CRFs for ASR: Extending to Word Recognition

Jeremy Morris
02/27/2009

speech & language technologies @osu

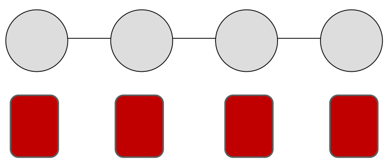
THE OHIO STATE UNIVERSITY

Outline

- Background
- Word Recognition – CRANDEM
- Word Recognition – Direct CRF
- 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



Background

$$P(Y \mid X) = \frac{\exp \sum_k \left(\sum_i \lambda_i s_i(x, y_k) + \sum_j \mu_j t_j(x, y_k, y_{k-1}) \right)}{Z(x)}$$

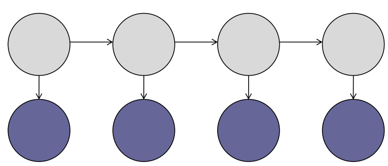
- CRFs extend maximum entropy models by adding weighted transition functions
  - Both types of functions can be defined to incorporate observed inputs

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
  - A model that uses CRFs directly opens up new directions for investigation

Background

- Tandem HMM
  - Generative probabilistic sequence model
  - Uses outputs of a discriminative model (e.g. ANN MLPs) as input feature vectors for a standard HMM



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

7

Background

- Tandem HMM
  - Posterior feature vectors are used by an HMM as inputs
  - In practice, posteriors are not used directly
    - Log posterior outputs or "linear" outputs are more frequently used
      - "linear" here means outputs of the MLP with no application of a softmax function
    - Since HMMs model phones as Gaussian mixtures, the goal is to make these outputs look more "Gaussian"
    - Additionally, Principle Components Analysis (PCA) is applied to features to decorrelate features for diagonal covariance matrices

8

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

9

Idea: Crandem

- 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

10

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

11

Forward-Backward Algorithm

$$P(y_{i,t} \mid X) = \frac{\alpha_{i,t}\beta_{i,t}}{Z(x)}$$

- 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

12

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
    - /t/, /d/, /aa/, etc.

Results (Fosler-Lussier & Morris 08)

Model	Phone Accuracy
PLP HMM reference	68.1%
Tandem	70.8%
CRF	69.9%
Crandem – log	71.1%

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

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

Initial Results

Model	WER
MFCC HMM reference	9.12%
Tandem MLP (39)	8.95%
Crandem (19) (1 epoch)	**8.85%**
Crandem (19) (10 epochs)	9.57%
Crandem (19) (20 epochs)	9.98%

* Significant ($p \leq 0.05$) 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)

Initial Results

Model	Phone Accuracy
MFCC HMM reference	70.09%
Tandem MLP (39)	75.58%
Crandem (19) (1 epoch)	72.77%
Crandem (19) (10 epochs)	72.81%
Crandem (19) (20 epochs)	72.93%

* Significant ($p \leq 0.05$) improvement at roughly 0.07% difference between models

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

Initial Results

Model	**Total Errors**	**Insertions**	**Deletions**	**Subs.**
Crandem (1 epoch)	542	57	144	341
Crandem (10 epochs)	622	77	145	400
Shared Errors	429	37	131* (102)	261** (211)
New Errors (1->10)	193	40	35	118

* 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

- 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

Word Recognition

- Further training of the CRF tends to degrade the result of the Crandem system
  - Why?
  - Current thought – perhaps the reduction in scores of the correct result is impacting the overall score
    - This appears to be happening in at least some cases, though it is not sufficient to explain everything

Word Recognition

MARCH vs. LARGE

Iteration 1

0	0	m	0.952271	**l**	**0.00878177**	en	0.00822043	em	0.00821897		
0	1	m	0.978378	em	0.00631441	**l**	**0.00500046**	en	0.00180805		
0	2	m	0.983655	em	0.00579973	**l**	**0.00334182**	hh	0.00128429		
0	3	m	0.980379	em	0.00679143	**l**	**0.00396782**	w	0.00183199		
0	4	m	0.935156	aa	0.0268882	em	0.00860147	**l**	**0.00713632**		
0	5	m	0.710183	aa	0.224002	em	0.0111564	w	0.0104974	**l**	**0.009005**

Iteration 10

0	0	m	0.982478	em	0.00661739	en	0.00355534	n	0.00242626	**l**	**0.001504**
0	1	m	0.989681	em	0.00626308	**l**	**0.00116445**	en	0.0010961		
0	2	m	0.991131	em	0.00610071	**l**	**0.00111827**	en	0.000643053		
0	3	m	0.989432	em	0.00598472	**l**	**0.00145113**	aa	0.00127722		
0	4	m	0.958312	aa	0.0292846	em	0.00523174	**l**	**0.00233473**		
0	5	m	0.757673	aa	0.225989	em	0.0034254	**l**	**0.00291158**		

Word Recognition

MARCH vs. LARGE - logspace

Iteration 1

0	0	m	-0.0489053	**l**	**-4.73508**	en	-4.80113	em	-4.80131		
0	1	m	-0.0218596	em	-5.06492	**l**	**-5.29822**	en	-6.31551		
0	2	m	-0.01648	em	-5.14994	**l**	**-5.70124**	hh	-6.65755		
0	3	m	-0.0198163	em	-4.99209	**l**	**-5.52954**	w	-6.30235		
0	4	m	-0.0670421	aa	-3.61607	em	-4.75582	**l**	**-4.94256**		
0	5	m	-0.342232	aa	-1.4961	em	-4.49574	w	-4.55662	**l**	**-4.71001**

Iteration 10

0	0	m	-0.017677	em	-5.01805	en	-5.6393	n	-6.02141	l	**-6.49953**
0	1	m	-0.0103729	em	-5.07308	**l**	**-6.75551**	en	-6.816		
0	2	m	-0.0089087	em	-5.09935	**l**	**-6.79597**	en	-7.34928		
0	3	m	-0.0106245	em	-5.11855	**l**	**-6.53542**	aa	-6.66307		
0	4	m	-0.0425817	aa	-3.53069	em	-5.25301	**l**	**-6.05986**		
0	5	m	-0.277504	aa	-1.48727	em	-5.67654	**l**	**-5.83906**		

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%)

Word Recognition

- A Different Approach
  - Instead of feeding these into an HMM as features, can we decode directly off the CRF?
  - Yes, but it requires some changes to the typical formulation for word recognition

Review - Word Recognition

$$\arg\max_{W} P(W \mid X)$$

- Problem: For a given input signal X, find the word string W that maximizes P(W|X)

Review - Word Recognition

$$\arg\max_{W} P(W \mid X) = \arg\max_{W} \frac{P(X \mid W)P(W)}{P(X)}$$

- 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

Review - Word Recognition

$$\arg\max_{W} P(W \mid X) = \arg\max_{W} P(X \mid W)P(W)$$

- 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 is the same for any W we are examining

Review - Word Recognition

$$\arg\max_{W} P(W \mid X) = \arg\max_{W} P(X \mid W)P(W)$$

- We want to build phone models, not whole word models…

Review - Word Recognition

$$\arg\max_W P(W \mid X) = \arg\max_W P(X \mid W)P(W)$$
$$= \arg\max_W \sum_\Phi P(X \mid \Phi)P(\Phi \mid W)P(W)$$

- We want to build phone models, not whole word models…
- … so we marginalize over the phones

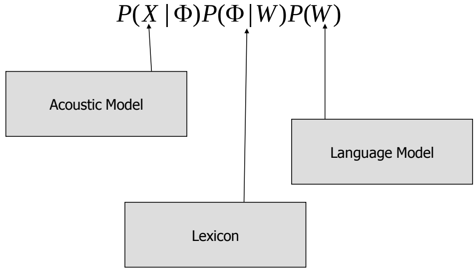
Review - Word Recognition

$$\arg\max_W P(W \mid X) = \arg\max_W P(X \mid W)P(W)$$
$$= \arg\max_W \sum_\Phi P(X \mid \Phi)P(\Phi \mid W)P(W)$$
$$\approx \arg\max_{W,\Phi} P(X \mid \Phi)P(\Phi \mid W)P(W)$$

- We want to build phone models, not whole word models…
- … so we marginalize over the phones
- and approximate by looking only at the maximal sequence

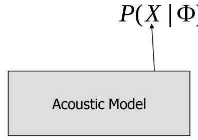
Review - Word Recognition

$$P(X \mid \Phi)P(\Phi \mid W)P(W)$$



Word Recognition

$$P(X \mid \Phi)P(\Phi \mid W)P(W)$$



- However - our CRFs model P(Φ|X) rather than P(X|Φ)
  - This makes the formulation of the problem somewhat different

Word Recognition

$$\arg\max_W P(W \mid X)$$

- We want a formulation that makes use of P(Φ|X)

Word Recognition

$$\arg\max_W P(W \mid X) = \arg\max_W \sum_\Phi P(W,\Phi \mid X)$$
$$= \arg\max_W \sum_\Phi P(W \mid \Phi, X)P(\Phi \mid X)$$

- We want a formulation that makes use of P(Φ|X)
- We can get that by marginalizing over the phone strings
- But … the CRF doesn't really give us P(Φ|X)

Word Recognition

$$P(W \mid \Phi, X)P(\Phi \mid X)$$

- Φ here is a *phone segment level* assignment of phone labels
- CRF gives related quantity – P(Q|X) where Q is the *frame level* assignment of phone labels

Word Recognition

- Frame level vs. Phone segment level
  - Mapping from frame level to phone level may not be deterministic
  - Example: The number "OH" with pronunciation /ow/
  - Consider this sequence of frame labels:

    ow ow ow ow ow ow ow
  - How many separate utterances of the word "OH" does that sequence represent?

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
  - What if we change this representation to a multi-state model?
    - This brings us closer to the HMM topology

    ow1 ow2 ow2 ow2 ow2 ow3 ow3
  - Now we can see a single "OH" in this utterance

Word Recognition

$$\begin{aligned} P(\Phi \mid X) &= \sum_{Q} P(\Phi, Q \mid X) \\ &= \sum_{Q} P(\Phi \mid Q, X)P(Q \mid X) \\ &\approx \sum_{Q} P(\Phi \mid Q)P(Q \mid X) \end{aligned}$$

- 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 problem – what if the multi-state model is inappropriate for the features we've chosen?

Word Recognition

$$\begin{aligned} \arg\max_{W} P(W \mid X) &= \arg\max_{W} \sum_{\Phi} P(W \mid \Phi, X)P(\Phi \mid X) \\ &\approx \arg\max_{W} \sum_{\Phi, Q} P(W \mid \Phi, X)P(\Phi \mid Q)P(Q \mid X) \\ &\approx \arg\max_{W} \sum_{\Phi, Q} P(W \mid \Phi)P(\Phi \mid Q)P(Q \mid X) \end{aligned}$$

- Replacing P(Φ|X) we now have a model with our CRF in it
- What about P(W| Φ,X)?
  - Conditional independence assumption gives P(W| Φ)

Word Recognition

$$P(W \mid X) \approx \sum_{\Phi, Q} P(W \mid \Phi)P(\Phi \mid Q)P(Q \mid X)$$

- 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

$$P(W \mid \Phi) = \frac{P(\Phi \mid W)P(W)}{P(\Phi)}$$

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

Word Recognition

- What is P(Φ) ?
  - Prior probability over possible phone sequences
  - Use a standard n-gram model over phone sequences to approximate this?
    - Seed it with our lexicon as well as phone-level statistics drawn from the same corpus we have built our language model over
- Benefit of this approach – reuse of standard models
  - Each model can be built as an FSM lattice
  - Best path evaluation through FSM composition

Word Recognition

$$\arg\max_{W} P(W \mid X) \approx \arg\max_{W,\Phi,Q} \frac{P(\Phi \mid W)P(W)}{P(\Phi)} P(\Phi \mid Q)P(Q \mid X)$$

- Our final model incorporates all of these pieces
  - We are also making one final assumption – that we can substitute the maximum value (best path) for a particular Q sequence from our CRF rather than marginalizing across all instances of the same Q
  - This is a standard assumption for HMM decoding, as well as for CRF decoding

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

$$\arg\max_{W} P(W \mid X) \approx \arg\max_{W,\Phi,Q} \frac{P(\Phi \mid W)P(W)}{P(\Phi)} P(\Phi \mid Q)P(Q \mid X)$$

- 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|Φ) easy to implement directly as FSTs

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 written to MLF file and scored using standard HTK tools
    - Results compared to HMM system trained on same features

Pilot Experiment: TIDIGITS

- Features
  - Choice of multi-state model for CRF may not be best fit with neural network posterior outputs
    - The neural network abstracts away distinctions among different parts of the phone across time (by design)
  - Phone Classification (Gunawardana et al., 2005)
    - Feature functions designed to take MFCCs, PLP or other traditional ASR inputs and use them in CRFs
    - Gives the equivalent of a single Gaussian per state model
    - Fairly easy to adapt to our CRFs

Pilot Experiment: TIDIGITS

- Labels
  - Unlike TIMIT, TIDIGITS files do not come with phone-level labels
  - To generate these labels for CRF training, weights derived from TIMIT were used to force align a state-level transcript
  - This label file was then used for training the CRF

Pilot Experiment: Results

Model	WER
HMM (monophone, 1 Gaussian)	1.26%
HMM (triphone, 1 Gaussian)	1.55%
HMM (monophone, 32 Gaussians)	0.07%
HMM (triphone, 32 Gaussians)	0.18%
CRF	1.19%

- CRF Performance falls in line with the single Gaussian models
  - CRF with these features achieves ~63% accuracy on TIMIT phone task, compared to ~69% accuracy of triphone HMM, 32 models
  - These results may not be the best we can get for the CRF

Future Work

- Move to a more interesting data set
  - WSJ 5K words test – compare to Crandem and Tandem baselines
- Work on the P(W|Φ) model
  - The WSJ 5K words task will require a more robust model – including computation of P(Φ) and the inclusion of an actual language model P(W)
  - For P(W) we can use the same n-gram language model that we use in an HMM to construct a lattice
  - For P(Φ), we can use our lexicon and our language model to generate synthetic strings of phones to gather statistics from (the LM is provided for the 5K word WSJ0 task)

Future Work

- Is it possible to compute P(W|Φ) directly?
  - Probability of a string of words given a string of phones
  - This looks like it could be computed as a Maximum Entropy Markov Model,

$$P(W \mid \Phi) = \prod_n P(w_n \mid w_{n-1}, \Phi)$$

  - This merges the lexicon and the language model into a single computation
  - It allows us to have a model of our pronunciation and our language that are tied not only to each other, but also to our acoustics
  - It also gives a model that provides a way of finding words with only partial evidence of the word

What Next?

$$P(W \mid \Phi) = \prod_n P(w_n \mid w_{n-1}, \Phi)$$

- But it also opens up a new set of questions
  - What are the features used as inputs (Bag of phones? Bag of biphones? Triphones? Indexed phones?)
  - How do we properly compare words of different phone lengths?
  - How do we account for "non-words" and out of vocabulary words?
    - These are two different effects – an OOV word is a real word that our system doesn't know. A non-word is a group of phones that are actually the ending of one word and the beginning of another

Discussion

References

- J. Lafferty et al, “Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data”, Proc. ICML, 2001
- A. Berger, “A Brief MaxEnt Tutorial”, http://www.cs.cmu.eu/afs/cs/user/aberger/www/html/tutorial/tutorial.html
- R. Rosenfeld, “Adaptive statistical language modeling: a maximum entropy approach”, PhD thesis, CMU, 1994
- A. Gunawardana et al, “Hidden Conditional Random Fields for phone classification”, Proc. Interspeech, 2005