Minimum Risk Estimation and Decoding in Large Vocabulary Continuous Speech Recognition

Bill Byrne

Machine Intelligence Laboratory
Cambridge University Engineering Department

Center for Language and Speech Processing
The Johns Hopkins University

Recent work by CLSP graduates:
Vlasios Doumpiotis, Vaibhava Goel,
Shankar Kumar, Veera Venkataramani
Overview

- Loss-Sensitive Modeling
  - loss functions, search algorithms, task-specific decoding
- Efficient lattice-based computation
  - Lattice-to-string alignment, induced loss functions
- Minimum Risk Discriminative Training
  - HMM parameter estimation, error properties
- Acoustic Codebreaking
  - Support Vector Machines for continuous speech recognition
Goal: A statistical modeling framework to study the integration of ASR, SMT and other language processing technologies

Loss Functions

- Describe performance under different applications

Reference:

Ref: HUGH TALKED ABOUT VOLCANOES ...  
Hyp: YOU TALKED ABOVE VOLCANOES ...

Loss functions capture application-specific errors

- Examples: Word Error Rate, Sentence Error Rate, Keyword Error Rate, F-Measure, BLEU score, ...
Statistical Speech Recognition

Acoustic sequence: $A = A_1, \ldots, A_m$

Word sequences: $W = W_1, \ldots, W_n \quad W_i \in \mathcal{V}$

Maximum Likelihood Parameter Estimation: $\hat{\theta} = \arg\max_{\theta} P(A|\bar{W}; \theta)$

Maximum Likelihood Decoder:

$$\hat{W} = \arg\max_W P(W|A) = \arg\max_W P(A|W)P(W)$$

![Diagram of Speech Recognition System](image.png)
Minimum Bayes-Risk Speech Recognition

Any decoder $\delta$ has an expected performance $E_{P(W,A)}L(W,\delta(A))$

An optimal decoding / rescoring strategy to minimize the Expected Loss:

- Make a list of all sentence hypotheses : $\mathcal{W}$
- Find the loss between all strings $W$ and $W'$ from $\mathcal{W}$
- Evaluate the risk associated with taking $W$ as the decision
  $$E(W) = \sum_{W' \in \mathcal{W}} L(W,W')P(W'|A)$$
- Pick the hypothesis with the minimum risk
  $$\hat{W} = \arg\min_{W \in \mathcal{W}} \sum_{W' \in \mathcal{W}} L(W,W')P(W'|A)$$

Given models and a loss function, MBR decoding is a search problem
- Efficient A* search is available
- Further efficiencies come from search space refinements
Efficient Lattice MBR Computation

For every lattice path \( W \in \mathcal{W} \) the following has to be computed

\[
E(W) = \sum_{W' \in \mathcal{W}} L(W, W') P(W' | A)
\]

- Computing with likelihoods over a lattice is easy
  - likelihoods are maintained over lattice arcs
  - easy to compute posterior distributions over words and sentences
    useful for discriminative training, system combination, ...
- What about computing with loss, specifically Word Error Rate?
Word Error Rate Requires String-to-String Alignment

How to align \( W_1, \ldots, W_N \) to \( W'_1, \ldots, W'_N \):

- Introduce an alignment variable \( a \)
- Find the optimum alignment based on a symbol-to-symbol cost

\[
L(W, W') = \min_a \sum_i L(W_i, W'_{a_i})
\]

Easily computed over permissible alignments by dynamic programming

How to efficiently align every lattice path \( W' \in W' \) to a reference path \( W \)?
Efficient (nearly) Exact Lattice-to-String Alignment

\[ L(\tilde{W}, W) = \min_a \sum_i L(\tilde{W}_i, W_{ai}) \]
Lattice Segmentation and Induced Loss Functions

Aligned Lattice

Sausage

No paths are lost yet

Pinched Lattice

Paths are pruned away

\[ L_I(W, W') = L(W_1, W'_1) + L(W_2, W'_2) + L(W_3, W'_3) + L(W_4, W'_4) \]

Alignments are approximate: \[ L(W, W') \leq L_I(W, W') \]
Segmental MBR Decoding

If MBR decoding is performed with the Induced Loss Function over the pinched lattice, decoding simplifies as follows:

\[
\arg\min_{W \in W} \sum_{W' \in W} L(W, W') P(W' | A) \approx \arg\min_{W \in \tilde{W}} \sum_{W' \in \tilde{W}} L_I(W, W') P(W' | A)
\]

\[
\approx \arg\min_{W \in \tilde{W}} \sum_{i} \sum_{w \in \tilde{W}_i} L(W_i, w) P_i(w | A)
\]

Sequence of independent MBR searches over sublattices using costs and probabilities derived from the original lattice

Can be used for LVCSR decoding, system combination, ...

- Improved search speed / pruning relative to A* over original lattices
Early Application of MBR - Transcription and Keyword Spotting

Specialized decoders for task-specific metrics

- **Transcription**: Word Error Rate
- **Keyword Spotting**: Modified Word Error Rate
  Ignores all errors for words not in a keyword list

**SWITCHBOARD Conversational Speech Recognition Task**

<table>
<thead>
<tr>
<th>MBR Recognition Strategy</th>
<th>Error Measure(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SER</strong></td>
<td><strong>WER</strong></td>
</tr>
<tr>
<td>ML (baseline)</td>
<td>65.9</td>
</tr>
<tr>
<td>N-best</td>
<td>66.8</td>
</tr>
<tr>
<td>A-star</td>
<td>66.8</td>
</tr>
<tr>
<td><strong>KER</strong></td>
<td><strong>N-best</strong></td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>A-star</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Similar gains for more recent LVCSR tasks with lower WER and for SMT under BLEU score

- Relatively small, but consistent, gains

How do we get bigger gains? Can we extend Risk-Based approaches to model training?
Overview

- Loss-Sensitive Modeling
  - loss functions, search algorithms, task-specific decoding
- Efficient lattice-based computation
  - lattice-to-string alignment, induced loss functions
- Minimum Risk Discriminative Training
  - error identification, HMM parameter estimation, error properties
- Acoustic Codebreaking
  - Support Vector Machines for continuous speech recognition
HMM Parameter Estimation

\[
b(A_\tau|s; \theta) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_s|}} e^{-\frac{1}{2}(A_\tau - \mu_s)'\Sigma_s^{-1}(A_\tau - \mu_s)}
\]

Assume we have transcribed training data : \((\bar{A}, \bar{W})\)

- **Maximum Likelihood Estimation Criteria** : \(\text{argmax}_\theta P(\bar{A} | \bar{W}; \theta)\)
- **Baum Welch Algorithm** : 

\[
\bar{\mu}_s = \frac{\sum_\tau P(S_\tau = s|\bar{A}, \bar{W}; \mu_s) \bar{A}_\tau}{\sum_\tau P(S_\tau = s|\bar{A}, \bar{W}; \mu_s)}
\]
Discriminative Training for MBR

Goal: drive down the likelihood of hypotheses that are `far’ from the truth

One possibility is to minimize the Expected Risk over the training data

$$\theta^* = \arg\min_{\theta} \sum_{W' \in \mathcal{W}} l(\bar{W}, W') P(W' | A; \theta)$$

**Motivation:** Hypotheses contribute to the Risk if they have errors and are relatively likely. The training objective is to move the probability mass away from these and towards hypotheses with fewer errors.

Variations of this have been studied for ASR

- Kaiser, Horvat, & Kacic, 2000; Povey & Woodland, 2002

 _good idea_ but difficult to implement via EM-style, fixed update, training algorithms

What happens if we use the loss function induced by lattice pinching?
Generalization of Established Procedures

Conditional Maximum Likelihood Criterion - MMI/CML Estimation

$$\max_{\theta} P(\tilde{W} | \tilde{A} ; \theta)$$

Instance of Minimum Risk Estimation under the 0/1 Loss

$$\min_{\theta} \sum_{W} L_{0/1}(W, \tilde{W}) P(W | \tilde{A}) \leftrightarrow \max_{\theta} P(\tilde{W} | \tilde{A} ; \theta)$$

$$L_{0/1}(W, W') = \begin{cases} 0 & W = W' \\ 1 & \text{otherwise} \end{cases}$$
The Extended Baum Welch Algorithm and Risk-Based Parameter Estimation

Kaiser, Horvat, & Kacic observed that the objective function

\[ \sum_{W' \in \mathcal{W}} L(\bar{W}, W') P(W' | \mathcal{A}; \theta) \]

can be optimized via the Extended Baum Welch algorithm

Update for a Gaussian mean of an HMM observation distribution:

\[
\bar{\mu}_s = \frac{\sum_{W' \in \mathcal{W}} K(W'; \mathcal{W}, \theta) \sum_{\tau} \gamma_s(\tau; W') A(\tau) + D_s \mu_s}{\sum_{W' \in \mathcal{W}} K(W'; \mathcal{W}, \theta) \sum_{\tau} \gamma_s(\tau; W') + D_s}
\]

where

\[
K(W'; \mathcal{W}, \theta) = \left[ \sum_{W'' \in \mathcal{W}} P(W'' | \mathcal{A}; \theta) L(\bar{W}, W'') - L(\bar{W}, W') \right] P(W' | \mathcal{A}; \theta)
\]

- simplifies to MMI under the 0/1 loss function
Direct Realization of the EBW Update Rule

\[ \bar{\mu}_s = \frac{\sum_{W' \in \mathcal{W}} K(W'; \mathcal{W}, \theta) \sum_{\tau} \gamma_s(\tau; W') A(\tau) + D_s \mu_s}{\sum_{W' \in \mathcal{W}} K(W'; \mathcal{W}, \theta) \sum_{\tau} \gamma_s(\tau; W') + D_s} \]

\[ K(W'; \mathcal{W}, \theta) = \left[ \sum_{W'' \in \mathcal{W}} P(W''|A; \theta) L(\bar{W}, W'') - L(\bar{W}, W') \right] P(W'|A; \theta) \]

Straightforward to implement when \( \mathcal{W} \) is an N-Best list:

- For each \( W' \in \mathcal{W} \):
  - \( L(\bar{W}, W') \) can be found by string-to-string alignment
  - The statistics \( \gamma_s(\tau, W') \) have to be found, e.g. by Forward-Backward

However N-Best lists are not ideal for LVCSR discriminative training:

- too small an evidence space can lead to underestimation of the risk
- training set lattices are needed
Risk-Based Pruning of the Evidence Space

N-Best lists in LVCSR are large because LVCSR lattices are large

- Can use lattice pinching and pruning to refine the lattices while retaining focus on problem hypotheses

Minimum Bayes Risk estimation Under the Induced Loss Function

\[ \theta^* = \arg\min_\theta \sum_{W' \in \tilde{W}} L_I(\bar{W}, W') P(A|W'; \theta) \]

- Extended Baum-Welch parameter estimation can be performed over N-Best lists extracted from the refined lattices
  - Time segmentation at word or phone boundaries is not needed
  - Statistics under \( P(A|W'; \theta) \) can be found by Forward-Backward
  - \( L_I(\bar{W}, W') \) is provided by lattice-to-string alignment
Lattice Pruning for MBR Estimation

1. Generate lattices over the acoustic training set
2. Perform lattice-to-string alignment to the reference transcriptions
   - retain alignment loss on arcs
3. Identify frequently occurring confusion pairs (optional)
4. Prune all other choices back to the reference path
5. Expand the pinched and pruned lattices to N-Best lists
6. Gather statistics needed by Risk-Based parameter estimation
MBR Estimation

SWITCHBOARD I
Resolve 149 binary choices identified by lattice cutting

AlphaDigits
Resolve 50 binary choices identified by lattice cutting
AlphaDigits : Within-Class Error Analysis

MBR-DT avoids CML’s tendency to fix some error types at the expense of other types

- Arguably due to a better match between training and evaluation criteria
### Lattice Segmentation and Training Set Refinement

Lattice pinching and pruning reduces the training set size
- utterances that do not contribute to the risk are discarded

<table>
<thead>
<tr>
<th></th>
<th>SWITCHBOARD (MINITRAIN)</th>
<th>MALACH-CZ</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training</strong>&lt;br&gt;(hours / utterances)</td>
<td>16.9 / 22,580</td>
<td>62.4 / 24,065</td>
</tr>
<tr>
<td><strong>Confusion Pairs</strong>&lt;br&gt;(types / tokens)</td>
<td>25,948 / 99,199</td>
<td>31,467 / 120,695</td>
</tr>
<tr>
<td><strong>Occurrence Threshold</strong></td>
<td>5</td>
<td>75</td>
</tr>
<tr>
<td><strong>Confusion Pairs</strong>&lt;br&gt;(types / tokens)</td>
<td>2,139 / 66,349</td>
<td>159 / 33,821</td>
</tr>
<tr>
<td><strong>Training</strong>&lt;br&gt;(hours / utterances)</td>
<td>15.0 / 19,687</td>
<td>13.0 / 15,741</td>
</tr>
<tr>
<td><strong>Avg. N-Best List Depth</strong></td>
<td>48.4</td>
<td>13.1</td>
</tr>
</tbody>
</table>

Lattice pinching and pruning reduces the training set size.
- Utterances that do not contribute to the risk are discarded.
Analysis

Lattice-to-string alignment originally developed for MBR decoding

○ Can be useful for parameter estimation - efficient computation of risk

○ Lattice-to-String alignment creates an induced loss function

Lattice pinching and pruning is used to restrict the lattices to the hypotheses that are both likely and errorful

○ These lattices are small enough to be expanded into N-Best lists

○ Exact Risk-based estimation can be carried out over the N-Best lists

○ Training set is reduced to the utterances relevant to the loss function

Improvements relative to MMI

○ More than simply MMI over pinched lattices -- there are gains in using Levenstein loss rather than 0/1 Loss over pinched lattices

Feasible to tune the estimation procedure to a task specific loss function
Overview

- Loss-Sensitive Modeling
  - loss functions, search algorithms, task-specific decoding
- Efficient lattice-based computation
  - lattice-to-string alignment, induced loss functions
- Minimum Risk Discriminative Training
  - error identification, HMM parameter estimation, error properties
- Acoustic Codebreaking
  - Support Vector Machines for continuous speech recognition
Acoustic Codebreaking

An unsupervised, divide-and-conquer approach to ASR:

- Perform a first decoding pass to generate lattices
- Perform lattice segmentation to locate and characterize errors

Unsupervised identification of recognition errors

- Search training data to create training sets for all these errors
- Train specialized models for each type of error

Can use different types of models for different types of errors:

- HMMs, SVMs, language models, pronunciation models, ...
- Decode pinched lattice using the specialized models

Not an attempt to replace HMMs - the goals are (a) to find weaknesses in the HMM hypotheses and (b) fix them
Case Study: SVMs for Continuous Speech Recognition

Support Vector Machines - binary pattern classifiers

- SVMs classify patterns into one of two classes
- Feature vectors are fixed dimensional, real valued, measurements
- SVMs are naturally suited for use as static pattern classifiers

SVMs work very well on problems to which they can be applied

Speech recognition isn’t a fixed dimensional, binary problem

- Only the simplest ASR problems are binary, e.g. words & phrases
- Speech is inherently variable:
  - speaking rate, pauses, variations in the grammar ...
Overview

Prior Work in Three Areas:

- Segmental Minimum Bayes Risk (SMBR) rescoring
  lattice-to-string alignment and loss-based lattice cutting

- SVMs
  giniSVMs - SVMs under the Gini entropy (S. Chakrabartty 2003)

- Score spaces (Smith et al. 2001)
  features for SVM classifiers derived from HMM likelihoods

SVMs + SMBR

- Use lattice cutting to transform lattice rescoring into a sequence of binary classification problems that can be solved with SVMs
SVMs in MBR: Acoustic Codebreaking

Training (one scenario)

- Use the baseline decoder to generate lattices for the training set, and align the lattices against the reference transcription
- Prune to obtain binary confusion pairs
- Train an SVM for each of the dominant confusion pairs, based on the score space of the best available HMMs

Decoding

- Use the baseline decoder to generate lattices for the test set, and align the lattices against the MAP hypothesis
- Prune aligned lattices and identify binary confusion sets
- Rescore each confusion set with the appropriate SVM
SVMs and Pinched Lattice Rescoring

OGI Alphadigits - 37 word vocabulary

- 39 dim. MFCC features, ~20 state, 12 mixture Gaussians, whole word models
- SVM - Rescore lattices produced using MMI models

Mean-Only Score Space : ~2K elements

Pinching & Pruning increases Lattice WER from ~1% to ~3%

- System Combination : voting based on posteriors over hypotheses

<table>
<thead>
<tr>
<th>HMM Training Procedure</th>
<th>Viterbi Decoding</th>
<th>MBR-SVM Decoding</th>
<th>System Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>10.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MMI</td>
<td>9.1</td>
<td>8.1</td>
<td>7.8</td>
</tr>
<tr>
<td>Pinched Lattice MMI</td>
<td>8.0</td>
<td>8.0</td>
<td>7.5</td>
</tr>
</tbody>
</table>

Can we extend this approach to Large Vocabulary Continuous Speech Recognition ?

How do we avoid being swamped by recognition errors ?

- One strategy is to be selective about which errors to try to fix
Finding and Fixing Small Problems in Large Vocabulary Speech Recognition

Goal: Identify regions of uncertainty in the first-pass ASR lattice that can be resolved by a binary classifier

Consider this instance of $C_5 = \{V : 5, B : 5\}$ - the MAP hyp is $V$

Three possibilities

- the MAP hyp might be wrong $\rightarrow$ consider replacing $V$ by $B$
- neither $V$ nor $B$ is right $\rightarrow$ choice won’t affect the baseline
- the MAP hyp is correct $\rightarrow$ don’t fix this instance of $C_5$

How do we decide whether to try to fix $C_5$?
A Pruning Scheme to Find Fixable Confusion Pairs

1. Prune links by their posteriors relative to a set threshold
2. Some segment sets shrink to pairs - these are *Natural Confusion Pairs*
3. Prune all other sets back to the MAP hypothesis

<table>
<thead>
<tr>
<th>Pruning Threshold</th>
<th>Lattice Error Rate</th>
<th>Avg. Hyps / Segment</th>
<th>Segment Sets</th>
<th>Natural Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Types</td>
<td>Tokens</td>
</tr>
<tr>
<td>0.0</td>
<td>27.3</td>
<td>11.7</td>
<td>94029</td>
<td>1393099</td>
</tr>
<tr>
<td>0.1</td>
<td>37.9</td>
<td>2.35</td>
<td>35278</td>
<td>134252</td>
</tr>
<tr>
<td>0.2</td>
<td>41.1</td>
<td>2.1</td>
<td>17132</td>
<td>63267</td>
</tr>
<tr>
<td>0.3</td>
<td>43.2</td>
<td>2.0</td>
<td>7288</td>
<td>26913</td>
</tr>
</tbody>
</table>

Error analysis of natural confusion pairs at a threshold of 0.1:
- confusion pairs are 3.3 times more likely to contain the truth than not
- MAP hyp is wrong roughly half the time
- 26 distinct confusion pairs occurring roughly 6860 times
  - 5 of these are homonyms
SVMs in LVCSR

Test Set: 2991 instances of the 21 acoustic confusion pairs

*Gini* SVMs were trained for each pair

- features: score-spaces derived from MMI-trained HMM word models
- training data extracted from continuous speech via forced alignment
- models trained over data by MMI, e.g., a V:5 / B:5 model pair
  - these models were used to generate features for the SVMs
- avoids generating ASR lattices over training data

Test Results Over 25 Hour Test Set:

Confidence based combination of SVM and baseline hyps

In 18 of 21 cases, WER decreased over the word pairs

LVCSR WER decreased from 45.6% to 45.5% with p-value < 0.001

Current Research: scaling up to vast collections of classifiers
Conclusion

In Particular: Novel procedure to incorporate SVMs into LVCSR

- risk-based lattice cutting transforms LVCSR to problems more easily solved by binary classifiers

In General: New framework for trying novel recognition and modeling strategies without losing the advantages of well-trained HMM systems

- Risk-based lattice segmentation was originally developed for MBR search provides general search space refinement and supports novel estimation techniques

Implications for novel techniques

- This analysis associates modeling `fixes' with specific search space choices
- Specific problems can be extracted for specialized classifiers
- e.g. front / back vowel classifiers
- Unsentimental analysis of overall gains that can be expected
thanks!