Advanced Artificial Intelligence

Part II. Statistical NLP

Conditional Random Fields

Wolfram Burgard, Luc De Raedt,
Bernhard Nebel, Lars Schmidt-Thieme

13.12.2005

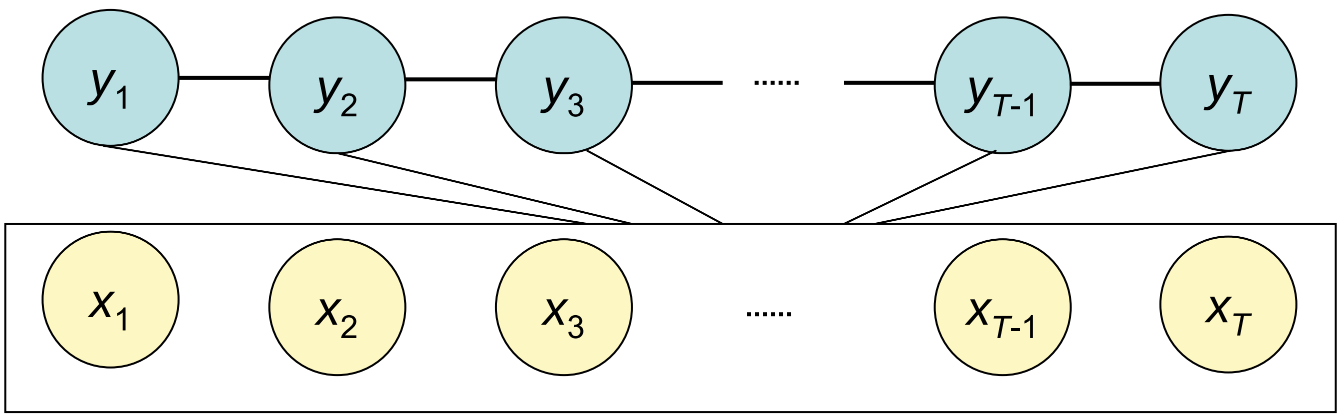
Slides by Bernd Gutmann

Outline

- Introduction
- Label-Bias Problem
- Potential Functions vs. Features
- Definition of P(Y|X)
- Forward-Backward Algorithm for CRFs
- Example
- Possible Classifiers
- Learning
- Literature

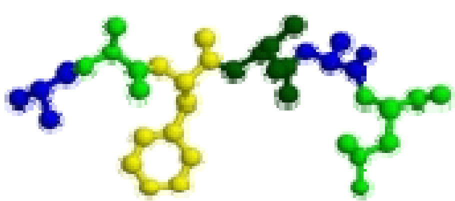
Introduction

- a CRF defines the conditional probability for sequences $P(y_1,\ldots,y_T \mid x_1,\ldots,x_T)$
- undirected graph structure
- global normalization instead of local normalization (e.g. HMMs)
- each potential function can read the complete input X
- for sequences a first order chain is used as graph structure:



Tagging Sequences

- Given: input sequence over the alphabet $\mathcal{Y}$
  $X = x_1, x_2, \ldots, x_T$
- Wanted: output sequence over the alphabet $\mathcal{X}$
  $Y = y_1, y_2, \ldots, y_T$
- Applications
  - Part-of-speech tagging
    X = **`He, drives, with, his, bike`**
    Y = **`noun, verb, preposition, pronoun, noun`**
  - predicting the secondary structure of proteins
    X = **`A, F, A, R, L, M, M, A`**
    Y = **`he, he, st, st, st, he, st, he`**



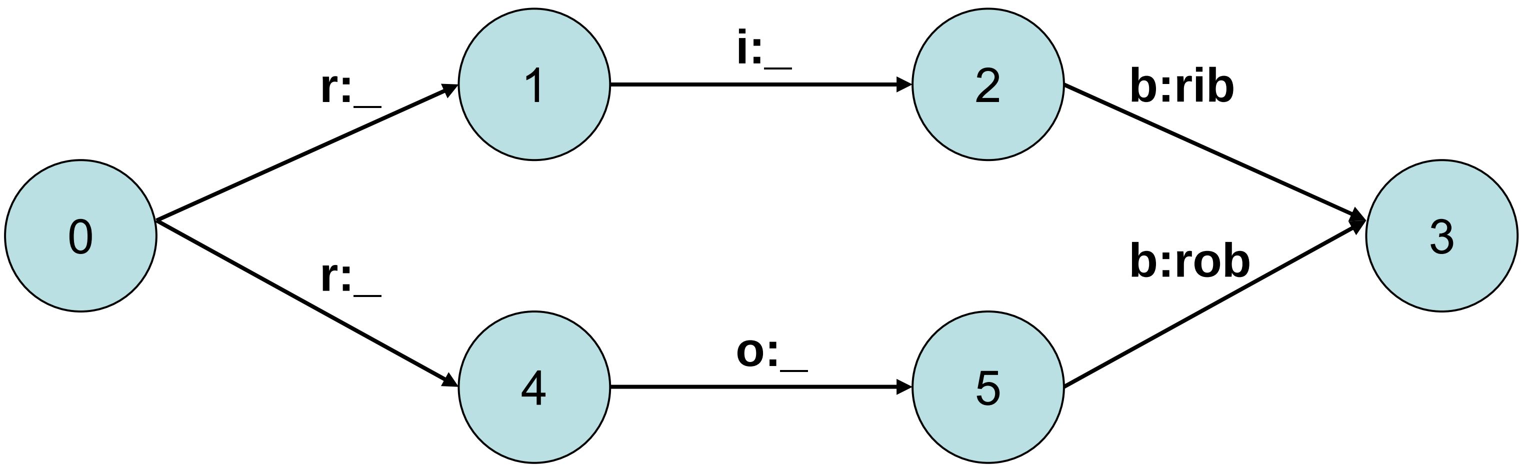
Primary structure

Strand

Helix

Label Bias Problem

A finite state machine to separate between rib and rob:



- the probability distribution for the next state is conditioned per state
- states with low entropy next state distributions will take little notice of observations (State 0)
- states with only one successor state even ignore the input (State 1 and 4)

Potential Functions vs. Features

- A *potential function* is a positive-valued function $F(x_1,...,x_l)$
- It doesn't return a probability but higher values indicate a higher preference for the variable assignment
- By normalizing the potentials we get probabilities (next slide)
- A *feature* is a function that returns a binary value

$$g(x_1,\ldots,x_l) = \begin{cases} 1 & \text{if} \quad x_2 = \text{sunny} \\ 0 & \quad \text{else} \end{cases}$$

- One can use features to represent potentials

$$F(x_1,\ldots,x_l) = \exp\left[\Omega \cdot G(x_1,\ldots x_l)^T\right]$$

where $\Omega = (\omega_1,\ldots,\omega_n)$ is a weight vector
and $G(X) = (g_1(X),\ldots,g_n(X))$ is the feature vector.
exp is needed to get a positive value

Probability Distribution for Sequences

- Normalizing the value of the potentials returns a probability value in the interval [0,1]
- *X* and *Y* are sequences of length *T* then

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left[\sum_{t=1}^{T} \Psi_t(y_t, X) + \Psi_{t-1,t}(y_{t-1}, y_t, X)\right]$$

$$\Psi_t(y_t, X) = \sum_{a \in A} \beta_a \cdot g_a(y_t, X) \qquad \Psi_{t-1,t}(y_{t-1}, y_t, X) = \sum_{b \in B} \lambda_b \cdot f_b(y_{t-1}, y_t, X)$$

- where the normalization constant is

$$Z(X) = \sum_{Y'} \exp\left[\sum_{t=1}^{T} \Psi_t(y'_t, X) + \Psi_{t-1,t}(y'_{t-1}, y'_t, X)\right]$$

(*Y*' runs over all possible output sequences of length *T*)

Forward-Backward Algorithm (1)

- In difference to HMMs we get accumulated potential values instead of probabilities
- forward procedure (assumption y_0=`start`)

$$\alpha_k(t) := \sum_{\substack{y_1,\ldots,y_t \\ y_t=k}} \exp\left[\sum_{t'=1}^{t} \Psi_{t'}(y_{t'}, X) + \Psi_{t'-1,t}(y_{t'-1}, y_{t'}, X)\right]$$

- recursive calculation of α

$$\alpha_k(1) = \exp\left(\Psi_1(k, X) + \Psi_{0,1}(\mathbf{start}, k, X)\right)$$

$$\alpha_k(t) = \sum_{k' \in \Upsilon} \left[\exp\left(\Psi_t(k', X) + \Psi_{t-1,t}(k, k', X)\right)\right] \cdot \alpha_{k'}(t-1)$$

Forward-Backward Algorithm (2)

- backward procedure

$$\beta_k(t) := \sum_{\substack{y_t,\ldots,y_T \\ y_t = k}} \exp\left[\sum_{t'=t+1}^{T} \Psi_{t'}(y_{t'}, X) + \Psi_{t'-1,t}(y_{t'-1}, y_{t'}, X)\right]$$

- recursive calculation of β

$$\beta_k(T) = 1$$

$$\beta_k(t) = \sum_{k' \in \Upsilon} \left[\exp\left(\Psi_t(k, X) + \Psi_{t-1,t}(k', k, X)\right)\right] \cdot \beta_{k'}(t+1)$$

Forward-Backward Algorithm (3)

- The normalization constant

$$Z(X) = \sum_{Y'} \exp\left[\sum_{t=1}^{T} \Psi_t(y'_t, X) + \Psi_{t-1,t}(y'_{t-1}, y'_t, X) \right]$$

(*Y'* runs over all possible output sequences of length *T*)

- Can now be computed for arbitrary *t* with

$$Z(X) = \sum_{k \in \Upsilon} \alpha_k(t) \cdot \beta_k(t)$$

- when we choose *t*=*T* the forward step is sufficient:

$$Z(X) = \sum_{k \in \Upsilon} \alpha_k(t)$$

Example

- we want to compute $Z(X)$ for the input sequence X=**in1**, **in3**, **in2**
- feature functions are

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in1} \\ 0 & else \end{cases}$$

- weights for features are $\beta_1=1$ $\lambda_1=2$ $\lambda_2=10$
- the output alphabet $\mathcal{Y}$=**{out1,out2}**

Example, Z(X)

X=**in1**, **in3**, **in2** β1=1 λ1=2 λ2=10

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in}1 \\ 0 & else \end{cases}$$

	t=1	*t*=2	*t*=3
k=**out2**			
k=**out1**	exp(10)		

$$\alpha_{\mathbf{out1}}(1) = \exp\left(1 \cdot g_1(\mathbf{out1}, X) + 2 \cdot f_1(\mathbf{start}, \mathbf{out1}, X) + 10 \cdot f2(\mathbf{start}, \mathbf{out1}, X)\right)$$
$$= \exp(0 + 0 + 10)$$

Example, Z(X)

X=**in1**, **in3**, **in2** β1=1 λ1=2 λ2=10

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in1} \\ 0 & else \end{cases}$$

	t=1	*t*=2	*t*=3
k=**out2**	exp(11)		
k=**out1**	exp(10)		

$$\alpha_{\mathbf{out2}}(1) = \exp\left(1 \cdot g_1(\mathbf{out2}, X) + 2 \cdot f_1(\mathbf{start}, \mathbf{out2}, X) + 10 \cdot f2(\mathbf{start}, \mathbf{out2}, X)\right)$$
$$= \exp(1 + 0 + 10)$$

Example, Z(X)

X=**in1**, **in3**, **in2** β1=1 λ1=2 λ2=10

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in1} \\ 0 & else \end{cases}$$

k=**out2**	exp(11)		
k=**out1**	exp(10)	2exp(11)	
	t=1	*t*=2	*t*=3

$$\alpha_{\mathbf{out1}}(2) = \exp\left(1 \cdot g_1(\mathbf{out1}, X) + 2 \cdot f_1(\mathbf{out1}, \mathbf{out1}, X) + 10 \cdot f2(\mathbf{out1}, \mathbf{out1}, X)\right) \cdot \alpha_{\mathbf{out1}}(1) + \exp\left(1 \cdot g_1(\mathbf{out1}, X) + 2 \cdot f_1(\mathbf{out2}, \mathbf{out1}, X) + 10 \cdot f2(\mathbf{out2}, \mathbf{out1}, X)\right) \cdot \alpha_{\mathbf{out2}}(1)$$
$$= \exp(0+1+0) \cdot \exp(10) + \exp(0+0+0) \cdot \exp(11) = 2\exp(11)$$

Example, Z(X)

X=**in1**, **in3**, **in2**   β1=1   λ1=2   λ2=10

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in1} \\ 0 & else \end{cases}$$

	t=1	t=2	t=3
k=**out2**	exp(11)	exp(11)+exp(14)	
k=**out1**	exp(10)	2exp(11)	

$$\alpha_{\mathbf{out2}}(2) = \exp\left(1 \cdot g_1(\mathbf{out2}, X) + 2 \cdot f_1(\mathbf{out1}, \mathbf{out2}, X) + 10 \cdot f2(\mathbf{out1}, \mathbf{out2}, X)\right) \cdot \alpha_{\mathbf{out1}}(1) + \exp\left(1 \cdot g_1(\mathbf{out2}, X) + 2 \cdot f_1(\mathbf{out2}, \mathbf{out2}, X) + 10 \cdot f2(\mathbf{out2}, \mathbf{out2}, X)\right) \cdot \alpha_{\mathbf{out2}}(1)$$

$$= \exp(1+0+0) \cdot \exp(10) + \exp(1+2+0) \cdot \exp(11) = \exp(11) + \exp(14)$$

Example, Z(X)

X=**in1**, **in3**, **in2** β1=1 λ1=2 λ2=10

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in}1 \\ 0 & else \end{cases}$$

	t=1	t=2	t=3
k=**out2**	exp(11)	exp(11)+exp(14)	
k=**out1**	exp(10)	2exp(11)	2exp(13)+exp(11)+exp(14)

$$\begin{aligned} \alpha_{\mathbf{out1}}(3) &= \exp\left(1 \cdot g_1(\mathbf{out1}, X) + 2 \cdot f_1(\mathbf{out1}, \mathbf{out1}, X) + 10 \cdot f2(\mathbf{out1}, \mathbf{out1}, X)\right) \cdot \alpha_{\mathbf{out1}}(2) + \\ &\quad \exp\left(1 \cdot g_1(\mathbf{out1}, X) + 2 \cdot f_1(\mathbf{out2}, \mathbf{out1}, X) + 10 \cdot f2(\mathbf{out2}, \mathbf{out1}, X)\right) \cdot \alpha_{\mathbf{out2}}(2) \\ &= \exp(0+2+0) \cdot 2\exp(11) + \exp(0+0+0) \cdot (\exp(11) + \exp(14)) = 2\exp(13) + \exp(11) + \exp(14) \end{aligned}$$

Example, Z(X)

X=**in1**, **in3**, **in2** β1=1 λ1=2 λ2=10

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \textbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\textbf{in2}, \textbf{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \textbf{in1} \\ 0 & else \end{cases}$$

	t=1	t=2	t=3
k=**out2**	exp(11)	exp(11)+exp(14)	2exp(12)+exp(14)+exp(17)
k=**out1**	exp(10)	2exp(11)	2exp(13)+exp(11)+exp(14)

$$\begin{aligned} \alpha_{\textbf{out2}}(3) &= \exp\left(1 \cdot g_1(\textbf{out2}, X) + 2 \cdot f_1(\textbf{out1}, \textbf{out2}, X) + 10 \cdot f2(\textbf{out1}, \textbf{out2}, X)\right) \cdot \alpha_{\textbf{out1}}(2) + \\ &\quad \exp\left(1 \cdot g_1(\textbf{out2}, X) + 2 \cdot f_1(\textbf{out2}, \textbf{out2}, X) + 10 \cdot f2(\textbf{out2}, \textbf{out2}, X)\right) \cdot \alpha_{\textbf{out2}}(2) \\ &= \exp(1+0+0) \cdot 2\exp(11) + \exp(1+2+0) \cdot (\exp(11) + \exp(14)) = 2\exp(12) + \exp(14) + \exp(17) \end{aligned}$$

Example, Z(X)

X=**in1**, **in3**, **in2** β1=1 λ1=2 λ2=10

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in1} \\ 0 & else \end{cases}$$

k=**out2**	exp(11)	exp(11)+exp(14)	2exp(12)+exp(14)+exp(17)
k=**out1**	exp(10)	2exp(11)	2exp(13)+exp(11)+exp(14)
	t=1	*t*=2	*t*=3

$$Z(X) = [2\exp(13) + \exp(11) + \exp(14)] + [2\exp(12) + \exp(14) + \exp(17)] \approx \mathbf{27\ 830\ 372}$$

Example, P(Y|X)

- We want to compute $P(\mathbf{out1, out2, out2} \mid \mathbf{in1, in3, in2})$

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left[\sum_{t=1}^{T} \Psi_t(y_t, X) + \Psi_{t-1,t}(y_{t-1}, y_t, X) \right]$$

- feature functions are

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\mathbf{in2}, \mathbf{in3}\} \\ 0 & else \end{cases}$$

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \mathbf{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \mathbf{in1} \\ 0 & else \end{cases}$$

- Weights for features are $\beta_1=1$ $\lambda_1=2$ $\lambda_2=10$
- the output alphabet $\mathcal{Y}=\{\mathbf{out1, out2}\}$
- from previous slide: $Z(\mathbf{in1, in3, in2}) \approx$ 27 830 372

Example, P(Y|X)

- We want to compute $P($**out1, out2, out2** | **in1, in3, in2**$)$
- feature functions are

$$g_1(y_t, X) = \begin{cases} 1 & y_t = \texttt{out2} \wedge x_t \neq x_{t-1} \\ 0 & else \end{cases}$$

$$f_1(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} = y_t \wedge x_t \in \{\texttt{in2}, \texttt{in3}\} \\ 0 & else \end{cases}$$

$$f_2(y_{t-1}, y_t, X) = \begin{cases} 1 & y_{t-1} \neq y_t \wedge x_t = \texttt{in1} \\ 0 & else \end{cases}$$

- Calculation of $\sum_{t=1}^{T} \Psi_t(y_t, X) + \Psi_{t-1,t}(y_{t-1}, y_t, X)$

	g_1	f_1	f_2	β_1	λ_1	λ_2	Result
t=1	0	0	1	1	2	10	10
t=2	1	0	0	1	2	10	1
t=3	1	1	0	1	2	10	3
							14

Example, P(Y|X)

- We want to compute *P*(**out1,out2,out2** | **in1,in3,in2**)

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left[\sum_{t=1}^{T} \Psi_t(y_t, X) + \Psi_{t-1,t}(y_{t-1}, y_t, X)\right]$$

- *Z*(**in1,in3,in2)**= 9 834 077 197.0
- From previous slide: $\sum_{t=1}^{T} \Psi_t(y_t, X) + \Psi_{t-1,t}(y_{t-1}, y_t, X) = 14$
- *P*(**out1,out2,out2|in1,in3,in2**) ≈ exp(14) / 27830372 ≈ 0.043

Possible Classifiers

- We have *X* and want to find the best output *Y*
- Predict the output per sequence
  (with Viterbi algorithm)

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

- Predict the output per item
  (with Forward-Backward algorithm)

$$H_t(X) = \arg\max_{1 \le k \le K} P(y_t = k \mid X)$$

where

$$P(y_t = k) = \frac{\alpha_k(t) \cdot \beta_k(t)}{Z(X)}$$

Parameter Learning

- (X_i, Y_i) are a training examples ($1 \le i \le n$)
- goal is to maximize the log-likelihood of the training data

$$J(\Theta) = \log \prod_{i=1}^{n} P(Y_i \mid X_i) \qquad \text{where} \quad \Theta = (\beta_1, \ldots, \beta_{|A|}, \lambda_1, \ldots, \lambda_{|B|})$$

- methods based on gradient descent:
  – Iterative Scaling (Lafferty 2001)
  – Generalized Iterative Scaling (Lafferty 2001)
  – Gradient Tree Boosting (Dietterich 2004)

Literature

- J. Lafferty, A. McCallum, F. Pereira. **Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data**. In *Proceedings of the Eighteenth International Conference on Machine Learning* (ICML-2001), 2001. http://www.aladdin.cs.cmu.edu/papers/pdfs/y2001/crf.pdf
- T. Dietterich, A. Ashenfelter, Y. Bulatov. **Training Conditional Random Fields via Gradient Tree Boosting**. In *Proceedings of the Twenty-First International Conference on Machine Learning* (ICML-2004), 2004. http://web.engr.oregonstate.edu/~tgd/publications/ml2004-treecrf.pdf
- Overview of CRF-related publications http://www.inference.phy.cam.ac.uk/hmw26/crf
- Mallet (an implementation) http://mallet.cs.umass.edu/index.php/Main_Page
- CRF package (an implementation) http://crf.sourceforge.net/introduction
- CRF Toolkit for Matlab http://cs.ubc.ca/~murphyk/Software/CRF/crf.html