Word Recognition with Conditional Random Fields

Jeremy Morris
12/03/2009

Outline

- Background
- Word Recognition – CRF Model
- Pilot System - TIDIGITS
- Larger Vocabulary - WSJ
- Future Work

Background

- Conditional Random Fields (CRFs)
  - Discriminative probabilistic sequence model
  - Directly defines a posterior probability $P(Y|X)$ of a label sequence $Y$ given a set of observations $X$

$$P(Y | X) = \frac{\exp \left( \sum_k \sum_j \lambda_k s_j(x, y_k) + \sum_j \mu_j f_j(x, y_k, y_{k-1}) \right)}{Z(x)}$$

- The form of the CRF model includes weighted state feature functions and weighted transition feature functions
  - Both types of functions can be defined to incorporate observed inputs

Background

- Our previous work compared CRF models for phone recognition to HMM models

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF (phone classes)</td>
<td>69.92%*</td>
</tr>
<tr>
<td>HMM Tandem16mix (phone classes)</td>
<td>69.34%</td>
</tr>
<tr>
<td>CRF (phone classes + phonological features)</td>
<td>70.63%*</td>
</tr>
<tr>
<td>HMM Tandem16mix (phone classes + phonological features)</td>
<td>69.40%</td>
</tr>
</tbody>
</table>

*Significantly (p<0.05) better than comparable Tandem 16mix triphone system (Morris & Fosler-Lussier 08)

Background

- Problem: How do we make use of CRF classification for word recognition?
  - Attempt to fit CRFs into current state-of-the-art models for speech recognition?
  - Attempt to use CRFs directly?

- Each approach has its benefits
  - Fitting CRFs into a standard framework lets us reuse existing code and ideas (Crandem system)
  - A model that uses CRFs directly opens up new directions for investigation
    - Requires some rethinking of the standard model for ASR
Problem: For a given input signal $X$, find the word string $W$ that maximizes $P(W | X)$.

In an HMM, we would make this a generative problem.

We can drop the $P(X)$ because it does not affect the choice of $W$.

We want to build phone models, not whole word models...

... so we marginalize over the phones and look for the best sequence that fits these constraints.
However - our CRFs model \( P(\Phi | X) \) rather than \( P(X | \Phi) \). This makes the formulation of the problem somewhat different.

We want a formulation that makes use of \( P(\Phi | X) \). We can get that by marginalizing over the phone strings. But the CRF as we formulate it doesn't give \( P(\Phi | X) \) directly.

\[ \arg \max_w P(W | X) = \arg \max_w \sum_{\Phi} P(W, \Phi | X) \]
\[ = \arg \max_w \sum_{\Phi} P(W | \Phi, X) P(\Phi | X) \]

Frame level vs. Phone level:
- Mapping from frame level to phone level may not be deterministic.
- Example: The word "OH" with pronunciation /ow/.
- Consider this sequence of frame labels: ow ow ow ow ow ow.
- This sequence can possibly be expanded many different ways for the word "OH" ("OH", "OH OH", etc.).

\( \Phi \) here is a phone level assignment of phone labels.
\( \Phi \) gives related quantity – \( P(Q | X) \) where \( Q \) is the frame level assignment of phone labels.
Word Recognition

- Frame level vs. Phone segment level
  - This problem occurs because we're using a single state to represent the phone /ow/
  - Phone either transitions to itself or transitions out to another phone
  - We can change our model to a multi-state model and make this decision deterministic
  - This brings us closer to a standard ASR HMM topology
  - ow1 ow2 ow2 ow2 ow2 ow3 ow3
  - Now we see a single "OH" in this utterance

- Multi-state model gives us a deterministic mapping of Q -> Φ
  - Each frame-level assignment Q has exactly one segment level assignment associated with it
  - Potential pitfalls if the multi-state model is inappropriate for the features we are using

Word Recognition

\[ P(\Phi | X) = \sum_{Q} P(\Phi, Q | X) \]
\[ = \sum_{Q} P(\Phi | Q, X) P(Q | X) \]
\[ \approx \sum_{Q} P(\Phi | Q) P(Q | X) \]

- What about P(W|Φ)?
  - Non-deterministic across sequences of words
  - Φ = / ah f eh r /
  - W = ? "a fair"? "affair"?
  - The more words in the string, the more possible combinations can arise

Word Recognition

\[ P(W | X) \approx \sum_{\Phi, Q} P(W | \Phi) P(\Phi | Q) P(Q | X) \]

- What is P(Φ)?
  - Prior probability over possible phone sequences
  - Essentially acts as a "phone fertility/penalty" term – lower probability sequences get a larger boost in weight than higher probability sequences
  - Approximate this with a standard n-gram model
  - Seed it with phone-level statistics drawn from the same corpus used for our language model

Bayes Rule

- P(W) –language model
- P(Φ|W) – dictionary model
- P(Φ) – prior probability of phone sequences
Word Recognition

\[
\arg\max_W P(W | X) \approx \arg\max_{W \in \mathcal{L}} \frac{P(W | \Phi) P(\Phi)}{P(\Phi)} P(\Phi | Q) P(Q | X)
\]

- Our final model incorporates all of these pieces together
- Benefit of this approach – reuse of standard models
  - Each element can be built as a finite state machine (FSM)
  - Evaluation can be performed via FSM composition and best path evaluation as for HMM-based systems (Mohri & Riley, 2002)

Pilot Experiment: TIDIGITS

- First word recognition experiment – TIDIGITS recognition
  - Both isolated and strings of spoken digits, ZERO (or OH) to NINE
  - Male and female speakers
- Training set – 112 speakers total
  - Random selection of 11 speakers held out as development set
  - Remaining 101 speakers used for training as needed

**Pilot Experiment: TIDIGITS**

- Important characteristics of the DIGITS problem:
  - A given phone sequence maps to a single word sequence
  - A uniform distribution over the words is assumed
  - \( P(W | \Phi) \) easy to implement directly as FSM

Pilot Experiment: TIDIGITS

- Implementation
  - Created a composed dictionary and language model FST
  - No probabilistic weights applied to these FSTs – assumption of uniform probability of any digit sequence
  - Modified CRF code to allow composition of above FST with phone lattice
  - Results scored using standard HTK tools
  - Compared to a baseline HMM system trained on the same features

Pilot Experiment: Results

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM (triphone, 1 Gaussian, ~4500 parameters)</td>
<td>1.26%</td>
</tr>
<tr>
<td>HMM (triphone, 16 Gaussians ~120,000 parameters)</td>
<td>0.57%</td>
</tr>
<tr>
<td>CRF (monophone, ~4200 parameters)</td>
<td>1.11%</td>
</tr>
<tr>
<td>CRF (monophone, windowed, ~37000 parameters)</td>
<td>0.57%</td>
</tr>
<tr>
<td>HMM (triphone, 16 Gaussians, MFCCs)</td>
<td>0.25%</td>
</tr>
</tbody>
</table>

- Basic CRF performance falls in line with HMM performance for a single Gaussian model
- Adding more parameters to the CRF enables the CRF to perform as well as the HMM on the same features
Larger Vocabulary

- Wall Street Journal 5K word vocabulary task
  - Bigram language model
  - MLPs trained on 75 speakers, 6488 utterances
    - Cross-validated on 8 speakers, 650 utterances
  - Development set of 10 speakers, 368 utterances for tuning purposes
- Results compared to HMM-Tandem baseline and HMM-MFCC baseline

- Phone penalty model $P(\Phi)$
  - Constructed using the transcripts and the lexicon
  - Currently implemented as a phone pair (bigram) model
  - More complex model might lead to better estimates
    - Planning to explore this in the near future

Larger Vocabulary

- Direct finite-state composition not feasible for this task
  - State space grows too large too quickly
- Instead Viterbi decoding performed using the weighted finite-state models as constraints
  - Time-synchronous beam pruning used to keep time and space usage reasonable

Larger Vocabulary – Initial Results

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM MFCC</td>
<td>9.3%</td>
</tr>
<tr>
<td>HMM Tandem MLP</td>
<td>9.1%</td>
</tr>
<tr>
<td>CRF</td>
<td>12.0%</td>
</tr>
<tr>
<td>CRF (windowed)</td>
<td>11.7%</td>
</tr>
</tbody>
</table>

- Preliminary numbers reported on development set only
- Continuing to tune elements of the system for performance (beam width, weights on language model and phone model, feature transforms, etc.)

Next Steps

- More tuning
  - Continue to work with the development set to find the best parameters for decoding
- Feature selection
  - Examine what features will help this model, especially features that may be useful for the CRF that are not useful for HMMs
- Phone penalty model
  - Currently just a bigram phone model
  - A more interesting model leads to more complexity but may lead to better results

Discussion
References


Background

- Tandem HMM
  - ANN MLP classifiers are trained on labeled speech data
  - Classifiers can be phone classifiers, phonological feature classifiers
  - Classifiers output posterior probabilities for each frame of data
    - E.g. P(Q | X), where Q is the phone class label and X is the input speech feature vector

Idea: Crandem

- Use a CRF model to create inputs to a Tandem-style HMM
  - CRF labels provide a better per-frame accuracy than input MLPs
  - We’ve shown CRFs to provide better phone recognition than a Tandem system with the same inputs
  - This suggests that we may get some gain from using CRF features in an HMM

- Problem: CRF output doesn’t match MLP output
  - MLP output is a per-frame vector of posteriors
  - CRF outputs a probability across the entire sequence

- Solution: Use Forward-Backward algorithm to generate a vector of posterior probabilities
Forward-Backward Algorithm

- Similar to HMM forward-backward algorithm
- Used during CRF training
- Forward pass collects feature functions for the timesteps prior to the current timestep
- Backward pass collects feature functions for the timesteps following the current timestep
- Information from both passes are combined together to determine the probability of being in a given state at a particular timestep

\[ P(y_{i,t} | X) = \frac{\alpha_{i,t} \beta_{i,t}}{Z(x)} \]

- This form allows us to use the CRF to compute a vector of local posteriors \( y \) at any timestep \( t \).
- We use this to generate features for a Tandem-style system
  - Take log features, decorrelate with PCA

Phone Recognition

- Pilot task – phone recognition on TIMIT
  - 61 feature MLPs trained on TIMIT, mapped down to 39 features for evaluation
  - Crandem compared to Tandem and a standard PLP HMM baseline model
  - As with previous CRF work, we use the outputs of an ANN MLP as inputs to our CRF
- Phone class attributes
  - Detector outputs describe the phone label associated with a portion of the speech signal
  - /N/, /d/, /aa/, etc.

Word Recognition

- Second task – Word recognition on WSJ0
  - Dictionary for word recognition has 54 distinct phones instead of 48
  - New CRFs and MLPs trained to provide input features
  - MLPs and CRFs trained on WSJ0 corpus of read speech
  - No phone level assignments, only word transcripts
  - Initial alignments from HMM forced alignment of MFCC features
  - Compare Crandem baseline to Tandem and original MFCC baselines

Results (Fosler-Lussier & Morris 08)

<table>
<thead>
<tr>
<th>Model</th>
<th>Phone Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP HMM reference</td>
<td>68.1%</td>
</tr>
<tr>
<td>Tandem</td>
<td>70.8%</td>
</tr>
<tr>
<td>CRF</td>
<td>69.9%</td>
</tr>
<tr>
<td>Crandem – log</td>
<td>71.1%</td>
</tr>
</tbody>
</table>

* Significantly (p<0.05) improvement at 0.6% difference between models

Initial Results

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC HMM reference</td>
<td>9.12%</td>
</tr>
<tr>
<td>Tandem MLP (39)</td>
<td>8.95%</td>
</tr>
<tr>
<td>Crandem (19) (1 epoch)</td>
<td>8.85%</td>
</tr>
<tr>
<td>Crandem (19) (10 epochs)</td>
<td>9.57%</td>
</tr>
<tr>
<td>Crandem (19) (20 epochs)</td>
<td>9.98%</td>
</tr>
</tbody>
</table>

* Significant (p<0.05) improvement at roughly 1% difference between models
Word Recognition

- CRF performs about the same as the baseline systems
- But further training of the CRF tends to degrade the result of the Crandem system
  - Why?
  - First thought – maybe the phone recognition results are deteriorating (overtraining)

Word Recognition

- Further training of the CRF tends to degrade the result of the Crandem system
  - Why?
  - First thought – maybe the phone recognition results are deteriorating (overtraining)
    - Not the case
  - Next thought – examine the pattern of errors between iterations

Word Recognition

- Training the CRF tends to degrade the result of the Crandem system
  - Why?
  - First thought – maybe the phone recognition results are deteriorating (overtraining)
    - Not the case
  - Next thought – examine the pattern of errors between iterations
    - There doesn’t seem to be much of a pattern here, other than a jump in substitutions
    - Word identity doesn’t give a clue – similar words wrong in both lists

Initial Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Phone Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC HMM reference</td>
<td>70.09%</td>
</tr>
<tr>
<td>Tandem MLP (39)</td>
<td>75.58%</td>
</tr>
<tr>
<td>Crandem (19) (1 epoch)</td>
<td>72.77%</td>
</tr>
<tr>
<td>Crandem (19) (10 epochs)</td>
<td>72.81%</td>
</tr>
<tr>
<td>Crandem (19) (20 epochs)</td>
<td>72.93%</td>
</tr>
</tbody>
</table>

* Significant (p<0.05) improvement at roughly 0.07% difference between models

<table>
<thead>
<tr>
<th>Model</th>
<th>Total Errors</th>
<th>Insertions</th>
<th>Deletions</th>
<th>Subs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crandem (1 epoch)</td>
<td>542</td>
<td>57</td>
<td>144</td>
<td>341</td>
</tr>
<tr>
<td>Crandem (10 epochs)</td>
<td>622</td>
<td>77</td>
<td>145</td>
<td>400</td>
</tr>
<tr>
<td>Shared Errors</td>
<td>429</td>
<td>37</td>
<td>131*</td>
<td>261**</td>
</tr>
<tr>
<td>(1-&gt;10)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* 29 deletions are substitutions in one model and deletions in the other
**50 of these subs are different words between the epoch 1 and epoch 10 models
Word Recognition

MARCH vs. LARGE
Iteration 1
0 0 m 0.98271 l 0.0078177 en 0.0052043 em 0.00621807
0 1 m 0.978378 en 0.0063144 l 0.0050046 en 0.00180805
0 2 m 0.98055 en 0.0057963 l 0.002341182 fth 0.00183429
0 3 m 0.98379 en 0.00679143 l 0.0020919
0 4 m 0.935116 aa 0.0268982 em 0.0060147 l 0.0071382
0 5 m 0.710183 aa 0.234062 en 0.011164 w 0.0104974 l 0.009005

Iteration 10
0 0 m 0.949523 l -4.73606 en -4.80113 em -4.80113
0 1 m -0.0218056 en -5.09492 l -5.29322 en -6.31551
0 2 m -0.01646 en -5.1494 fth -5.70124 fth -6.85755
0 3 m -0.0186163 en -6.99293 l -5.92934 w -6.32205
0 4 m -0.0674021 aa -3.61607 en -4.75667 l -4.94296
0 5 m -0.343222 aa -1.496 en -4.4674 e -4.5962 l -4.71001

Word Recognition

Additional issues
- Crandem results sensitive to format of input data
  - Posterior probability inputs to the CRF give very poor results on word recognition.
  - I suspect is related to the same issues described previously
- Crandem results also require a much smaller vector after PCA
  - MLP uses 39 features – Crandem only does well once we reduce to 19 features
  - However, phone recognition results improve if we use 39 features in the Crandem system (72.77% - 74.22%)