Dynamic Conditional Random Fields

Factorized Probabilistic Models for Labeling and Segmenting Data

Martin Zimmer Kristensen

October 29, 2016

Outline

Introduction

- Sequential Data
- Generative Versus Discriminative
- Conditional Random Fields

Sequential Data

Part-of-speech Tagging

The [DT] little [JJ] dog [NN] was [VBD] furious [JJ] and [CC] barked [VBD] at [IN] the [DT] large [JJ] human [NN]

Noun-phrase Chunking

The [B-NP] little [I-NP] dog [I-NP] was [O] furious [O] and [O] barked [O] at [O] the [B-NP] large [I-NP] human [I-NP]

Sequential Data

Other

- Named Entity Recognition
- Speech Recognition

Generative Versus Discriminative

Generative Models:

- The joint probability $p(x, y)$
  - Able to generate x
  - Assumptions to achieve tractability:
    - Naive Bayes assumption
  - Modeling interdependent features is difficult

Generative Versus Discriminative

Discriminative Models:

- The conditional probability: $p(y|x)$
  - Assumptions among y
  - Assumptions among y and x
  - Interdependent features
    - Capitalization, prefixes, suffixes, neighboring words...
  - Unseen words can be labeled by using their features

Conditional Random Fields (CRF)

Definition (CRF)

- Let G be an undirected model over sets of random variables y and x
- Let $C = \{\{y_c, x_c\}\}$ be the set of cliques in G
- Conditional probability defined as:

$$p_\Lambda(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \Phi(y_c, x_c)$$

- Φ is a potential function
- $Z(x)$ is a normalization factor

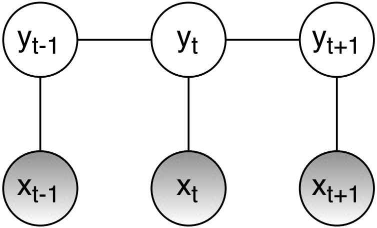
Potential function

Feature Functions

- Potentials factorize according to a set of features functions $\{f_k\}$:

$$f(y_c, x_c) = exp\left(\sum_k \lambda_k f_k(y_c, x_c)\right)$$

Linear-chain CRF



- A special case of CRFs where the first-order Markov assumption is made over the latent variables.
- Then the feature functions can be described as:

$$f_k(y_{t-1}, y_t, x, t)$$

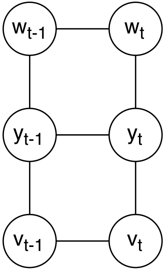
Linear-chain CRF

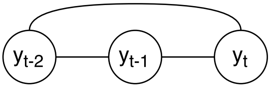
Feature Functions:

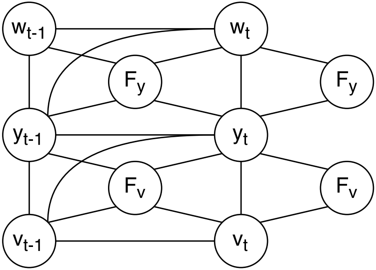
- $f(y_{t-1}, y_t, x, t) = 1$:
  - *iff* $y_{t-1} = adjective$, $y_t = proper\ noun$, and x_t begins with a capital letter.
- $f(y_{t-1}, y_t, x, t) = 1$:
  - *iff* $y_t = organization$, $x_t =$ *"New"*, $x_{t+1} =$ *"York"*, and $x_{t+2} =$ *"Times"*

Key Contributions

- Dynamic Conditional Random Fields (DCRF)
  - Factorial CRF
  - Exact inference for some models
  - Inference approximation:
    - Lower training time
    - Equal performance







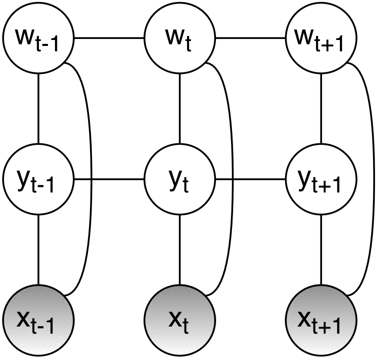
DCRF

Definition (Dynamic Conditional Random Field)

- Cliques are defined by its index j and its time offset $\Delta t = i - t$
  - I.e. y_{12} when $t = 1$ is denoted as y_{02} since $\Delta t = 0$
- $p(y|x) = \frac{1}{Z(x)} \prod_{t} \prod_{c \in C} \exp\left(\sum_{k} \lambda_k f_k(y_{t,c}, x, t) \right)$
- where $Z(x)$ is the partition function

Factorial CRF

- A DCRF which has linear chains of labels, with connection between cotemporal labels.



Factorial CRF

Cliques

- The cliques are of the form:
  - Within-chain edges: $\{(0, \ell), (1, \ell)\}$
  - Between-chain edges: $\{(0, \ell), (0, \ell + 1)\}$

Factorial CRF

Definition (Factorial CRF)

$$p(x|y) = \frac{1}{Z(x)} \left(\prod_{t=1}^{T-1} \prod_{\ell=1}^{L} \Phi_\ell(y_{\ell,t}, y_{\ell,t+1}, x, t) \right) \left(\prod_{t=1}^{T} \prod_{\ell=1}^{L-1} \Psi_\ell(y_{\ell,t}, y_{\ell+1,t}, x, t) \right)$$

- $\{\Phi_\ell\}$ are the factors over within-chain edges
- $\{\Psi_\ell\}$ are the factors over between-chain edges
- $Z(x)$ is the partition function.

Factorial CRF

Factors

- The factors are modeled using features $\{f_k\}$ and weights $\{\lambda_k\}$ of G as:

$$\Phi_\ell(y_{\ell,t}, y_{\ell,t+1}, x, t) = \exp\left\{ \sum_k \lambda_k f_k(y_{\ell,t}, y_{\ell,t+1}, x, t) \right\},$$

$$\Psi_\ell(y_{\ell,t}, y_{\ell+1,t}, x, t) = \exp\left\{ \sum_k \lambda_k f_k(y_{\ell,t}, y_{\ell+1,t}, x, t) \right\}.$$

Inference

- Exact inference can be expensive for many models
- Use approximate inference using loopy belief propagation

Inference

Loopy Belief Propagation

- Message from node x_u to node x_v:

$$m_{x_u}(x_v)$$

- Value of $m_{x_u}(x_v)$:
  - The belief of x_u about the probability $p(x_j)$
- Iteratively send messages until convergence
- Different schedules can be applied
  - Random
  - Tree-based (send messages from leaves to root and back)

Parameter Estimation

- Given training data $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$
  - Finding a set of parameters $\Lambda = \{\lambda_k\}$
- Assign weights λ_k such that we are accurate on the training data.

Experiments

Noun-phrase Chunking

The [B-NP] little [I-NP] dog [I-NP] was [O] furious [O] and [O] barked [O] at [O] the [B-NP] large [I-NP] human [I-NP]

Usual approach:

1. POS tagging
2. Noun-phrase Chunking

Challenge:

- Mistakes in POS tagging will cascade onto noun-phrase chunking

Experiments

Data:

- CoNLL 2000

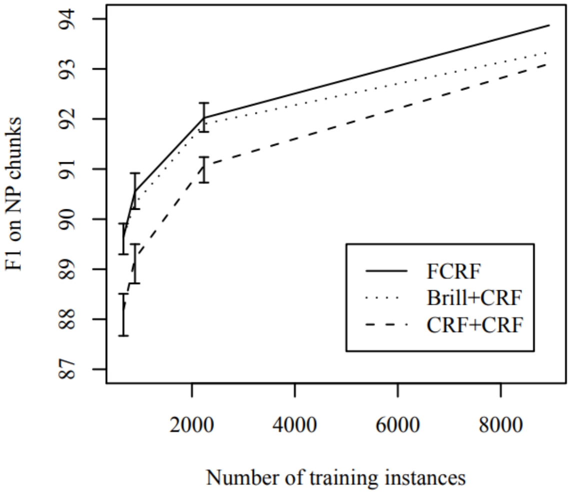
Approach:

- Use a factorial CRF to jointly do POS and chunking

Compare to:

- CRF+CRF
- Brill+CRF
  - Brill tagger trained on over four times more data including the CoNLL 2000

Results


F1 on NP chunks
87
88
89
90
91
92
93
94
2000
4000
6000
8000
Number of training instances
FCRF
Brill+CRF
CRF+CRF

Results

	Size	CRF+CRF	Brill+CRF	FCRF
POS accuracy	223	86.23	N/A	**93.12**
	447	90.44		**95.43**
	670	92.33		**96.34**
	894	93.56		**96.85**
	2234	96.18		**97.87**
	8936	98.28		**98.92**
NP accuracy	223	92.67	93.75	**93.87**
	447	94.09	94.91	**95.03**
	670	94.72	95.46	95.46
	894	95.17	95.75	**95.86**
	2234	96.08	96.38	**96.51**
	8936	96.98	97.09	**97.36**
Joint accuracy	223	81.92	N/A	**89.19**
	447	86.58		**91.85**
	670	88.68		**92.86**
	894	90.06		**93.60**
	2234	93.00		**94.90**
	8936	95.56		**96.48**
NP F1	223	83.84	86.02	**86.03**
	447	86.87	88.56	**88.59**
	670	88.19	**89.65**	89.64
	894	89.21	90.31	**90.55**
	2234	91.07	91.90	**92.02**
	8936	93.10	93.33	**93.87**

Inference Algorithms

Method	Time (hr)		NP F1		LBFGS iter
	μ	s	μ	s	μ
Random (3)	15.67	2.90	88.57	0.54	63.6
Tree (3)	13.85	11.6	88.02	0.55	32.6
Tree (∞)	13.57	3.03	88.67	0.57	65.8
Random (∞)	13.25	1.51	88.60	0.53	76.0
Exact	20.49	1.97	88.63	0.53	73.6

Conclusions

- Factorial CRFs are useful for NP tasks
- Loopy belief propagation:
  - Performs equally to exact inference
  - Reduces training time