CS 6784 Paper Presentation

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

John Lafferty, Andrew McCallum, Fernando C. N. Pereira

Presenters: Brad Gulko and Stephanie Hyland

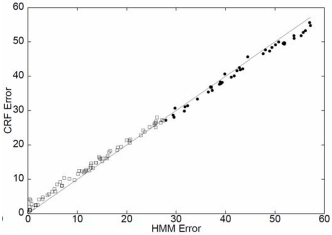
February 20, 2014

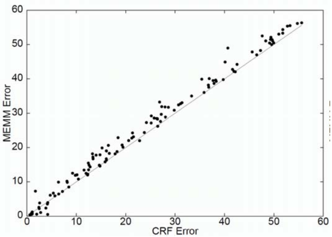
Main Contribution Summary

- This 2001 paper introduced the **C**onditional **R**andom **F**ield (CRF).
- Describes efficient representation of field potentials in terms of features.
- Provides two algorithms for finding Maximum Likelihood parameter values.
- Provides some really unconvincing examples...

Main Contribution Summary

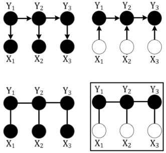
... examples are NOT the strongest point of this paper





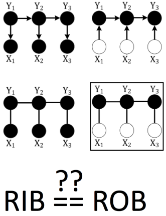
Talk Structure

- Brad
  - CRF in context



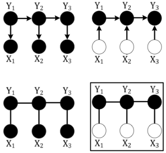
Talk Structure

- Brad
  - CRF in context
  - The Label Bias Problem



Talk Structure

- Brad
  - CRF in context
  - The Label Bias Problem
- Stephanie
  - Parameter Estimation



RIB $\overset{??}{==}$ ROB

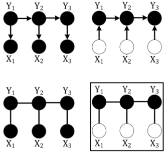
$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left(\sum_{e\in\mathcal{E},k}\lambda_k f_k(e,\mathbf{y}|_e,\mathbf{x}) + \sum_{v\in\mathcal{V},k}\mu_k g_k(v,\mathbf{y}|_v,\mathbf{x})\right)$$

$$\theta = (\lambda_1, \lambda_2, \cdots ; \mu_1, \mu_2, \cdots)$$

$$\theta_{t+1} = \theta_t - [Hf(\theta_t)]^{-1}\nabla f(\theta_t)$$

Talk Structure

- Brad
  - CRF in context
  - The Label Bias Problem
- Stephanie
  - Parameter Estimation
  - Experiments
  - Conclusion

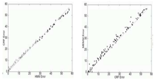


RIB ?? == ROB

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{e \in \mathcal{E}, k} \lambda_k f_k(e, \mathbf{y}|_e, \mathbf{x}) + \sum_{v \in \mathcal{V}, k} \mu_k g_k(v, \mathbf{y}|_v, \mathbf{x}) \right)$$

$$\theta = (\lambda_1, \lambda_2, \cdots ; \mu_1, \mu_2, \cdots)$$

$$\theta_{t+1} = \theta_t - [Hf(\theta_t)]^{-1} \nabla f(\theta_t)$$



CRF in Context

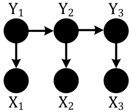
- $\mathbf{X} = \{X_1, X_2, \cdots\}$ be a set of observed RV
- $\mathbf{Y} = \{Y_1, Y_2, \cdots\}$ be a set of label RV
- X, Y be a set of joint observations of $\mathbf{X}, \mathbf{Y}$

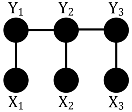
	Generative	Discriminative
Directed	???	???
Undirected	???	???

CRF in Context

- $\mathbf{X} = \{X_1, X_2, \cdots\}$ be a set of observed RV
- $\mathbf{Y} = \{Y_1, Y_2, \cdots\}$ be a set of label RV
- X, Y be a set of joint observations of $\mathbf{X}, \mathbf{Y}$

	Generative	Discriminative
Directed	HMM	???
Undirected	MRF	???

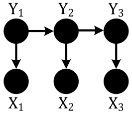


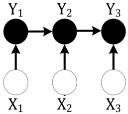


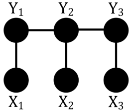
CRF in Context

- $\mathbf{X} = \{X_1, X_2, \cdots\}$ be a set of observed RV
- $\mathbf{Y} = \{Y_1, Y_2, \cdots\}$ be a set of label RV
- X, Y be a set of joint observations of $\mathbf{X}, \mathbf{Y}$

	Generative	Discriminative
Directed	HMM	ME-MM
Undirected	MRF	**???**





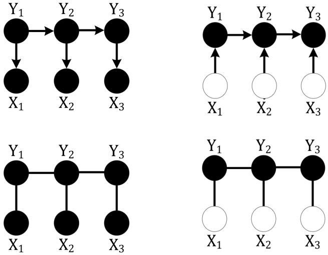


In 2001, HMM, ME-MM and MRF were well known,

CRF in Context

- $\mathbf{X} = \{X_1, X_2, \cdots\}$ be a set of observed RV
- $\mathbf{Y} = \{Y_1, Y_2, \cdots\}$ be a set of label RV
- X, Y be a set of joint observations of $\mathbf{X}, \mathbf{Y}$

	Generative	Discriminative
Directed	HMM	ME-MM
Undirected	MRF	**CRF**



In 2001, HMM, ME-MM and MRF were well known,the paper presents the CRF.

Generative vs. Discriminative

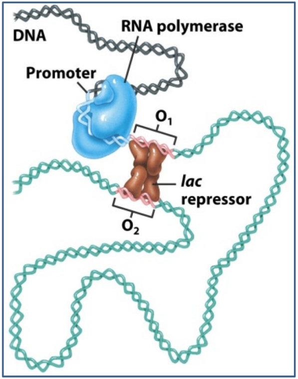
- Generative: maximise joint $P(Y, X) = P(Y|X)P(X)$
- Discriminative: maximise conditional $P(Y|X)$
- When is Discriminative helpful?
  - Tractability requires independence

Generative vs. Discriminative

- Generative: maximise joint $P(Y, X) = P(Y|X)P(X)$
- Discriminative: maximise conditional $P(Y|X)$
- When is Discriminative helpful?
  - Tractability requires independence
  - ...but sometimes there are important correlations in X.

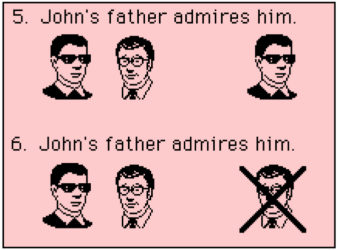
Examples: important correlations

- Long range interactions in human genomics



Examples: important correlations

- Long range interactions in human genomics
- Pronoun definition and binding

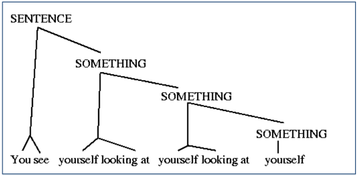


Examples: important correlations

- Long range interactions in human genomics
- Pronoun definition and binding
- Context in whole scene image recognition

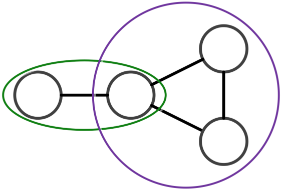
Examples: important correlations

- Long range interactions in human genomics
- Pronoun definition and binding
- Context in whole scene image recognition
- Recursive structure in language



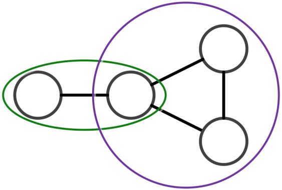
Directed vs. Undirected

- For a graphical model $\mathbf{G}(\mathbf{E}, \mathbf{V})$ with joint potential $\Psi(\mathbf{V})$.
- Let $\mathbf{C}$ be the set of **cliques** (fully connected subgroups) in $\mathbf{G}$, with $c \in \mathcal{C}$ having edges $\mathbf{E}_c$ and vertices $\mathbf{V}_c$.



Directed vs. Undirected

- For a graphical model $\mathbf{G}(\mathbf{E}, \mathbf{V})$ with joint potential $\Psi(\mathbf{V})$.
- Let $\mathbf{C}$ be the set of **cliques** (fully connected subgroups) in $\mathbf{G}$, with $c \in \mathcal{C}$ having edges $\mathbf{E}_c$ and vertices $\mathbf{V}_c$.
- Finally, $Dom(\mathbf{V})$ is the set of all values assumable by the random variables, $\mathbf{V}(= \mathbf{X} \cup \mathbf{Y})$.



$$P(\mathbf{V}) = \frac{1}{Z}\Psi(\mathbf{V}), \qquad Z = \sum_{vinDom(\mathbf{V})} \Psi(v)$$

Directed vs. Undirected, continued

- Compactness requires factorization (Hammersley-Clifford, 1971):

$$\Psi(\mathbf{V}) = \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{V}_c)$$

Directed vs. Undirected, continued

- Compactness requires factorization (Hammersley-Clifford, 1971):

$$\Psi(\mathbf{V}) = \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{V}_c)$$

- Directed: local Normalization -

$$\forall c \in \mathcal{C}, \quad \sum_{v \in Dom(\mathbf{V}_c)} \Psi_c(v) = 1$$

Directed vs. Undirected, continued

- Compactness requires factorization (Hammersley-Clifford, 1971):

$$\Psi(\mathbf{V}) = \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{V}_c)$$

- Directed: local Normalization - each Ψ_c is a *probability*.

$$\forall c \in \mathcal{C}, \quad \sum_{v \in Dom(\mathbf{V}_c)} \Psi_c(v) = 1$$

Directed vs. Undirected, continued

- Compactness requires factorization (Hammersley-Clifford, 1971):

$$\Psi(\mathbf{V}) = \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{V}_c)$$

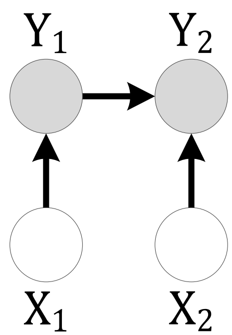
- Directed: local Normalization - each Ψ_c is a *probability*.

$$\forall c \in \mathcal{C}, \quad \sum_{v \in Dom(\mathbf{V}_c)} \Psi_c(v) = 1$$

- Undirected: Global Normalization - relaxes this constraint... but what does it buy us?

The Label Bias Problem: Conditional Markov Model (EM-MM)

Toy Problem – fragment of a ME-MM



$Y_1 \in \{1,2\}$ $Y_2 \in \{3,4\}$

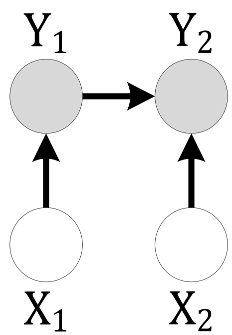
Training Data:

8: $X = \{RI\}$ $Y = \{13\}$

2: $X = \{RO\}$ $Y = \{24\}$

The Label Bias Problem: Conditional Markov Model (EM-MM)

Toy Problem – fragment of a ME-MM



$Y_1 \in \{1,2\}$ $Y_2 \in \{3,4\}$

Training Data:

8: $X = \{RI\}$ $Y = \{13\}$

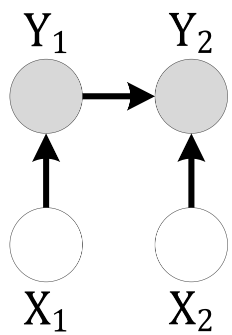
2: $X = \{RO\}$ $Y = \{24\}$

$P(Y_1 \mid X_1)$	Y_1 = 1	Y_1 = 2
$X_1 = R$	0.8	0.2

Rel. Joint $\Psi(Y_2, X_2, Y_1)$		Y_2 = 3	Y_2 = 4
$Y_1 = 1$	$X_2 = I$	**8**	ε
	$X_2 = O$	ε	ε
$Y_1 = 2$	$X_2 = I$	ε	ε
	$X_2 = O$	ε	**2**

The Label Bias Problem: Conditional Markov Model (EM-MM)

Toy Problem – fragment of a ME-MM



$Y_1 \in \{1,2\}\ Y_2 \in \{3,4\}$

Training Data:

$8: X = \{RI\}\ \ Y = \{13\}$

$2: X = \{RO\}\ Y = \{24\}$

$P(Y_1 \mid X_1)$	Y_1: 1	Y_1: 2
$X_1 = R$	0.8	0.2

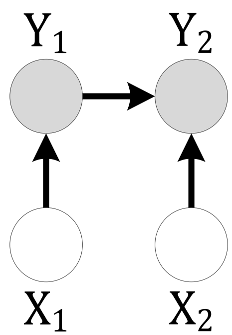
Rel. Joint $\Psi(Y_2, X_2, Y_1)$		Y_2: 3	Y_2: 4
$Y_1=1$	$X_2=I$	8	ε
$Y_1=1$	$X_2=O$	ε	ε
$Y_1=2$	$X_2=I$	ε	ε
$Y_1=2$	$X_2=O$	ε	2

→

Conditional $P(Y_2 \mid X_2, Y_1)$		Y_2: 3	Y_2: 4
$Y_1=1$	$X_2=I$	1-ε	ε
$Y_1=1$	$X_2=O$	0.5	0.5
$Y_1=2$	$X_2=I$	0.5	0.5
$Y_1=2$	$X_2=O$	ε	1-ε

The Label Bias Problem: Conditional Markov Model (EM-MM)

Toy Problem – fragment of a ME-MM



$Y_1 \in \{1,2\}\ Y_2 \in \{3,4\}$

Training Data:

8: $X = \{RI\}\ \ Y = \{13\}$

2: $X = \{RO\}\ Y = \{24\}$

$P(Y_1 \mid X_1)$	Y_1 = 1	Y_1 = 2
$X_1 = R$	0.8	0.2

Rel. Joint $\Psi(Y_2, X_2, Y_1)$		Y_2 = 3	Y_2 = 4
$Y_1=1$	$X_2=I$	8	ε
	$X_2=O$	ε	ε
$Y_1=2$	$X_2=I$	ε	ε
	$X_2=O$	ε	2

Conditional $P(Y_2 \mid X_2, Y_1)$		Y_2 = 3	Y_2 = 4
$Y_1=1$	$X_2=I$	1-ε	ε
	$X_2=O$	0.5	0.5
$Y_1=2$	$X_2=I$	0.5	0.5
	$X_2=O$	ε	1-ε

Viterbi is $P(Y_1, Y_2 \mid X)$

$= P(Y_2 \mid Y_1, X) P(Y_1 \mid X)$

$= P(Y_1 \mid X_1) P(Y_2 \mid Y_1, X_2)$

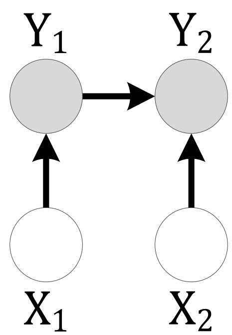
Lets try it for $X = \{RO\}$

Y_1, Y_2	$P(Y_1 \mid R)$	$P(Y_2 \mid Y_1, O)$	$P(Y_1, Y_2 \mid RO)$
1,3	0.8	0.5	
1,4	0.8	0.5	
2,3	0.2	ε	
2,4	0.2	1-ε	

Which labeling wins?

The Label Bias Problem: Conditional Markov Model (EM-MM)

Toy Problem – fragment of a ME-MM



$Y_1 \in \{1,2\}\ Y_2 \in \{3,4\}$

Training Data:

8: $X = \{RI\}\ \ Y = \{13\}$

2: $X = \{RO\}\ Y = \{24\}$

$P(Y_1 \mid X_1)$	Y_1	
	1	2
$X_1 = R$	0.8	0.2

Rel. Joint $\Psi(Y_2, X_2, Y_1)$		Y_2	
		3	4
$Y_1 = 1$	$X_2 = I$	8	ε
	$X_2 = O$	ε	ε
$Y_1 = 2$	$X_2 = I$	ε	ε
	$X_2 = O$	ε	2

Conditional $P(Y_2 \mid X_2, Y_1)$		Y_2	
		3	4
$Y_1 = 1$	$X_2 = I$	1-ε	ε
	$X_2 = O$	0.5	0.5
$Y_1 = 2$	$X_2 = I$	0.5	0.5
	$X_2 = O$	ε	1-ε

Viterbi is $P(Y_1, Y_2 \mid X)$

$= P(Y_2 \mid Y_1, X) P(Y_1 \mid X)$

$= P(Y_1 \mid X_1) P(Y_2 \mid Y_1, X_2)$

Lets try it for $X = \{RO\}$

Y_1, Y_2	$P(Y_1 \mid R)$	$P(Y_2 \mid Y_1, O)$	$P(Y_1, Y_2 \mid RO)$
1,3	0.8	0.5	0.4
1,4	0.8	0.5	0.4
2,3	0.2	ε	ε
2,4	0.2	1-ε	0.2

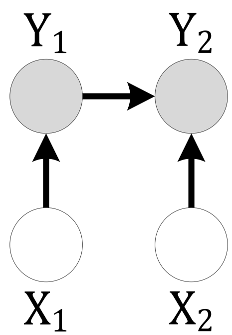
But we want

$Y = \{2,4\}$

What happened?

The Label Bias Problem: Conditional Markov Model (EM-MM)

Toy Problem – fragment of a ME-MM



$Y_1 \in \{1,2\}$ $Y_2 \in \{3,4\}$

Training Data:

8: $X = \{RI\}$ $Y = \{13\}$

2: $X = \{RO\}$ $Y = \{24\}$

Local Normalization requires a probability…. So..

$$\frac{\epsilon}{2\epsilon} \Rightarrow \frac{\mathbf{1}}{\mathbf{2}}$$

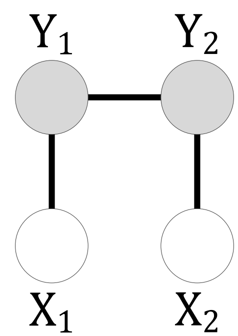
Rel. Joint $\Psi(Y_2,X_2,Y_1)$		Y_2 = 3	Y_2 = 4
$Y_1=1$	$X_2=I$	8	ε
$Y_1=1$	$X_2=O$	ε	ε
$Y_1=2$	$X_2=I$	ε	ε
$Y_1=2$	$X_2=O$	ε	2

Conditional $P(Y_2 \vert X_2,Y_1)$		Y_2 = 3	Y_2 = 4
$Y_1=1$	$X_2=I$	1-ε	ε
$Y_1=1$	$X_2=O$	0.5	0.5
$Y_1=2$	$X_2=I$	0.5	0.5
$Y_1=2$	$X_2=O$	ε	1-ε

Y_1,Y_2	$P(Y_1 \vert R)$	$P(Y_2 \vert Y_1,O)$	$P(Y_1,Y_2 \vert RO)$
1,3	0.8	0.5	0.4
1,4	0.8	0.5	0.4
2,3	0.2	ε	ε
2,4	0.2	1-ε	0.2

The Label Bias Problem: Potentials

Toy Problem – fragment of a CRF



$Y_1 \in \{1,2\}\ Y_2 \in \{3,4\}$

Training Data:

8: $X = \{RI\}\ \ Y = \{13\}$

2: $X = \{RO\}\ Y = \{24\}$

$$\Psi(X, Y_1, Y_2) = \Psi(X_1, Y_1)\Psi(Y_1, Y_2)\Psi(X_2, Y_2)$$

The Label Bias Problem: Potentials

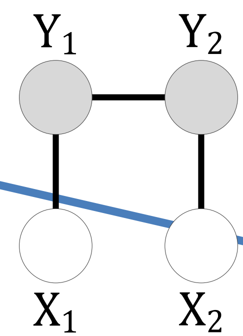
Toy Problem – fragment of a CRF

Ψ		Y_1 = 1	Y_1 = 2
X_1	R	8	2
	-	ε	ε

Ψ		Y_2 = 3	Y_2 = 4
Y_1	1	8	ε
	2	ε	2

Ψ		Y_2 = 3	Y_2 = 4
X_2	I	8	ε
	O	ε	2

$$\Psi(X, Y_1, Y_2) = \Psi(X_1, Y_1)\Psi(Y_1, Y_2)\Psi(X_2, Y_2)$$



$Y_1 \in \{1,2\}$ $Y_2 \in \{3,4\}$

Training Data:

8: $X = \{RI\}$ $Y = \{13\}$

2: $X = \{RO\}$ $Y = \{24\}$

The Label Bias Problem: Potentials

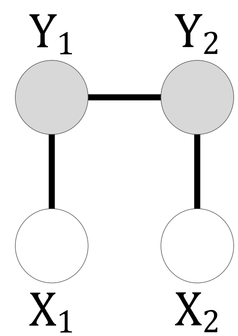
Toy Problem – fragment of a CRF

Ψ		Y_1 = 1	Y_1 = 2
X_1	R	8	2
	-	ε	ε

Ψ		Y_2 = 3	Y_2 = 4
Y_1	1	8	ε
	2	ε	2

Ψ		Y_2 = 3	Y_2 = 4
X_2	I	8	ε
	O	ε	2

$$\Psi(X, Y_1, Y_2) = \Psi(X_1, Y_1)\Psi(Y_1, Y_2)\Psi(X_2, Y_2)$$



$Y_1 \in \{1,2\}$ $Y_2 \in \{3,4\}$

Training Data:

8: $X = \{RI\}$ $Y = \{13\}$

2: $X = \{RO\}$ $Y = \{24\}$

Y_1,Y_2	$\Psi(X_1,Y_1)$	$\Psi(Y_1,Y_2)$	$\Psi(X_2,Y_2)$	$\Psi(Y_1,Y_2,X=RO)$	$P(Y_1,Y_2 \mid X)$
1,3	8	8	ε		
1,4	8	ε	2		
2,3	2	ε	ε		
2,4	2	2	2		

Which labeling wins, now?

The Label Bias Problem: Potentials

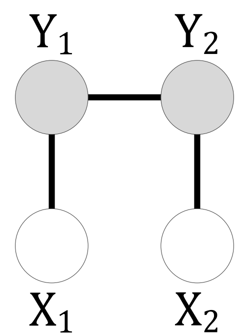
Toy Problem – fragment of a CRF

Ψ		Y_1	
		1	2
X_1	R	8	2
	-	ε	ε

Ψ		Y_2	
		3	4
Y_1	1	8	ε
	2	ε	2

Ψ		Y_2	
		3	4
X_2	I	8	ε
	O	ε	2

$$\Psi(X, Y_1, Y_2) = \Psi(X_1, Y_1)\Psi(Y_1, Y_2)\Psi(X_2, Y_2)$$



$Y_1 \in \{1,2\}$ $Y_2 \in \{3,4\}$

Training Data:
8: $X = \{RI\}$ $Y = \{13\}$
2: $X = \{RO\}$ $Y = \{24\}$

Y_1,Y_2	$\Psi(X_1,Y_1)$	$\Psi(Y_1,Y_2)$	$\Psi(X_2,Y_2)$	$\Psi(Y_1,Y_2,X=RO)$	$P(Y_1,Y_2 \mid X)$
1,3	8	8	ε	64ε	small
1,4	8	ε	2	16ε	tiny
2,3	2	ε	ε	$2\varepsilon^2$	infinitesimal
2,4	2	2	2	8	~100%

Because potentials do not have to normalize into probabilities until AFTER aggregation, they don't suffer from inappropriate conditioning.

Fun fact: We have seen this in class before!

- Graphical model $\mathbf{G}(\mathbf{E}, \mathbf{V})$ with joint potential $\Psi(\mathbf{V})$, $\mathcal{C}$ the set of *cliques* in $\mathbf{G}$ with $c \in \mathcal{C}$ having edges $\mathbf{E}_c$ and vertices $\mathbf{V}_c$

$$P(\mathbf{V}) \propto \Psi(\mathbf{V}) = \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{V}_c)$$

Fun fact: We have seen this in class before!

- Graphical model $\mathbf{G}(\mathbf{E}, \mathbf{V})$ with joint potential $\Psi(\mathbf{V})$, $\mathcal{C}$ the set of *cliques* in $\mathbf{G}$ with $c \in \mathcal{C}$ having edges $\mathbf{E}_c$ and vertices $\mathbf{V}_c$

$$P(\mathbf{V}) \propto \Psi(\mathbf{V}) = \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{V}_c)$$

- $\mathbf{M}^3$ **nets**: cliques are pairs, and all conditioned on observed $\mathbf{x}$:

$$P(\mathbf{y}|\mathbf{x}) \propto \prod_{(i,j) \in \mathbf{E}} \psi_{ij}(y_i, y_j, x)$$

Fun fact: We have seen this in class before!

- Graphical model $\mathbf{G}(\mathbf{E}, \mathbf{V})$ with joint potential $\Psi(\mathbf{V})$, $\mathcal{C}$ the set of *cliques* in $\mathbf{G}$ with $c \in \mathcal{C}$ having edges $\mathbf{E}_c$ and vertices $\mathbf{V}_c$

$$P(\mathbf{V}) \propto \Psi(\mathbf{V}) = \prod_{c \in \mathcal{C}} \Psi_c(\mathbf{V}_c)$$

- $\mathbf{M}^3$ **nets**: cliques are pairs, and all conditioned on observed $\mathbf{x}$:

$$P(\mathbf{y}|\mathbf{x}) \propto \prod_{(i,j) \in \mathbf{E}} \psi_{ij}(y_i, y_j, x)$$

- **AMN**: cliques are pairs of nodes and singletons:

$$P_\phi(y) = \frac{1}{Z} \prod_{i}^{N} \phi_i(y_i) \prod_{i,j \in \mathbf{E}} \phi_{i,j}(y_i, y_j)$$

Where do Parameters Come From?

CRF's are part of the same general class, $P(\boldsymbol{V}) \propto \Psi(\boldsymbol{V}) = \prod_{\boldsymbol{c} \in \boldsymbol{C}} \Psi_c(\boldsymbol{V}_{\boldsymbol{c}})$

Where do Parameters Come From?

CRF's are part of the same general class, $P(\boldsymbol{V}) \propto \Psi(\boldsymbol{V}) = \prod_{\boldsymbol{c} \in \boldsymbol{C}} \Psi_c(\boldsymbol{V_c})$

For trees, cliques are pairs of vertices sharing an edge ($\boldsymbol{y}|_e$), and single vertices ($\boldsymbol{y}|_v$):

$$\Psi(\boldsymbol{V}) = \prod_{\boldsymbol{e} \in \boldsymbol{Edge}} \Psi_e\left(\boldsymbol{y}\Big|_{\boldsymbol{e}}\right) \prod_{\boldsymbol{v} \in \boldsymbol{V}} \Psi_v\left(\boldsymbol{y}\Big|_{\boldsymbol{v}}\right)$$

Where do Parameters Come From?

CRF's are part of the same general class, $P(\boldsymbol{V}) \propto \Psi(\boldsymbol{V}) = \prod_{\boldsymbol{c} \in \boldsymbol{C}} \Psi_c(\boldsymbol{V_c})$

For trees, cliques are pairs of vertices sharing an edge ($\boldsymbol{y}|_e$), and single vertices ($\boldsymbol{y}|_v$):

$$\Psi(\boldsymbol{V}) = \prod_{\boldsymbol{e} \in \boldsymbol{Edge}} \Psi_e\left(\boldsymbol{y}\Big|_{\boldsymbol{e}}\right) \prod_{\boldsymbol{v} \in \boldsymbol{V}} \Psi_v\left(\boldsymbol{y}\Big|_{\boldsymbol{v}}\right)$$

And because this is a <u>conditional</u> network with $\boldsymbol{V} = \boldsymbol{X} \cup \boldsymbol{Y}$

$$\Psi(\boldsymbol{Y}|X) = \prod_{\boldsymbol{e} \in \boldsymbol{Edge}} \Psi_e\left(\boldsymbol{y}\Big|_{\boldsymbol{e}}, x\right) \prod_{\boldsymbol{v} \in \boldsymbol{V}} \Psi_v\left(\boldsymbol{y}\Big|_{\boldsymbol{v}}, x\right)$$

Where do Parameters Come From?

CRF's are part of the same general class, $P(\boldsymbol{V}) \propto \Psi(\boldsymbol{V}) = \prod_{\boldsymbol{c} \in \boldsymbol{C}} \Psi_c(\boldsymbol{V_c})$

For trees, cliques are pairs of vertices sharing an edge ($\boldsymbol{y}|_e$), and single vertices ($\boldsymbol{y}|_v$):

$$\Psi(\boldsymbol{V}) = \prod_{\boldsymbol{e} \in \boldsymbol{Edge}} \Psi_e\left(\boldsymbol{y}\big|_{\boldsymbol{e}}\right) \prod_{\boldsymbol{v} \in \boldsymbol{V}} \Psi_v\left(\boldsymbol{y}\big|_{\boldsymbol{v}}\right)$$

And because this is a <u>conditional</u> network with $\boldsymbol{V} = \boldsymbol{X} \cup \boldsymbol{Y}$

$$\Psi(\boldsymbol{Y}|X) = \prod_{\boldsymbol{e} \in \boldsymbol{Edge}} \Psi_e\left(\boldsymbol{y}\big|_{\boldsymbol{e}}, x\right) \prod_{\boldsymbol{v} \in \boldsymbol{V}} \Psi_v\left(\boldsymbol{y}\big|_{\boldsymbol{v}}, x\right)$$

An exponential identity gives us

$$\Psi(\boldsymbol{Y}|X) = \exp\left(\sum_{e \in \mathrm{E}} \log\left(\Psi_e\left(\boldsymbol{y}\big|_{\boldsymbol{e}}, x\right)\right) + \sum_{\boldsymbol{v} \in \boldsymbol{V}} \log\left(\Psi_v\left(\boldsymbol{y}\big|_{\boldsymbol{v}}, x\right)\right)\right)$$

Where do Parameters Come From?

CRF's are part of the same general class, $P(\boldsymbol{V}) \propto \Psi(\boldsymbol{V}) = \prod_{\boldsymbol{c}\in\boldsymbol{C}} \Psi_c(\boldsymbol{V_c})$

For trees, cliques are pairs of vertices sharing an edge ($\boldsymbol{y}|_e$), and single vertices ($\boldsymbol{y}|_v$):

$$\Psi(\boldsymbol{V}) = \prod_{\boldsymbol{e}\in \boldsymbol{Edge}} \Psi_e\left(\boldsymbol{y}\big|_{\boldsymbol{e}}\right) \prod_{\boldsymbol{v}\in\boldsymbol{V}} \Psi_v\left(\boldsymbol{y}\big|_{\boldsymbol{v}}\right)$$

And because this is a <u>conditional</u> network with $\boldsymbol{V} = \boldsymbol{X} \cup \boldsymbol{Y}$

$$\Psi(\boldsymbol{Y}|X) = \prod_{\boldsymbol{e}\in \boldsymbol{Edge}} \Psi_e\left(\boldsymbol{y}\big|_{\boldsymbol{e}}, x\right) \prod_{\boldsymbol{v}\in\boldsymbol{V}} \Psi_v\left(\boldsymbol{y}\big|_{\boldsymbol{v}}, x\right)$$

An exponential identity gives us

$$\Psi(\boldsymbol{Y}|X) = \exp\left(\sum_{e\in \mathrm{E}} \log\left(\Psi_e\left(\boldsymbol{y}\big|_{\boldsymbol{e}}, x\right)\right) + \sum_{\boldsymbol{v}\in\boldsymbol{V}} \log\left(\Psi_v\left(\boldsymbol{y}\big|_{\boldsymbol{v}}, x\right)\right)\right)$$

Potentials can be ANY positive values… like linear combinations of arbitrary features

$$\Psi(\boldsymbol{Y}|X) = \exp\left(\sum_{\boldsymbol{e}\in\boldsymbol{E},\boldsymbol{k}\in\boldsymbol{K}} \lambda_k f_k\left(\boldsymbol{e}, \boldsymbol{y}\big|_e, x\right) + \sum_{\boldsymbol{v}\in\boldsymbol{V},\boldsymbol{k'}\in\boldsymbol{K'}} \mu_{k'} g_{k'}\left(\boldsymbol{v}, \boldsymbol{y}\big|_{\boldsymbol{v}}, x\right)\right)$$

Improved iterative scaling

- Want to maximize log-likelihodod with respect to parameters

  $\theta = (\lambda_1, \lambda_2, \cdots; \mu_1, \mu_2, \cdots)$

[1] Della Pietra *et al.* (1997)

Improved iterative scaling

- Want to maximize log-likelihodod with respect to parameters $\theta = (\lambda_1, \lambda_2, \cdots; \mu_1, \mu_2, \cdots)$
- Method: *Improved Iterative Scaling*[1]: Extension of Generalised Iterative Scaling (Darroch and Ratcliff 1972).

[1] Della Pietra *et al.* (1997)

Improved iterative scaling

- Want to maximize log-likelihodod with respect to parameters $\theta = (\lambda_1, \lambda_2, \cdots; \mu_1, \mu_2, \cdots)$
- Method: *Improved Iterative Scaling*[1]: Extension of Generalised Iterative Scaling (Darroch and Ratcliff 1972).
- *Improved* because features need not sum to constant.

[1] Della Pietra *et al.* (1997)

Improved iterative scaling

- Want to maximize log-likelihodod with respect to parameters $\theta = (\lambda_1, \lambda_2, \cdots; \mu_1, \mu_2, \cdots)$
- Method: *Improved Iterative Scaling*[1]: Extension of Generalised Iterative Scaling (Darroch and Ratcliff 1972).
- *Improved* because features need not sum to constant.
- Idea: new set of parameters $\theta' = \theta + \delta\theta = (\lambda_1 + \delta\lambda_1, \cdots; \mu_1 + \delta\mu_1 \cdots)$ which will not decrease objective function. Iteratively apply!

[1] Della Pietra *et al.* (1997)

Improved iterative scaling

- Want to maximize log-likelihodod with respect to parameters $\theta = (\lambda_1, \lambda_2, \cdots ; \mu_1, \mu_2, \cdots)$
- Method: *Improved Iterative Scaling*[1]: Extension of Generalised Iterative Scaling (Darroch and Ratcliff 1972).
- *Improved* because features need not sum to constant.
- Idea: new set of parameters $\theta' = \theta + \delta\theta = (\lambda_1 + \delta\lambda_1, \cdots ; \mu_1 + \delta\mu_1 \cdots)$ which will not decrease objective function. Iteratively apply!
- Problem: slow, and nobody uses this any more.

[1] Della Pietra *et al.* (1997)

Modern CRF training - L-BFGS

- Generally use L-BFGS[2] algorithm.

[2] Limited-Memory Broyden-Fletcher-Goldfarb-Shanno Algorithm

Modern CRF training - L-BFGS

- Generally use L-BFGS[2] algorithm.
- Approximates Newton's method. Optimise multivariate function $f(\theta)$ through updates

$$\theta_{t+1} = \theta_t - [Hf(\theta_t)]^{-1} \nabla f(\theta_t)$$

[2] Limited-Memory Broyden-Fletcher-Goldfarb-Shanno Algorithm

Modern CRF training - L-BFGS

- Generally use L-BFGS[2] algorithm.
- Approximates Newton's method. Optimise multivariate function $f(\theta)$ through updates

$$\theta_{t+1} = \theta_t - [Hf(\theta_t)]^{-1} \nabla f(\theta_t)$$

- *Quasi*-Newtonian: approximates Hessian $Hf(\theta)$.

[2] Limited-Memory Broyden-Fletcher-Goldfarb-Shanno Algorithm

Modern CRF training - L-BFGS

- Generally use L-BFGS[2] algorithm.
- Approximates Newton's method. Optimise multivariate function $f(\theta)$ through updates

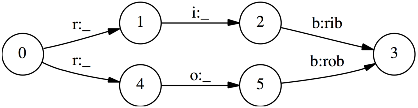
$$\theta_{t+1} = \theta_t - [Hf(\theta_t)]^{-1} \nabla f(\theta_t)$$

- *Quasi*-Newtonian: approximates Hessian $Hf(\theta)$.
- Limited-memory: doesn't store full (approximate) Hessian.

[2] Limited-Memory Broyden-Fletcher-Goldfarb-Shanno Algorithm

Label bias

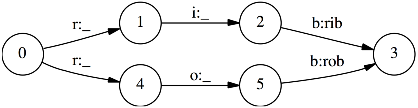
- Generate data with noisy HMM.
- 4-state system (not counting 'initial state'), transitions:
  - $1 \Rightarrow 2 \Rightarrow 3$
  - $4 \Rightarrow 5 \Rightarrow 3$



- Emissions: highly biased!
  - $P(X = Y\text{'s preferred value} \,|\, Y) = 29/32$
  - $P(X = \text{other} \,|\, Y) = 1/32$
- Preferred values: 1→ **'r'**, 4→ **'r'**, 2→ **'i'**, 5→ **'o'**, 3→ **'b'**.

Label bias

- Generate data with noisy HMM.
- 4-state system (not counting 'initial state'), transitions:
  - $1 \Rightarrow 2 \Rightarrow 3$
  - $4 \Rightarrow 5 \Rightarrow 3$

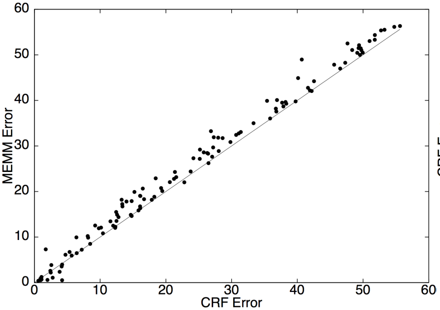


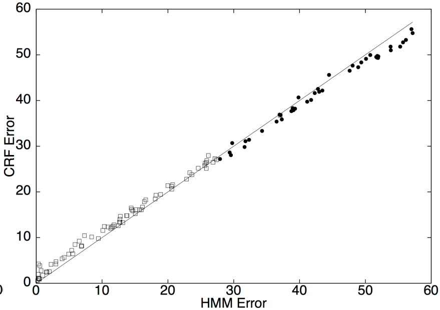
- Emissions: highly biased!
  - $P(X = Y\text{'s preferred value} \mid Y) = 29/32$
  - $P(X = \text{other} \mid Y) = 1/32$
- Preferred values: 1→ **'r'**, 4→ **'r'**, 2→ **'i'**, 5→ **'o'**, 3→ **'b'**.
- Result: CRF error 4.6%, MEMM error 42%.

Mixed-order sources

- Generate data with mixed-order HMM:
  - Transitions: $(1-\alpha)p_1(\mathbf{y}_i|\mathbf{y}_{i-1}) + \alpha p_2(\mathbf{y}_i|\mathbf{y}_{i-1}, \mathbf{y}_{i-2})$
  - Emissions: $(1-\alpha)p_1(\mathbf{x}_i|\mathbf{y}_i) + \alpha p_2(\mathbf{x}_i|\mathbf{y}_i, \mathbf{x}_{i-1})$
- Five labels, 26 observation values.
- Training/testing: 1000 sequences of length 25.
- CRF trained with Algorithm S (modified IIS). MEMM trained with iterative scaling.
- Viterbi to label test set.

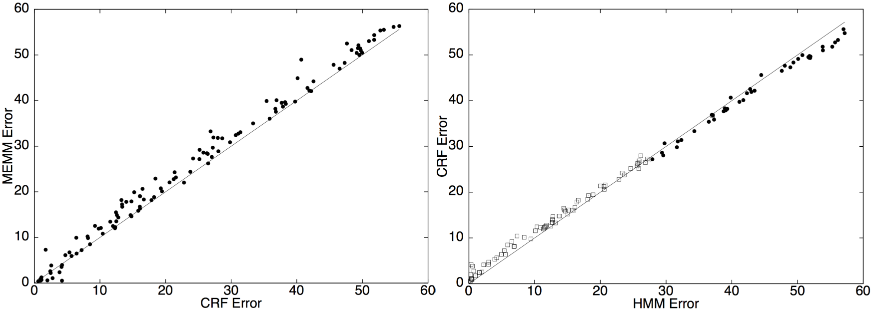
Mixed-order sources: results





- Squares : $\alpha < 0.5$.

Mixed-order sources: results



- Squares : $\alpha < 0.5$.
- CRF sort of wins?

Part of Speech Tagging

- Penn Treebank: 45 syntactic tags, label each word in sentence.
- Train first-order HMM, MEMM, CRF.

Part of Speech Tagging

- Penn Treebank: 45 syntactic tags, label each word in sentence.
- Train first-order HMM, MEMM, CRF.

model	*error*	*oov error*
HMM	5.69%	45.99%
MEMM	6.37%	54.61%
CRF	5.55%	48.05%
MEMM$^+$	4.81%	26.99%
CRF$^+$	4.27%	23.76%

$^+$Using spelling features

Part of Speech Tagging

- Penn Treebank: 45 syntactic tags, label each word in sentence.
- Train first-order HMM, MEMM, CRF.

model	*error*	*oov error*
HMM	5.69%	45.99%
MEMM	6.37%	54.61%
CRF	5.55%	48.05%
MEMM^{+}	4.81%	26.99%
CRF^{+}	4.27%	23.76%

$^{+}$Using spelling features

- Spelling features exploit conditional framework.
- Examples: starts with number/upper case?, contains hyphen, has suffix?

Skip-chain CRF

- Example: skip-chain CRF[3].

[3] From *An Introduction to Conditional Random Fields for Relational Learning*, Charles Sutton and Andrew McCallum, 2006.

Skip-chain CRF

- Example: skip-chain CRF[3].
- Has long-range features!

[3] From *An Introduction to Conditional Random Fields for Relational Learning*, Charles Sutton and Andrew McCallum, 2006.

Skip-chain CRF

- Example: skip-chain CRF[3].
- Has long-range features!
- Basic idea: extend linear-chain CRF by joining some distant observations with 'skip edges'.

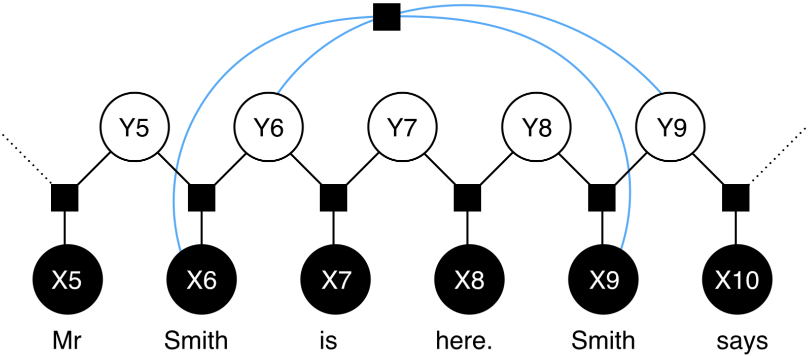
[3] From *An Introduction to Conditional Random Fields for Relational Learning*, Charles Sutton and Andrew McCallum, 2006.

Skip-chain CRF

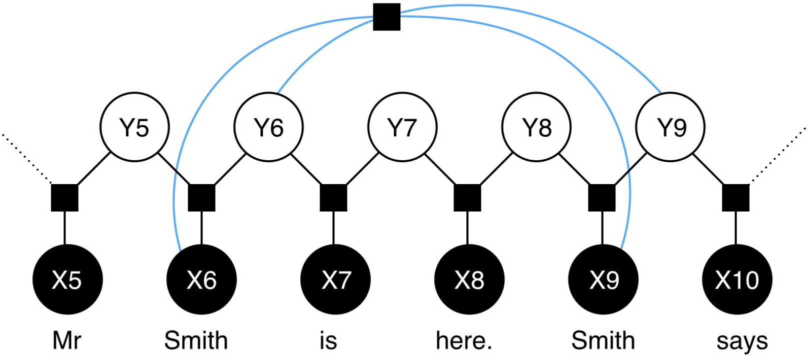
- Example: skip-chain CRF[3].
- Has long-range features!
- Basic idea: extend linear-chain CRF by joining some distant observations with 'skip edges'.
- Connect multiple mentions of entity across whole document.

[3] From *An Introduction to Conditional Random Fields for Relational Learning*, Charles Sutton and Andrew McCallum, 2006.

Example: Note: Squares denote factors (e.g. potential functions).

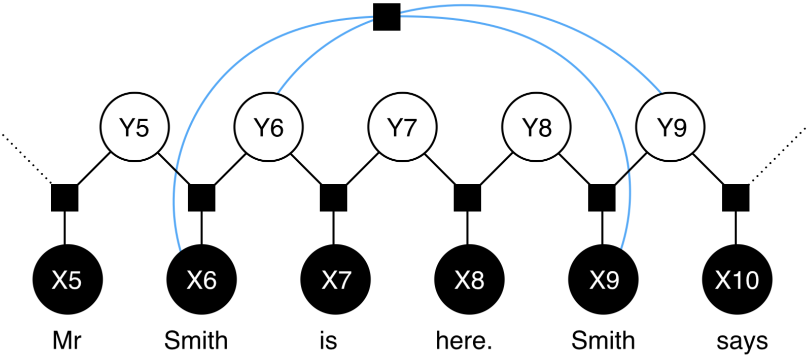


Example: Note: Squares denote factors (e.g. potential functions).



Question: Ignoring the skip edges (in blue), what potentials does Y_i appear in?

Example: Note: Squares denote factors (e.g. potential functions).



Question: Ignoring the skip edges (in blue), what potentials does Y_i appear in? Answer: $\psi(Y_i, Y_{i-1}, X_i)$, $\psi(Y_{i+1}, Y_i, X_{i+1})$.

Skip-chain task

- Data: 485 email announcements for seminars at CMU.

Skip-chain task

- Data: 485 email announcements for seminars at CMU.
- Task: identify start time, end time, location, speaker.

Skip-chain task

- Data: 485 email announcements for seminars at CMU.
- Task: identify start time, end time, location, speaker.
- Linear chain CRF with skip edges between identical capitalised words.

Skip-chain task

- Data: 485 email announcements for seminars at CMU.
- Task: identify start time, end time, location, speaker.
- Linear chain CRF with skip edges between identical capitalised words.
- Other word-specific features e.g. ‘appears in list of first names’, ‘upper case’, ‘appears to be part of time/date’ (by regex), etc.

Skip-chain results

System	stime	etime	location	speaker	overall
BIEN Peshkin and Pfeffer [2003]	96.0	**98.8**	87.1	76.9	89.7
Linear-chain CRF	**97.5**	97.5	**88.3**	77.3	90.2
Skip-chain CRF	96.7	97.2	88.1	**80.4**	**90.6**

- Values are F1 scores.

Skip-chain results

System	stime	etime	location	speaker	overall
BIEN Peshkin and Pfeffer [2003]	96.0	**98.8**	87.1	76.9	89.7
Linear-chain CRF	**97.5**	97.5	**88.3**	77.3	90.2
Skip-chain CRF	96.7	97.2	88.1	**80.4**	**90.6**

- Values are F1 scores.
- Repeated occurrences of speaker improve skip-chain performance.

Skip-chain results

System	stime	etime	location	speaker	overall
BIEN Peshkin and Pfeffer [2003]	96.0	**98.8**	87.1	76.9	89.7
Linear-chain CRF	**97.5**	97.5	**88.3**	77.3	90.2
Skip-chain CRF	96.7	97.2	88.1	**80.4**	**90.6**

- Values are F1 scores.
- Repeated occurrences of speaker improve skip-chain performance.
- Tokens are *consistently* classified by skip-chain. Linear-chain is inconsistent on **30.2** speakers, skip-chain: **4.8**.

Summary

- CRFs combine *discriminative* (e.g. MEMM) and *undirected* (e.g. MRF) properties to solve problems:

Summary

- CRFs combine *discriminative* (e.g. MEMM) and *undirected* (e.g. MRF) properties to solve problems:
  - Global normalisation avoids label bias.

Summary

- CRFs combine *discriminative* (e.g. MEMM) and *undirected* (e.g. MRF) properties to solve problems:
  - Global normalisation avoids label bias.
  - Conditioning on observations avoids modelling complex dependencies.

Summary

- CRFs combine *discriminative* (e.g. MEMM) and *undirected* (e.g. MRF) properties to solve problems:
    - Global normalisation avoids label bias.
    - Conditioning on observations avoids modelling complex dependencies.
    - Enables use of features using global structure.

Summary

- CRFs combine *discriminative* (e.g. MEMM) and *undirected* (e.g. MRF) properties to solve problems:
  - Global normalisation avoids label bias.
  - Conditioning on observations avoids modelling complex dependencies.
  - Enables use of features using global structure.
- Examples in paper strangely insubstantial, but CRFs are widely and successfully used.