

Conditional Random Fields For Speech and Language Processing

Jeremy Morris
10/27/2008



Outline

- Background
- Maximum Entropy models and CRFs
- CRF Example
- SLaTe experiments with CRFs

2

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 - $P(Y|X)$

3

Background – Discriminative Models

- Directly model the association between the observed features and labels for those features
 - e.g. neural networks, maximum entropy models
 - Attempt to model boundaries between competing classes
- Probabilistic discriminative models
 - Give conditional probabilities instead of hard class decisions
 - Find the class y that maximizes $P(y|x)$ for observed features x

4

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

5

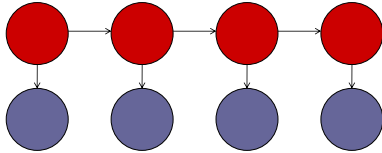
Background – Sequential Models

- 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

6

Background – Sequential Models

- Sample Sequence Model - HMM



7

Conditional Random Fields

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

8

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

9

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

10

Maximum Entropy Example

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

11

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
 - $P(A|red) + P(B|red) = 0.8$
 - The updated model that reflects this would be:
 - $P(A|red) = 0.4$
 - $P(B|red) = 0.4$
 - $P(C|red) = 0.1$
 - $P(D|red) = 0.1$
 - As we make more observations and find more constraints, the model gets more complex

12

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

13

Maximum Entropy Models

$$P(y|x) = \frac{\exp \sum_i \lambda_i s_i(x, y)}{\exp \sum_k \sum_i \lambda_i s_i(x, y_k)}$$

- The maxent model for k competing classes
- Each *feature function* $s(x, y)$ is defined in terms of the input observation (x) and the associated label (y)
- Each feature function has an associated weight (λ)

14

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

15

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

16

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

17

Maximum Entropy – Feature Funcs.

- MaxEnt models do not make assumptions about the independence of features
 - Depending on the application, feature functions can benefit from context

$$s_1(y, X, n) = \begin{cases} 1 & \text{iff } (y = \text{NOUN}, x_n = \text{"dog"}, x_{n-1} = \text{"my"}) \\ 0 & \text{otherwise} \end{cases}$$

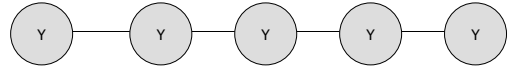
18

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.

19

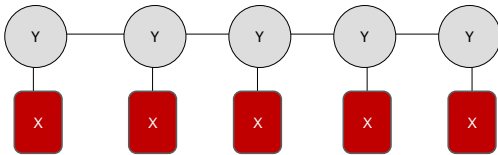
Conditional Random Fields



- Extends the idea of maxent models to sequences

20

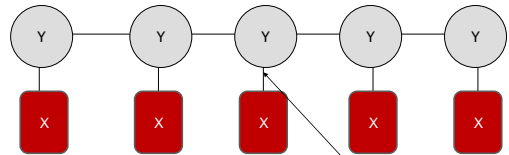
Conditional Random Fields



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

21

Conditional Random Fields

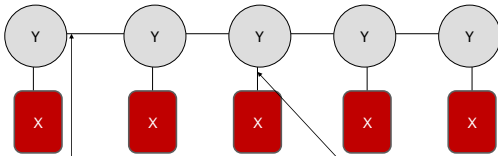


- Extends the idea of maxent models to sequences
 - Label sequence Y has a Markov structure
 - Observed sequence X may have any structure

State functions help determine the identity of the state

22

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

Transition functions add associations between transitions from one label to another

State functions help determine the identity of the state

23

Conditional Random Fields

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

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

24

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

25

Conditional Random Fields

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

- 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)

26

Conditional Random Fields

- Example – POS tagging (Lafferty, 2001)
 - State feature functions defined as word/label pairs
 - Transition feature functions defined as label/label pairs
 - Achieved results comparable to an HMM with the same features

Model	Error	OOV error
HMM	5.69%	45.99%
CRF	5.55%	48.05%

27

Conditional Random Fields

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

Model	Error	OOV error
HMM	5.69%	45.99%
CRF	5.55%	48.05%
CRF+	4.27%	23.76%

28

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. Karhunen-Loeve transform)

29

SLaTe Experiments - Background

- 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.

30

SLaTe Experiments - Background

- CRFs for ASR
 - Phone Classification (Gunawardana et al., 2005)
 - Uses sufficient statistics to define feature functions
 - Start with an HMM, using Gaussian mixture models for state likelihoods
 - Mathematically transform the HMM to a CRF
 - Any HMM can be rewritten as a CRF (the reverse is not true)
 - Use this transformed HMM to provide feature functions and starting weights for a CRF model

31

SLaTe Experiments - Background

$$\begin{aligned}f_s^{(M1)}(s, o, t) &= \delta(s_t = s) o_t & \lambda_s^{(M1)} &= \frac{\mu_s}{\sigma_s^2} \\f_s^{(M2)}(s, o, t) &= \delta(s_t = s) o_t^2 & \lambda_s^{(M2)} &= -\frac{1}{2\sigma_s^2} \\f_s^{(Occ)}(s, o, t) &= \delta(s_t = s) & \lambda_s^{(Occ)} &= -\frac{1}{2} \left(\log 2\pi\sigma^2 + \frac{\mu}{\sigma^2} \right) \\f_{s's'}^{(Tr)}(s, o, t) &= \delta(s_t = s) \delta(s_{t-1} = s') & \lambda_{s's}^{(Tr)} &= \log a_{s's}\end{aligned}$$

Feature functions and associated lambda-weights computed using a sufficient statistics model per (Gunawardana et al., 2005)

32

SLaTe Experiments - Background

- Phone Classification (Gunawardana et al., 2005)
 - 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)
 - We have investigated a sufficient statistics model
 - MLP pre-processing usually provides a better result in our domain
 - Experiments are ongoing

33

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

34

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

35

SLaTe Experiments - Setup

- CRF code
 - Built on the Java CRF toolkit from Sourceforge
 - <http://crf.sourceforge.net>
 - Performs maximum log-likelihood training
 - Uses Limited Memory BGFS algorithm to perform minimization of the log-likelihood gradient

36

Experimental Setup

$$s_{/t/,f}(y, x) = \begin{cases} NN_f(x), & \text{if } y = /t/ \\ 0, & \text{otherwise} \end{cases}$$

- Feature functions built using the neural net output
 - Each attribute/label combination gives one feature function
 - Phone class: $S_{N/N}$ or $S_{N/sl}$
 - Feature class: $S_{N/stop}$ or $S_{N,dental}$

37

Experimental Setup

- Baseline system for comparison
 - Tandem/HMM baseline (Hermansky et al., 2000)
 - Use outputs from neural networks as inputs to gaussian-based HMM system
 - Built using HTK HMM toolkit
- Linear inputs
 - Better performance for Tandem with linear outputs from neural network
 - Decorrelated using a Karhunen-Loeve (KL) transform

38

Initial Results (Morris & Fosler-Lussier, 06)

Model	Params	Phone Accuracy
Tandem [1] (phones)	20,000+	60.82%
Tandem [3] (phones) 4mix	420,000+	68.07%*
CRF [1] (phones)	5280	67.32%*
Tandem [1] (feas)	14,000+	61.85%
Tandem [3] (feas) 4mix	360,000+	68.30%*
CRF [1] (feas)	4464	65.45%*
Tandem [1] (phones/feas)	34,000+	61.72%
Tandem [3] (phones/feas) 4mix	774,000+	68.46%
CRF (phones/feas)	7392	68.43%*

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

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

40

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

Feature Combinations - Results

Model	Phone Accuracy
CRF (phone posteriors)	67.32%
CRF (phone linear KL)	66.80%
CRF (phone post+linear KL)	68.13%*
CRF (phono. feature post.)	65.45%
CRF (phono. feature linear KL)	66.37%
CRF (phono. feature post+linear KL)	67.36%*

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

41

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

42

Viterbi Realignment - Results

Model	Accuracy
CRF (phone posteriors)	67.32%
CRF (phone posteriors – realigned)	69.92%***
Tandem[3] 4mix (phones)	68.07%
Tandem[3] 16mix (phones)	69.34%
CRF (phono. fea. linear KL)	66.37%
CRF (phono. fea. lin-KL – realigned)	68.99%**
Tandem[3] 4mix (phono fea.)	68.30%
Tandem[3] 16mix (phono fea.)	69.13%
CRF (phones+feas)	68.43%
CRF (phones+feas – realigned)	70.63%***
Tandem[3] 16mix (phones+feas)	69.40%

* 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

44

Current Work - Crandem Systems

- Idea – use the CRF model to generate features for an HMM
 - Similar to the Tandem HMM systems, replacing the neural network outputs with CRF outputs
 - Use forward-backward algorithm to compute posterior probabilities for each frame of input data
 - Preliminary phone-recognition experiments show promise
 - Preliminary attempts to incorporate CRF features at the word level are less promising

45

Current Work – Crandem Systems

- Baseline systems for comparison
 - Tandem/HMM baseline (Hermansky et al., 2000)
- Models trained using TIMIT corpus as above
 - Tested for word recognition on WSJ0 corpus
 - Corpus of read article from Wall Street Journal archives
 - TIMIT not built for word recognition experiments – WSJ0 is

46

Current Work - Crandem Systems

Model	Word Accuracy
Baseline Tandem MLP [phone classes]	90.30%
Crandem MLP [phone classes]	90.95%
Baseline Tandem MLP [phone + phono]	91.26%
Crandem MLP [phone + phono]	91.31%

47

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 – so far, no improvement in word recognition

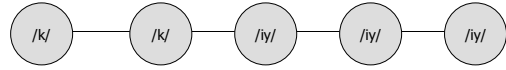
48

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.edu/afs/cs/user/ab Berger/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

49

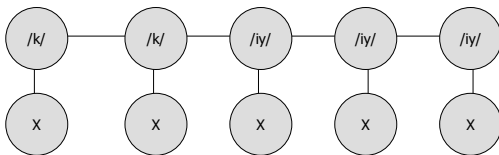
Conditional Random Fields



- Based on the framework of Markov Random Fields

50

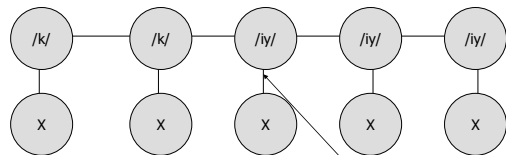
Conditional Random Fields



- 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)

51

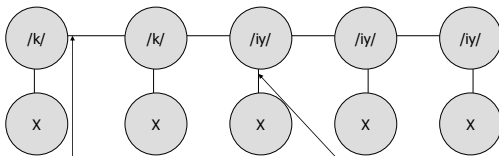
Conditional Random Fields



- Based on the framework of Markov Random Fields
 - A CRF iff the graph of the label sequence is an MRF when conditioned on the input observations

52

Conditional Random Fields



- Based on the framework of Markov Random Fields
 - A CRF iff the graph of the label sequence is an MRF when conditioned on the input observations

53

Conditional Random Fields

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

- 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

54