Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequential Data

John Lafferty
Andrew McCallum
Fernando Pereira

Papers & Tutorials

- Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequential Data
  - Lafferty
  - McCallum
  - Pereira

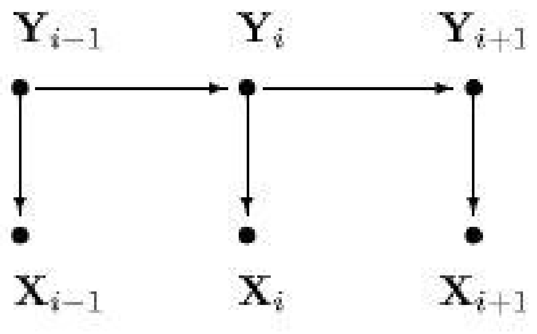
  http://www.aladdin.cs.cmu.edu/papers/pdfs/y2001/crf.pdf

- Efficient Training of Conditional Random Fields
  - Hanna Wallach

  http://www. cogsci.ed.ac.uk/~osborn...wallach.ps.gz

Directed Graphical Models

- HMMs
  - Cannot represent multiple interacting features or long range dependencies between observed elements.

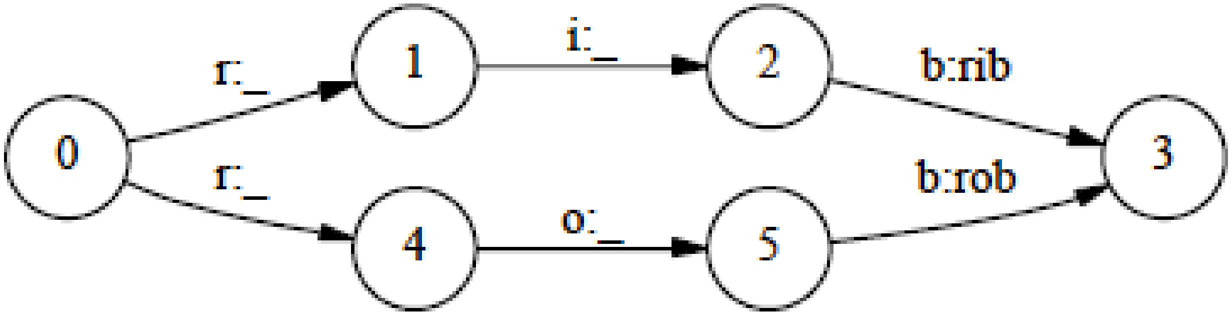


- MEMMs
  - Label-bias problem: the probability transitions leaving any given state must sum to one.

Conditional Random Fields: CRF

- Conditional probabilistic sequential models
- Undirected graphical models
- A single log-linear distribution over the joint probability of an entire label sequence given a particular observation sequence.
- Weights of different features at different states can be traded off against each other.

Label Bias Problem



- States with low entropy next state distributions will take little notice of observations.
- Two solutions:
  - Change the state-transition structure
  - Start with fully-connected model and let the training procedure figure out a good structure.

- CRF accounts for whole state sequence at once by letting some transitions vote more strongly than others depending on the corresponding observations

- Score mass will not be conserved: Individual transitions can ‘amplify’ or ‘dampen’ the mass they receive

Definition

X : r.v over data sequences to be labeled

Y: r.v over corresponding label sequences.

(**X**,**Y**) is a CRF, in case, when conditioned on **X**, the r.v $\mathbf{Y}_v$ obey the Markov property

$$p(\mathbf{Y}_v \,|\, \mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v \,|\, \mathbf{X}, \mathbf{Y}_w, w \sim v)$$

i.e. The probability of $\mathbf{Y}_v$ given **X** and all other r.v is equal to the probability of $\mathbf{Y}_v$ given **X** and those r.v corresponding to nodes neighbouring **v** in G(**V**,**E**)

CRF is a random field globally conditioned on the observation **X**

- By fundamental theorem of random fields, the joint distribution over the label sequence **Y** given **X** has the form

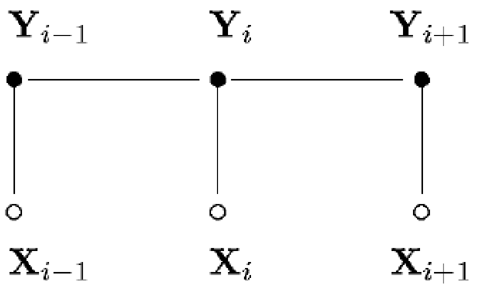
$$p_\theta(\mathbf{y} \mid \mathbf{x}) \propto \exp\left(\sum_{e \in E, k} \lambda_k f_k(e, \mathbf{y}|_e, \mathbf{x}) + \sum_{v \in V, k} \mu_k g_k(v, \mathbf{y}|_v, \mathbf{x}) \right)$$

- Where λ_k and μ_k are lagrrang multipliers for each feature $\mathbf{f_k}$ and $\mathbf{g_k}$
- $\mathbf{f_k}$ is some feature of the entire observation sequence and the labels $\mathbf{y_e}$ and $\mathbf{y_{e-1}}$
- $\mathbf{g_k}$ is a feature of label $\mathbf{y_v}$ and the observation sequence **x**
- $\mathbf{y|_s}$ is the set of components of y associated with the vertices in subgraph **S**

HMM like CRF model

Single feature for each state-state pair **(y’, y)**and state-observation pair **(y,x)** in the data set used to train CRF.

$$
\begin{aligned}
f_{y',y}(<u,v>, \mathbf{y}|_{<u,v>}, \mathbf{x}) &= \delta(\mathbf{y}_u, y')\,\delta(\mathbf{y}_v, y) \\
&= \begin{cases} 1 & \text{if } \mathbf{y}_u = y' \text{ and } \mathbf{y}_v = y \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$



$$
\begin{aligned}
g_{y,x}(v, \mathbf{y}|_{v}, \mathbf{x}) &= \delta(\mathbf{y}_v, y)\,\delta(\mathbf{x}_v, x) \\
&= \begin{cases} 1 & \text{if } \mathbf{y}_v = y \text{ and } \mathbf{x}_v = x \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

The parameters $\lambda_{y',y}$ and $\mu_{y,x}$ correspond to these features are equivalent to the logarithm of the HMM transition probability **p(y’|y)** and emission probability **p(x|y)** for a chain structure, the conditional probability of a label sequence can be expressed in matrix form

for each position **i** in the observed sequence **x**, we define matrix r.v.

$$M_i(\mathbf{x}) = [M_i(y', y \,|\, \mathbf{x})]$$

$$\begin{aligned} M_i(y', y \,|\, \mathbf{x}) &= \exp\left(\Lambda_i(y', y \,|\, \mathbf{x})\right) \\ \Lambda_i(y', y \,|\, \mathbf{x}) &= \textstyle\sum_k \lambda_k \, f_k(e_i, \mathbf{Y}|_{e_i} = (y', y), \mathbf{x}) + \\ &\quad \textstyle\sum_k \mu_k \, g_k(v_i, \mathbf{Y}|_{v_i} = y, \mathbf{x}) \ , \end{aligned}$$

where e_i is the edge with labels $(\mathbf{y_{i-1}}, \mathbf{y_i})$ and $\mathbf{v_i}$ is the vertex with label $\mathbf{y_i}$

$$p_\theta(\mathbf{y} \,|\, \mathbf{x}) \propto \exp\left(\begin{array}{l} \sum \textstyle\sum_k \lambda_k \, f_k(e_i, \mathbf{Y}|_{e_i} = (y', y), \mathbf{x}) + \\ \displaystyle\sum_i \textstyle\sum_k \mu_k \, g_k(v_i, \mathbf{Y}|_{v_i} = y, \mathbf{x}) \ , \end{array} \right)$$

this may not satisfy axioms of probability, so we define normalization factor **Z** to ensure that it is indeed a joint probability distribution over r.v represented by nodes in **G** the normalization function is the (start, stop) entry of the product of these matrices

$$
\begin{aligned}
Z_\theta(\mathbf{x}) &= \left(M_1(\mathbf{x})\, M_2(\mathbf{x}) \cdots M_{n+1}(\mathbf{x})\right)_{\mathsf{start},\mathsf{stop}} \\
&= \left(\prod_{i=1}^{n+1} M_i(\mathbf{x})\right)_{\mathsf{start},\mathsf{stop}}
\end{aligned}
$$

so, the conditional probability of label sequence y is

$$
p_\theta(\mathbf{y} \,|\, \mathbf{x}) = \frac{\prod_{i=1}^{n+1} M_i(\mathbf{y}_{i-1}, \mathbf{y}_i \,|\, \mathbf{x})}{Z_\theta(\mathbf{x})}
$$

where, $\mathbf{y}_0 = \mathsf{start}$ and $\mathbf{y}_{n+1} = \mathsf{stop}$

Parameter Estimation Problem: Determine the parameters $\theta = (\lambda_1, \lambda_2, \ldots; \mu_1, \mu_2, \ldots)$ from training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ with empirical distribution $\widetilde{p}(\mathbf{x}, \mathbf{y})$

We want to maximize the log-likelihood objective function $\mathcal{O}(\theta)$

$$
\begin{aligned}
\mathcal{O}(\theta) &= \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)} \,|\, \mathbf{x}^{(i)}) \\
&\propto \sum_{\mathbf{x},\mathbf{y}} \widetilde{p}(\mathbf{x}, \mathbf{y}) \log p_\theta(\mathbf{y} \,|\, \mathbf{x})
\end{aligned}
$$

Improved Iterative Scaling (IIS) Algorithms

update the weights as $\lambda_k \leftarrow \lambda_k + \delta\lambda_k$ and $\mu_k \leftarrow \mu_k + \delta\mu_k$ for appropriately chosen $\delta\lambda_k$ and $\delta\mu_k$

The aim is to identify a growth transformation that updates the parameters of our model so as to increase the log-likelihood as much as possible.

for IIS update $\delta\lambda_k$ for edge feature **$\mathbf{f_k}$** is the solution of

$$\widetilde{E}[f_k] \overset{\text{def}}{=} \sum_{\mathbf{x},\mathbf{y}} \widetilde{p}(\mathbf{x},\mathbf{y}) \sum_{i=1}^{n+1} f_k(e_i, \mathbf{y}|_{e_i}, \mathbf{x})$$

$$= \sum_{\mathbf{x},\mathbf{y}} \widetilde{p}(\mathbf{x})\, p(\mathbf{y}\,|\,\mathbf{x}) \sum_{i=1}^{n+1} f_k(e_i, \mathbf{y}|_{e_i}, \mathbf{x})\, e^{\,\delta\lambda_k T(\mathbf{x},\mathbf{y})}$$

where $T(\mathbf{x},\mathbf{y})$ is the *total feature count*

$$T(\mathbf{x},\mathbf{y}) \overset{\text{def}}{=} \sum_{i,k} f_k(e_i, \mathbf{y}|_{e_i}, \mathbf{x}) + \sum_{i,k} g_k(v_i, \mathbf{y}|_{v_i}, \mathbf{x})$$

T(x,y) is a global property of **(x,y)** and efficently computer the exponential sums on the RHS of these equation is a problem.

Lafferty proposes two algorithms:

Alogrithm S: we define *slack feature* by

$$s(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} S - \sum_i \sum_k f_k(e_i, \mathbf{y}|_{e_i}, \mathbf{x}) - \sum_i \sum_k g_k(v_i, \mathbf{y}|_{v_i}, \mathbf{x})$$

for each index i=0,….,n+1 we define forward vectors

$$\alpha_0(y \,|\, \mathbf{x}) = \begin{cases} 1 & \text{if } y = \mathsf{start} \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_i(\mathbf{x}) = \alpha_{i-1}(\mathbf{x})\, M_i(\mathbf{x})$$

similarly the backward vectors are defined by,

$$\beta_{n+1}(y \,|\, \mathbf{x}) = \begin{cases} 1 & \text{if } y = \mathsf{stop} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_i(\mathbf{x})^\top = M_{i+1}(\mathbf{x})\, \beta_{i+1}(\mathbf{x})$$

so, update equations become,

$$\delta\lambda_k = \frac{1}{S}\log\frac{\widetilde{E}f_k}{Ef_k}, \quad \delta\mu_k = \frac{1}{S}\log\frac{\widetilde{E}g_k}{Eg_k}$$

where, $$Ef_k = \sum_{\mathbf{x}} \widetilde{p}(\mathbf{x}) \sum_{i=1}^{n+1} \sum_{y',y} f_k(e_i, \mathbf{y}|_{e_i} = (y', y), \mathbf{x}) \times \frac{\alpha_{i-1}(y' \mid \mathbf{x})\, M_i(y', y \mid \mathbf{x})\, \beta_i(y \mid \mathbf{x})}{Z_\theta(\mathbf{x})}$$

$$Eg_k = \sum_{\mathbf{x}} \widetilde{p}(\mathbf{x}) \sum_{i=1}^{n} \sum_{y} g_k(v_i, \mathbf{y}|_{v_i} = y, \mathbf{x}) \times \frac{\alpha_i(y \mid \mathbf{x})\, \beta_i(y \mid \mathbf{x})}{Z_\theta(\mathbf{x})} .$$

The factors involving the forward and backward vectors in the above equations have the same meaning as for HMMs.

The rate of convergence is governed by step size which is inversely proportional to constant **S**. But **S** is generally quite large.

Algorithm T: keeps track of partial T totals

it accumulates feature expectations into counters indexed by **T(x)**

where, $T(\mathbf{x}) \overset{\text{def}}{=} \max_{\mathbf{y}} T(\mathbf{x}, \mathbf{y})$

we use forward-backward recurrences to compute the expectation $\mathbf{a}_{\mathbf{k,t}}$ of feature $\mathbf{f}_{\mathbf{k}}$ and $\mathbf{b}_{\mathbf{k,t}}$ of feature $\mathbf{g}_{\mathbf{k}}$ given that **T(x) = t.**

parameter updates are $\delta\lambda_k = \log \beta_k$ and $\delta\mu_k = \log \gamma_k$

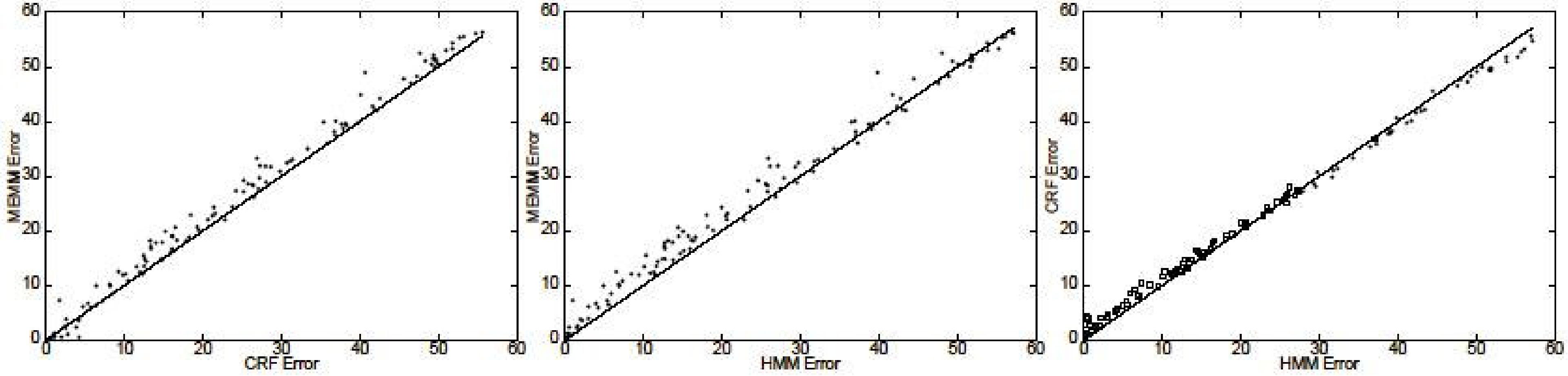
where β_k and γ_k re unique positive roots to the following polynomial equations,

$$\sum_{i=0}^{T_{\max}} a_{k,t}\, \beta_k^t = \widetilde{E} f_k\,, \quad \sum_{i=0}^{T_{\max}} b_{k,t}\, \gamma_k^t = \widetilde{E} g_k$$

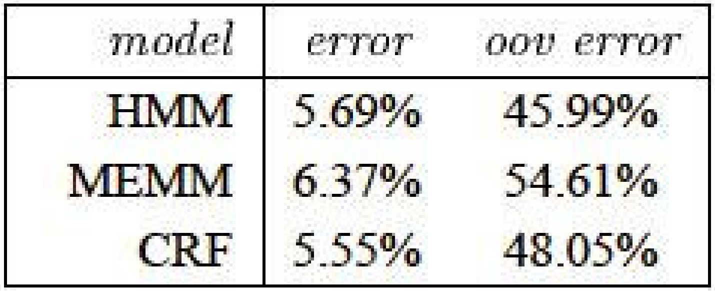
which can be computed by Newton’s method

Experiments

- Modeling label bias problem
  - 2,000 training and 500 test samples generated by HMM
  - CRF error is 4.6 %
  - MEMM error is 42%
- Modeling mixed order sources
  - CRF converge in 500 iterations – slow
  - MEMMs converge in 100 iterations

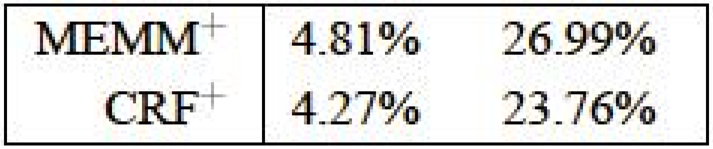


- POS tagging experiment
  - First order HMM, HEMM and CRF model
  - 50%-50% test-train split



model	*error*	*oov error*
HMM	5.69%	45.99%
MEMM	6.37%	54.61%
CRF	5.55%	48.05%

  - Add a small amount of topographic features



MEMM^+	4.81%	26.99%
CRF^+	4.27%	23.76%

- CRF training is slow… so we can uses MEMM paramater vector as a starting point for training the corresponding CRF.
- We can use boosting: treat each label as a separate classification problem.