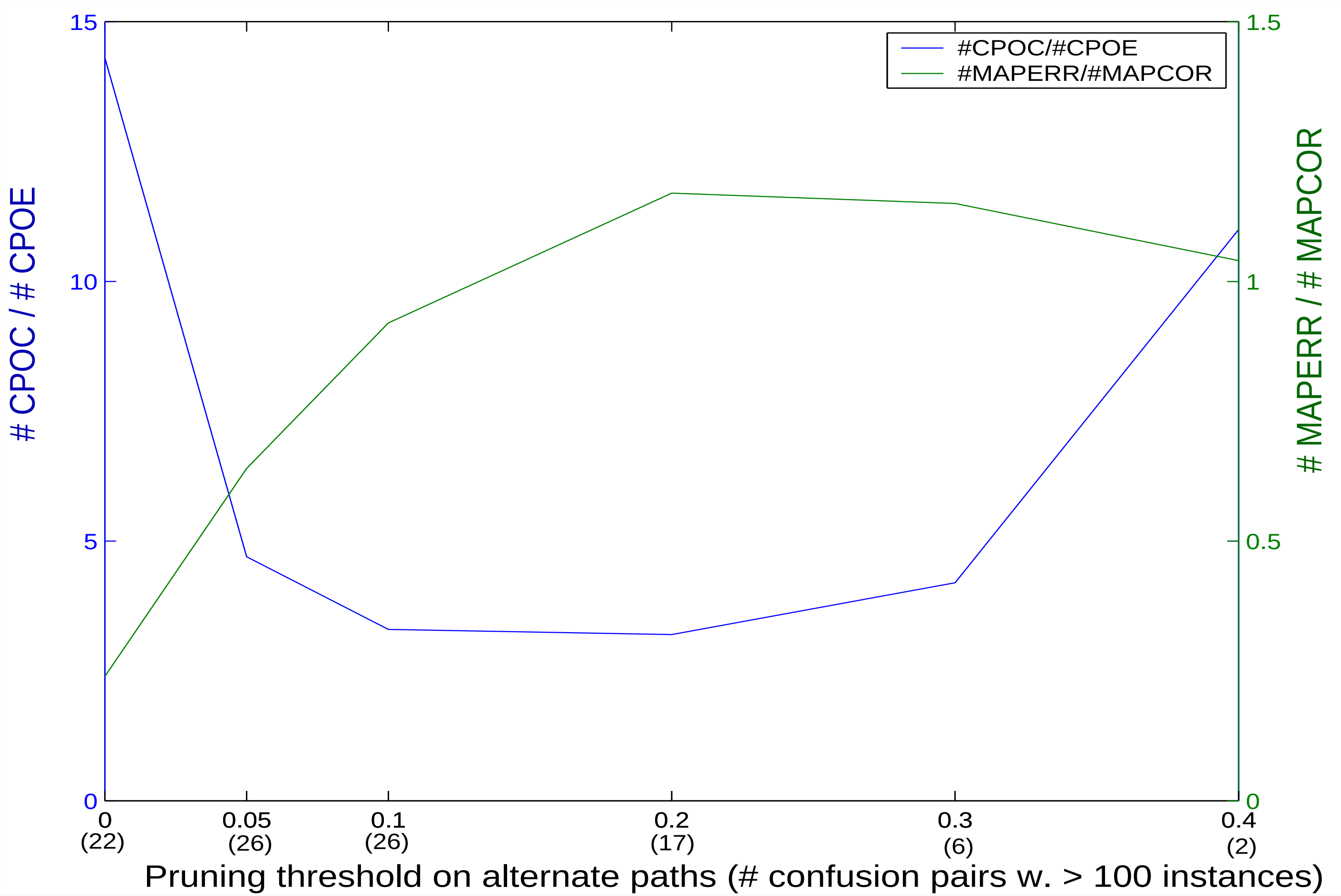


- Confusion Pair Oracle Correct (CPOC) vs. Confusion Pair Oracle Error (CPOE)
- MAP Correct (MAPCOR) vs. MAP Error (MAPERR)



Want to have as large a green region as possible.

Choosing the Code Breaking Set



Choose threshold=0.10 to balance #CPOC/#CPOE, #MAPERR/#MAPCOR and sparsity.

Choosing the Code Breaking Set (contd.)

RECAP:

1. Pinch test set lattices.
2. Prune from collapsed segment sets, any path with posterior < 0.10.
3. Only keep confusion pairs, *i.e.*, binary problems alone.
4. Only confusion pairs that occur at least a 100 times.
5. Homonym confusion pairs are pruned back to the MAP.

Our final code-breaking set: 21 confusion pairs with 2991 total segment sets.

Of these around 1200 are MAPERR $\Rightarrow$ *utmost* 0.8% WER improvement.

Identified confusion pairs involved function words,

e.g., [PAK,TAK], [TAM,TO] and [SE,SEM].

Training Specialized HMMs and SVMs

Need to train *word-level* HMM models to obtain scores.
Let [PAK, TAK] be uniquely indexed by 7.

- Initialize *word* level models, PAK and TAK, by concatenating monophone models.
- Re-estimate the word models using EM.
- Clone these models as PAK:7 and TAK:7.
- Create a [PAK, TAK]-specific training set that contains all instances of PAK and TAK from the acoustic training set.
- Train PAK:7 and TAK:7 using MMI.
- Repeat for all confusion pairs.

SVMs:

- Obtain Scores from the MMI word level HMMs.
- Train $Gini$SVMs for each confusion pair.

< > - +

Testing - HMM+SVM system combination

Testing: For each instance of a confusion pair,

1. Obtain log-likelihood ratio Scores from the MMI word HMMs.
2. Obtain posterior probability estimates.
3. Perform system combination with HMM posteriors from the pinched lattice.

$$p_\lambda(w_i) = \lambda p_h(w_i) + (1-\lambda)p_s(w_i), \quad 0 \leq \lambda \leq 1$$

$p_h(w_i)$ is the HMM posterior estimate obtained from the pinched lattice,
$p_s(w_i)$ is the SVM posterior estimate.

RESULTS:

- Error Counts decrease in 18 of 21 confusion pairs for the MAP+SVM system.
- Statistically significant improvements (0.1% WER) obtained for $\lambda = 0.4, 0.5, 0.6,$ and 0.7.

Conclusions & Contributions

- New framework to evaluate novel techniques in ASR.
  - Identify regions of weakness of a state-of-the-art HMM decoder.
  - Train specialized decoders for each kind of confusion.
  - Resolve confusions with these decoders.
- Developed the framework to gainfully use SVMs in continuous speech recognition.
- Showed Posterior distribution estimated by $Gini$SVMs can be used favorably in system combination.
- Validated the use of SVMs on a small vocabulary task.
- Studied the effects of pruning on lattice cutting on an LVCSR task.
- Demonstrated the feasibility of the framework on an LVCSR task; showed small but statistically significant gains.

Future Work

Future Work:

- Further gains on MALACH.
  - Can cluster confusion pairs if the source of confusion is similar. *e.g.*, (TA, TO) and (NA, NO).
  - Provides more instances of confusion pairs.
  - Will use phone level HMMs to obtain scores.
- Multi-class classifiers.
- Language Model code-breaking.
  - Bias will be an issue. LMs tend to learn the training data more.
- Study of confusions.
  - What kinds of confusions are tougher to resolve?

Publications:

- V. Venkataramani, S. Chakrabartty and W. Byrne, SVMs for Segmental Minimum Bayes Risk Decoding of Continuous Speech, ASRU '03.
- V. Venkataramani, W. Byrne, Lattice Segmentation and SVMs for LVCSR, ICASSP '05.
- V. Venkataramani, S. Chakrabartty and W. Byrne, $Gini$SVMs for Segmental Minimum Bayes Risk Decoding of Continuous Speech, Submitted CSL.

< > - +

Acknowledgements (Thesis related)

- Prof. Bill Byrne
- Prof. Gert Cauwenberghs
  - Course project (520.774 Kernel Machine Learning).
- Prof. Shantanu Chakrabartty
  - $Gini$SVM toolkit
  - Countless hints and suggestions
- Dr. Vlasios Doumpiotis
  - MMI models
- Dr. Kumar Shankar
  - Lattice Cutting support
- Dr. Teressa Kamm
  - ML models for alphadigits

< > - +

References

1. F. Jelinek, Speech Recognition as Code Breaking, CLSP Research Notes 5, 1996.
2. J. Fritsch *et al*, Applying Divide and Conquer to Large Scale Pattern Recognition Tasks, Neural Networks: Tricks of the Trade,pp. 315-342, 1996.
3. L. Mangu *et al*, Finding consensus in speech recognition: word error minimization and other applications of confusion networks, CSL, vol. 14(4), pp. 373-400, 2000.
4. G. Evermann *et al*, "Posterior probability decoding, confidence estimation and system combination," Proc. NIST Speech Transcription Workshop, 2000.
5. L. Bahl *et al*, Estimating hidden Markov models parameters so as to maxmise speech recognition accuracy, IEEE Trans, 1 (1):77-83, 1993.
6. Y. Freund *et al*. Decision theoretic generalization of on-line leargning and an application to boositng, 2nd Euro Conf. on Conputational Learning Theory, 1995.
7. V. Goel and S. Kumar and W. Byrne. Segmental Minimum Bayes-Risk Decoding for Automatic Speech Recognition. *IEEE Trans.*, Vol 12(3), 234-250. 2004.
8. T. Jaakkola and D. Haussler. Exploiting Generative Models in Discriminative Classifiers.*Advances in Neural Information Processing Systems 11*, 1999.

References

9 N. Smith, M. Gales, and M. Niranjan. Data-dependent kernels in SVM classification of speech patterns. Technical Report CUED/F-INFENG/TR387, April 2001.

10 S. Chakrabartty and G. Cauwenberghs. Forward Decoding Kernel Machines.*Proc. SVM'2002, Lecture Notes in Computer Science*, p. 278-292.

11 Vlasios Doumpiotis, Stavros Tsakalidis, and William Byrne. Discriminative Training for Segmental Minimum Bayes Risk Decoding. *Proc. ICASSP '03*.

12 A. Ganapathiraju and J. Hamaker and J. Picone. Advances in Hybrid SVM/HMM Speech Recognition. *GSPx / International Signal Processing Conference*, 2003.

13 S. Fine *et al*, A hybrid GMM/SVM approach to speaker identification, Proc. ICASSP, 2001.

< > - +