Conditional Random Fields for ASR

Jeremy Morris
11/23/2009

Outline
- Background
  - Maximum Entropy models and CRFs
  - CRF Example
  - ASR experiments with CRFs

Background
- Conditional Random Fields (CRFs)
  - Discriminative probabilistic sequence model
  - Used successfully in various domains such as part of speech tagging and named entity recognition
  - Directly defines a posterior probability of a label sequence Y given an input observation sequence X - P(Y|X)

Background – Discriminative Models
- Contrast with generative models
  - e.g. GMMs, HMMs
  - Find the best model of the distribution to generate the observed features
  - Find the label y that maximizes the joint probability P(y,x) for observed features x
    - More parameters to model than discriminative models
    - More assumptions about feature independence required

Background – Sequential Models
- Used to classify sequences of data
  - HMMs the most common example
  - Find the most probable sequence of class labels
  - Class labels depend not only on observed features, but on surrounding labels as well
  - Must determine transitions as well as state labels
Background – Sequential Models

- Sample Sequence Model - HMM

Conditional Random Fields

- A probabilistic, discriminative classification model for sequences
  - Based on the idea of Maximum Entropy Models (Logistic Regression models) expanded to sequences

Maximum Entropy Models

- Probabilistic, discriminative classifiers
  - Compute the conditional probability of a class $y$ given an observation $x$ – $P(y|x)$
  - Build up this conditional probability using the principle of *maximum entropy*
    - In the absence of evidence, assume a uniform probability for any given class
    - As we gain evidence (e.g. through training data), modify the model such that it supports the evidence we have seen but keeps a uniform probability for unseen hypotheses

Maximum Entropy Example

- Suppose we have a bin of candies, each with an associated label (A,B,C, or D)
  - Each candy has multiple colors in its wrapper
  - Each candy is assigned a label randomly based on some distribution over wrapper colors
  * Example inspired by Adam Berger’s Tutorial on Maximum Entropy

Maximum Entropy Example

- For any candy with a red label pulled from the bin:
  - $P(A|\text{red}) + P(B|\text{red}) + P(C|\text{red}) + P(D|\text{red}) = 1$
  - Infinite number of distributions exist that fit this constraint
  - The distribution that fits with the idea of maximum entropy is:
    - $P(A|\text{red})=0.25$
    - $P(B|\text{red})=0.25$
    - $P(C|\text{red})=0.25$
    - $P(D|\text{red})=0.25$

Maximum Entropy Example

- Now suppose we add some evidence to our model
  - We note that 80% of all candies with red labels are either labeled A or B
    - $P(A|\text{red}) + P(B|\text{red}) = 0.8$
  - The updated model that reflects this would be:
    - $P(A|\text{red}) = 0.4$
    - $P(B|\text{red}) = 0.4$
    - $P(C|\text{red}) = 0.1$
    - $P(D|\text{red}) = 0.1$
  - As we make more observations and find more constraints, the model gets more complex
Maximum Entropy Models

- “Evidence” is given to the MaxEnt model through the use of feature functions
  - Feature functions provide a numerical value given an observation
  - Weights on these feature functions determine how much a particular feature contributes to a choice of label
    - In the candy example, feature functions might be built around the existence or non-existence of a particular color in the wrapper
    - In NLP applications, feature functions are often built around words or spelling features in the text

Maximum Entropy – Feature Funcs.

- Feature functions for a maxent model associate a label and an observation
  - For the candy example, feature functions might be based on labels and wrapper colors
  - In an NLP application, feature functions might be based on labels (e.g. POS tags) and words in the text

Maximum Entropy – Feature Funcs.

\[
s(y, x) = \begin{cases} 
1 & \text{iff}(y = \text{NOUN}, x = \text{"dog"}) \\
0 & \text{otherwise}
\end{cases}
\]

- Example: MaxEnt POS tagging
  - Associates a tag (NOUN) with a word in the text ("dog")
  - This function evaluates to 1 only when both occur in combination
    - At training time, both tag and word are known
    - At evaluation time, we evaluate for all possible classes and find the class with highest probability

Maximum Entropy – Feature Funcs.

\[
s_1(y, x) = \begin{cases} 
1 & \text{iff}(y = \text{NOUN}, x = \text{"dog"}) \\
0 & \text{otherwise}
\end{cases}
\]

\[
s_2(y, x) = \begin{cases} 
1 & \text{iff}(y = \text{VERB}, x = \text{"dog"}) \\
0 & \text{otherwise}
\end{cases}
\]

- These two feature functions would never fire simultaneously
  - Each would have its own lambda-weight for evaluation

Maximum Entropy – Feature Funcs.

\[
s_i(y, X, n) = \begin{cases} 
1 & \text{iff}(y = \text{NOUN}, x_n = \text{"dog"}, x_{n+1} = \text{"my"}) \\
0 & \text{otherwise}
\end{cases}
\]

- MaxEnt models do not make assumptions about the independence of features
  - Depending on the application, feature functions can benefit from context
Maximum Entropy – Feature Funcs.

- Other feature functions possible beyond simple word/tag association
  - Does the word have a particular prefix?
  - Does the word have a particular suffix?
  - Is the word capitalized?
  - Does the word contain punctuation?
- Ability to integrate many complex but sparse observations is a strength of maxent models.

Conditional Random Fields

- Extends the idea of maxent models to sequences
  - Label sequence $Y$ has a Markov structure
  - Observed sequence $X$ may have any structure

State functions help determine the identity of the state

Transition functions add associations between transitions from one label to another

CRF extends the maxent model by adding weighted transition functions
- Both types of functions can be defined to incorporate observed inputs

\[
P(Y \mid X) = \frac{\exp \sum_k (\sum_i \lambda_i s_i(x, y_k) + \sum_j \mu_j t_j(x, y_k, y_{k-1}))}{Z(x)}
\]
Conditional Random Fields

- Feature functions defined as for maxent models
  - Label/observation pairs for state feature functions
  - Label/label/observation triples for transition feature functions
    - Often transition feature functions are left as “bias features” – label/label pairs that ignore the attributes of the observation

$s(y, y', x) = \begin{cases} 
1 & \text{iff}(y = \text{NOUN}, y' = \text{DET}, x = \text{"dog"}) \\
0 & \text{otherwise}
\end{cases}$

Example: CRF POS tagging

- Associates a tag (NOUN) with a word in the text ("dog") AND with a tag for the prior word (DET)
- This function evaluates to 1 only when all three occur in combination
  - At training time, both tag and word are known
  - At evaluation time, we evaluate for all possible tag sequences and find the sequence with highest probability (Viterbi decoding)

SLaTe Experiments - Background

- Goal: Integrate outputs of speech attribute detectors together for recognition
  - e.g. Phone classifiers, phonological feature classifiers
- Attribute detector outputs highly correlated
  - Stop detector vs. phone classifier for /t/ or /d/
- Accounting for correlations in HMM
  - Ignore them (decreased performance)
  - Full covariance matrices (increased parameters)
  - Explicit decorrelation (e.g. PCA)

Speech Attributes

- Phonological feature attributes
  - Detector outputs describe phonetic features of a speech signal
    - Place, Manner, Voicing, Vowel Height, Backness, etc.
  - A phone is described with a vector of feature values
- Phone class attributes
  - Detector outputs describe the phone label associated with a portion of the speech signal
    - /t/, /d/, /aa/, etc.

SLaTe Experiments - Background

- CRFs for ASR
  - Phone Classification (Gunawardana et al., 2005)
    - Uses sufficient statistics to define feature functions
  - Different approach than NLP tasks using CRFs
    - Define binary feature functions to characterize observations
  - Our approach follows the latter method
    - Use neural networks to provide “soft binary” feature functions (e.g. posterior phone outputs)

SLaTe Experiments

- Implemented CRF models on data from phonetic attribute detectors
  - Performed phone recognition
  - Compared results to Tandem/HMM system on same data
- Experimental Data
  - TIMIT corpus of read speech
SLaTe Experiments - Attributes

- Attribute Detectors
  - ICSI QuickNet Neural Networks
- Two different types of attributes
  - Phonological feature detectors
    - Place, Manner, Voicing, Vowel Height, Backness, etc.
    - N-ary features in eight different classes
    - Posterior outputs -- P(Place=dental | X)
  - Phone detectors
    - Neural networks output based on the phone labels
    - Trained using PLP 12+deltas

SLaTe Experiments - Setup

- CRF code
  - Built on the Java CRF toolkit from Sourceforge
  - Performs maximum log-likelihood training
  - Uses Limited Memory BGFS algorithm to perform minimization of the log-likelihood gradient

Experimental Setup

\[
s_{1t/f}(y, x) = \begin{cases} 
    \text{NN}_f(x), & \text{if } y = t/l \\
    0, & \text{otherwise}
\end{cases}
\]

- Feature functions built using the neural net output
  - Each attribute/label combination gives one feature function
  - Phone class: \(s_{t/l, t/l}\) or \(s_{t/l, s/l}\)
  - Feature class: \(s_{t/l, \text{stop}}\) or \(s_{t/l, \text{dental}}\)

Feature Combinations

- CRF model supposedly robust to highly correlated features
  - Makes no assumptions about feature independence
- Tested this claim with combinations of correlated features
  - Phone class outputs + Phono. Feature outputs
  - Posterior outputs + transformed linear outputs
- Also tested whether linear, decorrelated outputs improve CRF performance

Results (Morris & Fosler-Lussier ‘08)

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF (phone posteriors)</td>
<td>67.32%</td>
</tr>
<tr>
<td>CRF (phone posteriors – realigned)</td>
<td>69.92%***</td>
</tr>
<tr>
<td>Tandem[3] 4mix (phones)</td>
<td>68.07%</td>
</tr>
<tr>
<td>Tandem[3] 16mix (phones)</td>
<td>69.34%</td>
</tr>
<tr>
<td>CRF (phone, feat. linear KL)</td>
<td>66.37%</td>
</tr>
<tr>
<td>CRF (phone, feat. lin-KL – realigned)</td>
<td>68.99%**</td>
</tr>
<tr>
<td>Tandem[3] 4mix (phone feat.)</td>
<td>68.30%</td>
</tr>
<tr>
<td>Tandem[3] 16mix (phone feat.)</td>
<td>69.13%</td>
</tr>
<tr>
<td>CRF (phones+feas)</td>
<td>68.43%</td>
</tr>
<tr>
<td>CRF (phones+feas – realigned)</td>
<td>70.63%***</td>
</tr>
<tr>
<td>Tandem[3] 16mix (phones+feas)</td>
<td>69.40%</td>
</tr>
</tbody>
</table>

* Significantly (p<0.05) better than comparable CRF monophone system
** Significantly (p<0.05) better than comparable Tandem 4mix triphone system
*** Significantly (p<0.05) better than comparable Tandem 16mix triphone system
Conclusions

- Using correlated features in the CRF model did not degrade performance
  - Extra features improved performance for the CRF model across the board
- Viterbi realignment training significantly improved CRF results
  - Improvement did not occur when best HMM-aligned transcript was used for training

Extension – Word Decoding

- Use the CRF model to generate features for an HMM
  - “Crandem” system (Morris & Fosler-Lussier, 09)
  - Performance similar to a similarly trained Tandem HMM system
- Direct word word decoding over CRF lattice
  - In progress – preliminary experiments over restricted vocabulary (digits) match state of the art performance
  - Currently working on extending to larger vocabulary

References


Initial Results (Morris & Fosler-Lussier, 06)

<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>Phone Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tandem [1] (phones)</td>
<td>20,000+</td>
<td>60.82%</td>
</tr>
<tr>
<td>Tandem [3] (phones) 4mix</td>
<td>420,000+</td>
<td>68.07%*</td>
</tr>
<tr>
<td>CRF [1] (phones)</td>
<td>5280</td>
<td>67.32%*</td>
</tr>
<tr>
<td>Tandem [1] (feas)</td>
<td>14,000+</td>
<td>61.85%</td>
</tr>
<tr>
<td>Tandem [3] (feas) 4mix</td>
<td>360,000+</td>
<td>68.30%*</td>
</tr>
<tr>
<td>CRF [1] (feas)</td>
<td>4464</td>
<td>65.45%*</td>
</tr>
<tr>
<td>Tandem [1] (phones/feas)</td>
<td>34,000+</td>
<td>61.72%</td>
</tr>
<tr>
<td>Tandem [3] (phones/feas) 4mix</td>
<td>774,000+</td>
<td>68.46%</td>
</tr>
<tr>
<td>CRF (phones/feas)</td>
<td>7392</td>
<td>66.43%*</td>
</tr>
</tbody>
</table>

* Significantly (p<0.05) better than comparable Tandem monophone system
* Significantly (p<0.05) better than comparable CRF monophone system

Feature Combinations - Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Phone Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF (phone posteriors)</td>
<td>67.32%</td>
</tr>
<tr>
<td>CRF (phone linear KL)</td>
<td>66.80%</td>
</tr>
<tr>
<td>CRF (phone post+linear KL)</td>
<td>68.13%*</td>
</tr>
<tr>
<td>CRF (phono. feature post.)</td>
<td>65.45%</td>
</tr>
<tr>
<td>CRF (phono. feature linear KL)</td>
<td>66.37%</td>
</tr>
<tr>
<td>CRF (phono. feature post+linear KL)</td>
<td>67.36%*</td>
</tr>
</tbody>
</table>

* Significantly (p<0.05) better than comparable posterior or linear KL systems

Viterbi Realignment

- Hypothesis: CRF results obtained by using only pre-defined boundaries
  - HMM allows “boundaries” to shift during training
  - Basic CRF training process does not
- Modify training to allow for better boundaries
  - Train CRF with fixed boundaries
  - Force align training labels using CRF
  - Adapt CRF weights using new boundaries
Future Work

- Recently implemented stochastic gradient training for CRFs
  - Faster training, improved results
- Work currently being done to extend the model to word recognition
- Also examining the use of transition functions that use the observation data
  - Crandem system does this with improved results for phone recognition

Conditional Random Fields

- Example – POS tagging (Lafferty, 2001)
  - Adding more complex and sparse features improved the CRF performance
    - Capitalization?
    - Suffixes? (-ly, -ing, -ogy, -ed, etc.)
    - Contains a hyphen?

<table>
<thead>
<tr>
<th>Model</th>
<th>Error</th>
<th>OOV error</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>5.69%</td>
<td>45.99%</td>
</tr>
<tr>
<td>CRF</td>
<td>5.55%</td>
<td>48.05%</td>
</tr>
<tr>
<td>CRF+</td>
<td>4.27%</td>
<td>23.76%</td>
</tr>
</tbody>
</table>

- Based on the framework of Markov Random Fields
  - A CRF iff the graph of the label sequence is an MRF when conditioned on a set of input observations (Lafferty et al., 2001)
Conditional Random Fields

- A CRF defined by a weighted sum of state and transition functions
- Both types of functions can be defined to incorporate observed inputs
- Weights are trained by maximizing the likelihood function via gradient descent methods

\[
P(Y | X) = \frac{\exp \sum_k (\sum_i \lambda_i s_i(x, y_i) + \sum_j \mu_i f_i(x, y_i, y_{i-1}))}{Z(x)}
\]