

# Conditional Random Fields

A most painless introduction into  
Conditional Random Fields

Stefan Maetschke

University of Queensland

<http://www.itee.uq.edu.au/~stefan>

Introduction

» Background

Linear chain CRFs

Questions

Appendix

# Introduction

# Background

Introduction

» Background

Linear chain CRFs

Questions

Appendix

- Introduced by J. Lafferty, A. McCallum, F. Pereira in 2001 [2].
- Improvement of training algorithm (L-BFGS) pushed field [5].
- CRFs are hot. Explosion in numbers of papers in 2003,2004.
- Good performance esp. for part of speech tagging and name entity recognition.
- Sound mathematics and highly flexible.
- Solve difficulties with HMMs, MEMMs (indep. assumptions, Label-Bias problem).
- Structured data, multi-label multi-class problems, sequential data, graphical models
- Best of generative and classification models
- Successful applications: POS, NER, Transmembrane helix prediction, Secondary structure prediction, Image labeling, ...
- For an introduction read Sutton et al. [4] and Gupta [1].

Introduction

---

Linear chain CRFs

- » Exponential model
- » Parameter estimation
- » Gradient
- » Forward-Backward procedure
- » Optimization
- » Optimization — L-BFGS
- » Summary

Questions

---

Appendix

---

# Linear chain CRFs

# Exponential model

Sequence  $\mathbf{x} = (x_1, x_2, \dots, x_T)$  with labels  $\mathbf{y} = (y_1, y_2, \dots, y_T)$

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \Psi_t(y_t, y_{t-1}, \mathbf{x}) \quad (1)$$

$$\Psi_t(y_t, y_{t-1}, \mathbf{x}) = \exp \left( \sum_i \lambda_i f_i(y_t, y_{t-1}, \mathbf{x}) \right) \quad (2)$$

$$f_i(y_{t-1}, y_t, \mathbf{x}) = \begin{cases} s_k(y_t, \mathbf{x}) \\ t_j(y_{t-1}, y_t, \mathbf{x}) \end{cases} \quad (3)$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^T \Psi_t(y_t, y_{t-1}, \mathbf{x}) \quad (4)$$

Introduction

Linear chain CRFs

» Exponential model

» Parameter estimation

» Gradient

» Forward-Backward procedure

» Optimization

» Optimization — L-BFGS

» Summary

Questions

Appendix

# Parameter estimation

Introduction

Linear chain CRFs

» Exponential model

» Parameter estimation

» Gradient

» Forward-Backward procedure

» Optimization

» Optimization — L-BFGS

» Summary

Questions

Appendix

Dataset  $D = \left\{ (\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) \right\}_{i=1}^N$

Maximization of conditional log likelihood  $\log p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)})$  over all samples:

$$\operatorname{argmax}_{\lambda} L(\lambda) = \sum_{i=1}^N \log p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}) \quad (5)$$

Objective function is:

$$L(\lambda) = \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^N \log Z(\mathbf{x}^{(i)}) - \sum_{k=1}^K \frac{\lambda_k^2}{2\sigma^2} \quad (6)$$

with  $\sum_{k=1}^K \frac{\lambda_k^2}{2\sigma^2}$  is regularizer (Gaussian prior) to avoid overfitting.

# Gradient

The first derivatives of  $L(\lambda)$  are:

$$\begin{aligned} \nabla_{\lambda} L(\lambda) = & \sum_{i=1}^N \sum_{t=1}^T f_k(y_{t-1}^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}) - \\ & \sum_{i=1}^N \sum_{t=1}^T \sum_{y, y'} f_k(y_{t-1}^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}) p(y, y' | \mathbf{x}^{(i)}) - \quad (7) \\ & \sum_{k=1}^K \frac{\lambda_k}{\sigma^2} \end{aligned}$$

Introduction

Linear chain CRFs

» Exponential model

» Parameter estimation

» Gradient

» Forward-Backward procedure

» Optimization

» Optimization — L-BFGS

» Summary

Questions

Appendix

# Forward-Backward procedure

Introduction

Linear chain CRFs

» Exponential model

» Parameter estimation

» Gradient

» Forward-Backward procedure

» Optimization

» Optimization — L-BFGS

» Summary

Questions

Appendix

$$p(y_t, y_{t-1} | \mathbf{x}) = \frac{\alpha_{t-1}(y_{t-1}) \Psi(y_t, y_{t-1}, \mathbf{x}) \beta_t(y_t)}{Z(\mathbf{x})} \quad (8)$$

with forward variables  $\alpha_t(j)$  :

$$\alpha_t(j) = \sum_{i \in \mathcal{S}} \Psi(j, i, \mathbf{x}) \alpha_{t-1}(i) \quad (9)$$

and backward variables  $\beta_t(i)$  :

$$\beta_t(i) = \sum_{j \in \mathcal{S}} \Psi(j, i, \mathbf{x}) \beta_{t+1}(j) \quad (10)$$

Normalization term  $Z(\mathbf{x})$  becomes

$$Z(\mathbf{x}) = \sum_y \alpha_T(y, T) = \sum_y \beta(y, 1) \quad (11)$$

# Optimization

Introduction

Linear chain CRFs

» Exponential model

» Parameter estimation

» Gradient

» Forward-Backward procedure

» Optimization

» Optimization — L-BFGS

» Summary

Questions

Appendix

Iterative scaling, gradient decent, conjugated gradients but second order methods have proved to be superior [5].

Second order Taylor expansion of  $L(\mathbf{x})$  as quadratic model of the objective function around  $\mathbf{x}$ :

$$L(\mathbf{x} + \Delta) \approx \Delta^T \nabla L(\mathbf{x}) + \frac{1}{2} \Delta^T \nabla^2 L(\mathbf{x}) \Delta \quad (12)$$

Newton method:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{H}_t^{-1} \nabla L(\mathbf{x}) \quad (13)$$

$\eta_t$  is computed via line search (is one for quadratic objective function).

$\mathbf{H} = \nabla^2 L(\mathbf{x})$ , Hesse matrix contains second derivatives, inversion required and is quadratic in memory. We don't like that :-(((

# Optimization — L-BFGS

Introduction

Linear chain CRFs

» Exponential model

» Parameter estimation

» Gradient

» Forward-Backward procedure

» Optimization

» Optimization — L-BFGS

» Summary

Questions

Appendix

But we are lucky. Iterative approximation of the inverse Hesse matrix is possible (BFGS-method). Big advantage: No inversion, no second derivatives - phew :-)

Let  $B_t$  be an approximation of  $H_t^{-1}$ , and you don't want to know why  $B_t$  is calculated as follows:

$$B_{t+1} = B_t + \frac{s_t s_t^T}{y_t^T s_t} \left( \frac{y_t^T B_t y_t}{y_t^T s_t} + 1 \right) - \frac{1}{y_t^T s_t} \left( s_t y_t^T B_t + B_t y_t s_t^T \right) \quad (14)$$

with  $y_t = \nabla L(x_t) - \nabla L(x_{t-1})$  and  $s_t = x_t - x_{t-1}$ .

Good, but still quadratic in memory. Limited memory BFGS (L-BFGS) is storing only a subset of all  $y_t$  and  $s_t$  [3]. I skip the details and I also don't talk about the line search required to determine  $\eta_t$ .

# Summary

Introduction

Linear chain CRFs

» Exponential model

» Parameter estimation

» Gradient

» Forward-Backward procedure

» Optimization

» Optimization — L-BFGS

» Summary

Questions

Appendix

So, what's so special about CRFs:

- Conditional likelihood  $p(\mathbf{y}|\mathbf{x})$ . Does not try to model  $p(\mathbf{x})$ .
- No Label-Bias problem as in MEMMs.
- Unimodal objective.
- No EM = no cheap unsupervised learning, but there are semi-supervised methods.
- Linear chain architecture allows efficient factorization and "fast" test and training.
- Weaker independence assumption than HMMs.
- Allows for long range features:  $f_i(y_{t-1}, y_t, \mathbf{x})$ .
- More general CRFs for other data structures such as trees, graphs, ... but much more expensive to train.

Introduction

Linear chain CRFs

Questions

Appendix

# Questions

Introduction

---

Linear chain CRFs

---

Questions

---

Appendix

- » Generative-discriminative pairs
- » Comparison HMM - CRF
- » HMMs - MEMMs - CRFs
- » Label Bias Problem
- » Software
- » References

# Appendix

# Generative-discriminative pairs

Introduction

Linear chain CRFs

Questions

Appendix

» Generative-discriminative pairs

» Comparison HMM - CRF

» HMMs - MEMMs - CRFs

» Label Bias Problem

» Software

» References

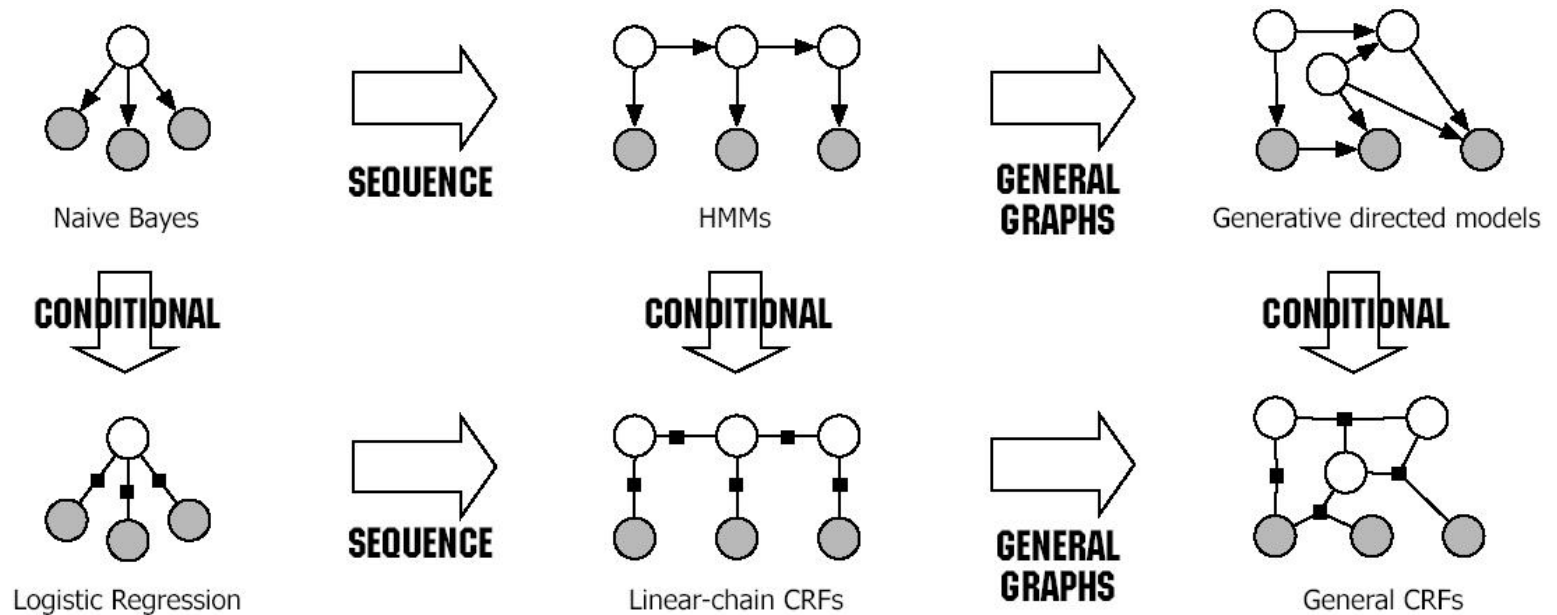


Figure 1: Related models and generative-discriminative pairs.

# Comparison HMM - CRF

Introduction

Linear chain CRFs

Questions

Appendix

» Generative-discriminative pairs

» Comparison HMM - CRF

» HMMs - MEMMs - CRFs

» Label Bias Problem

» Software

» References

- HMMs and CRFs both associate sequence of labels  $(y_1, \dots, y_n)$  to observations  $(x_1, \dots, x_n)$
- HMMs are Bayes nets and estimated by MLE.
- CRFs are MRFs and estimated by CMLE.
- HMM and CRFs have same complexity of decoding (Viterbi).
- Estimating an HMM from data is easy (MLE is relative frequency).
- Estimating a CRF from data is difficult (iterative numerical optimization).
- HMMs assume  $x_i$  are conditionally independent.
- CRFs do not assume that  $x_i$  are conditionally independent.
- CRFs better than HMMs when independence assumptions between  $X_i$  are invalid.
- It is easier to add new features to a CRF.
- There is no EM for CRFs.

# HMMs - MEMMs - CRFs

Introduction

Linear chain CRFs

Questions

Appendix

» Generative-discriminative

pairs

» Comparison HMM - CRF

» HMMs - MEMMs - CRFs

» Label Bias Problem

» Software

» References

HMM:

$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^T p(y_t|y_{t-1})p(x_t|y_t) \quad (15)$$

MEMM:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^T \frac{1}{Z(y_{t-1}, x_t)} \exp \left( \sum_i \lambda_i f_i(y_{t-1}, y_t) \right) \quad (16)$$

CRF:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \exp \left( \sum_i \lambda_i f_i(y_{t-1}, y_t) \right) \quad (17)$$

# Label Bias Problem

Problem: Per-state normalization (conservation of probability mass) can cause states to effectively ignore observations.

CRFs avoid label-bias problem through global normalization.

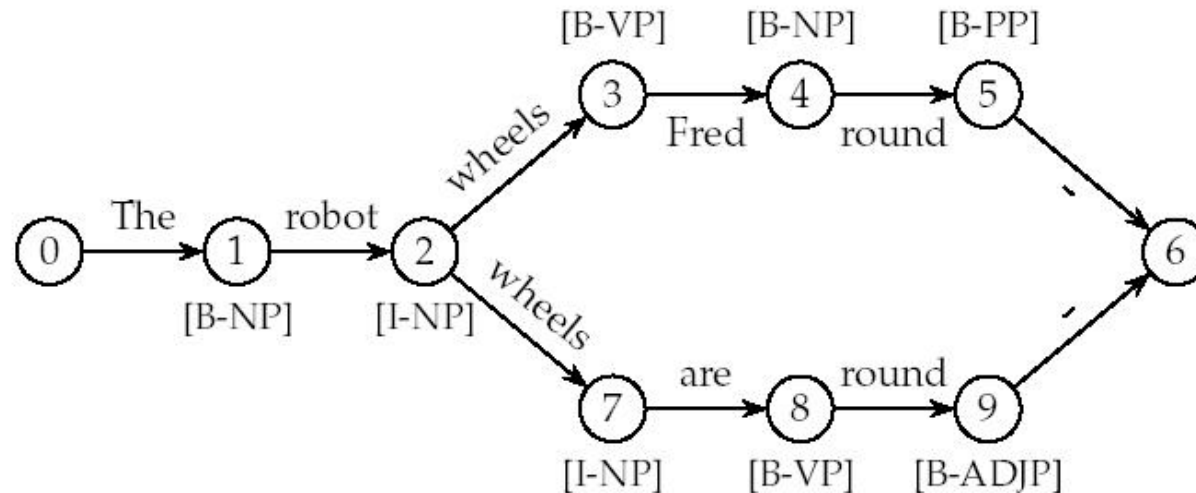


Figure 2: Label-Bias problem

# Software

Introduction

Linear chain CRFs

Questions

Appendix

» Generative-discriminative

pairs

» Comparison HMM - CRF

» HMMs - MEMMs - CRFs

» Label Bias Problem

» Software

» References

- CRF at Sourceforge, <http://crf.sourceforge.net/>  
(Sunita Sarawagi, Java, monolithic code, usable but hard to change)
- Mallet, [http://mallet.cs.umass.edu/index.php/Main\\_Page](http://mallet.cs.umass.edu/index.php/Main_Page)  
(Andrew McCallum et. al., Java, big lib, text processing oriented)
- FlexCrf,  
<http://www.jaist.ac.jp/~hieuxuan/flexcrfs/flexcrfs.html>  
(Xuan-Hieu Phan and Le-Minh Nguyen, C++, nice code with documentation, parallel version available)
- CRF++ <http://www.chasen.org/~taku/software/CRF++>  
(C++, small, text processing oriented)
- MatLab  
<http://www.cs.ubc.ca/~murphyk/Software/CRF/crf.html>  
(no longer available, new version in preparation)

# References

Introduction

Linear chain CRFs

Questions

Appendix

» Generative-discriminative pairs

» Comparison HMM - CRF

» HMMs - MEMMs - CRFs

» Label Bias Problem

» Software

» References

- [1] Rahul Gupta. Conditional random fields. 2005.
- [2] John Lafferty, Andrew McCallum, and Fernando 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.
- [3] D.C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528, 1989.
- [4] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.
- [5] Hanna Wallach. Efficient training of conditional random fields. Master's thesis, Division of Informatics, University of Edinburgh, 2002.