Mayank Kejriwal
University of Southern California

Conditional Random Fields (Advanced)

Outline

- Modeling
- Inference
- Training
- Applications

slide by Daniel Khashabi

Parameter Learning

- Given the training data, $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$ wish to learn parameters of the model.
- For chain or tree structured CRFs, they can be trained by maximum likelihood
  - The objective function for chain-CRF is convex(see Lafferty et al(2001)).
- General CRFs are intractable hence approximation solutions are necessary

Parameter Learning

- Given the training data, $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$ we wish to learn parameters of the mode.
- Conditional log-likelihood for a general CRF:

$$\mathcal{L}(\boldsymbol{\lambda}) = \sum_k \left[\log \frac{1}{Z(\boldsymbol{x}^{(k)})} + \sum_j \lambda_j F_j(\boldsymbol{y}^{(k)}, \boldsymbol{x}^{(k)}) \right]$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\lambda})}{\partial \lambda_j} = E_{\tilde{p}(\boldsymbol{Y}, \boldsymbol{X})} \left[F_j(\boldsymbol{Y}, \boldsymbol{X}) \right] - \sum_k E_{p(\boldsymbol{Y} | \boldsymbol{x}^{(k)}, \boldsymbol{\lambda})} \left[F_j(\boldsymbol{Y}, \boldsymbol{x}^{(k)}) \right]$$

Empirical Distribution

Hard to calculate!

- It is not possible to analytically determine the parameter values that maximize the log-likelihood – setting the gradient to zero and solving for λ does not always yield a closed form solution. (Almost always)

Parameter Learning

- This could be done using gradient descent

$$\lambda \propto \max_{\lambda} \mathrm{L}(\lambda; y \mid x) \propto \max_{\lambda} \log \sum_{i=1}^{N} p(\mathbf{y} \mid \mathbf{x}; \lambda)$$

$$\lambda_{i+1} \leftarrow \lambda_i + \alpha . \nabla_{\lambda} \mathrm{L}(\lambda; y \mid x)$$

- Until we reach convergence

$$|\mathrm{L}(\lambda_{i+1}; y \mid x) - \mathrm{L}(\lambda_i; y \mid x)| < \text{Ć}$$

- Or any other optimization:
  - Quasi-Newton methods: BFGS [Bertsekas,1999] or L-BFGS [Byrd, 1994]
- General CRFs are intractable hence approximation solutions are necessary

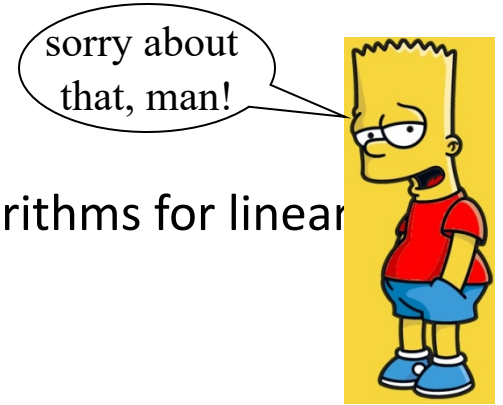
Compared with Markov chains, CRF's should be more discriminative, much slower to train and possibly more susceptible to over-training.

- Regularization:
  - σ is a regularization parameter

$$f_{objective}(\theta) = P_{\theta}(\mathbf{y} \mid \mathbf{x}) - \frac{\| \theta \|^2}{2\sigma^2}$$

Training (and Inference): General Case

- Approximate solution, to get faster inference.
- Treat inference as shortest path problem in the network consisting of paths(with costs)
  - Max Flow-Min Cut (Ford-Fulkerson, 1956)
- Pseudo-likelihood approximation:
  - Convert a CRF into separate patches; each consists of a hidden node and true values of neighbors; Run ML on separate patches
  - Efficient but may over-estimate inter-dependencies
- Belief propagation?!
  - variational inference algorithm
  - it is a direct generalization of the exact inference algorithms for linear chain CRFs
- Sampling based method(MCMC)



CRF frontiers

- Bayesian CRF:
  - Because of the large number of parameters in typical applications of CRFs
    - prone to overfitting.
    - Regularization?
    - Instead of
      - $\mathbf{y}^* = \max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x};\hat{\theta})$
      - $\mathbf{y}^* = \max_{\mathbf{y}} \int p(\mathbf{y}|\mathbf{x};\theta) p(\theta|\mathbf{x}^{(1)},\mathbf{y}^{(1)},\ldots,\mathbf{x}^{(N)},\mathbf{y}^{(N)}) d\theta$
      - Too complicated! How can we approximate this?
- Semi-supervised CRF:
  - The need to have big labeled data!
  - Unlike in generative models, it is less obvious how to incorporate unlabelled data into a conditional criterion, because the unlabelled data is a sample from the distribution

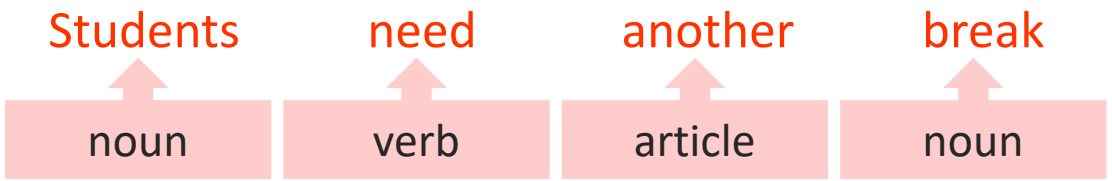
$$p(\mathbf{x})$$

Outline

- Modeling
- Inference
- Training
- Some Applications

Some applications: Part-of-Speech-Tagging

- POS(part of speech) tagging; the identification of words as nouns, verbs, adjectives, adverbs, etc.



- CRF features:

Feature Type	Description
Transition	$\forall k,k'\ y_i = k$ and $y_{i+1}=k'$
Word	$\forall k,w\ y_i = k$ and $x_i=w$ $\forall k,w\ y_i = k$ and $x_{i-1}=w$ $\forall k,w\ y_i = k$ and $x_{i+1}=w$ $\forall k,w,w'\ y_i = k$ and $x_i=w$ and $x_{i-1}=w'$ $\forall k,w,w'\ y_i = k$ and $x_i=w$ and $x_{i+1}=w'$
Orthography: Suffix	$\forall s$ in {"ing","ed","ogy","s","ly","ion","tion", "ity", …} and $\forall k\ y_i=k$ and x_i ends with s
Orthography: Punctuation	$\forall k\ y_i = k$ and x_i is capitalized $\forall k\ y_i = k$ and x_i is hyphenated …

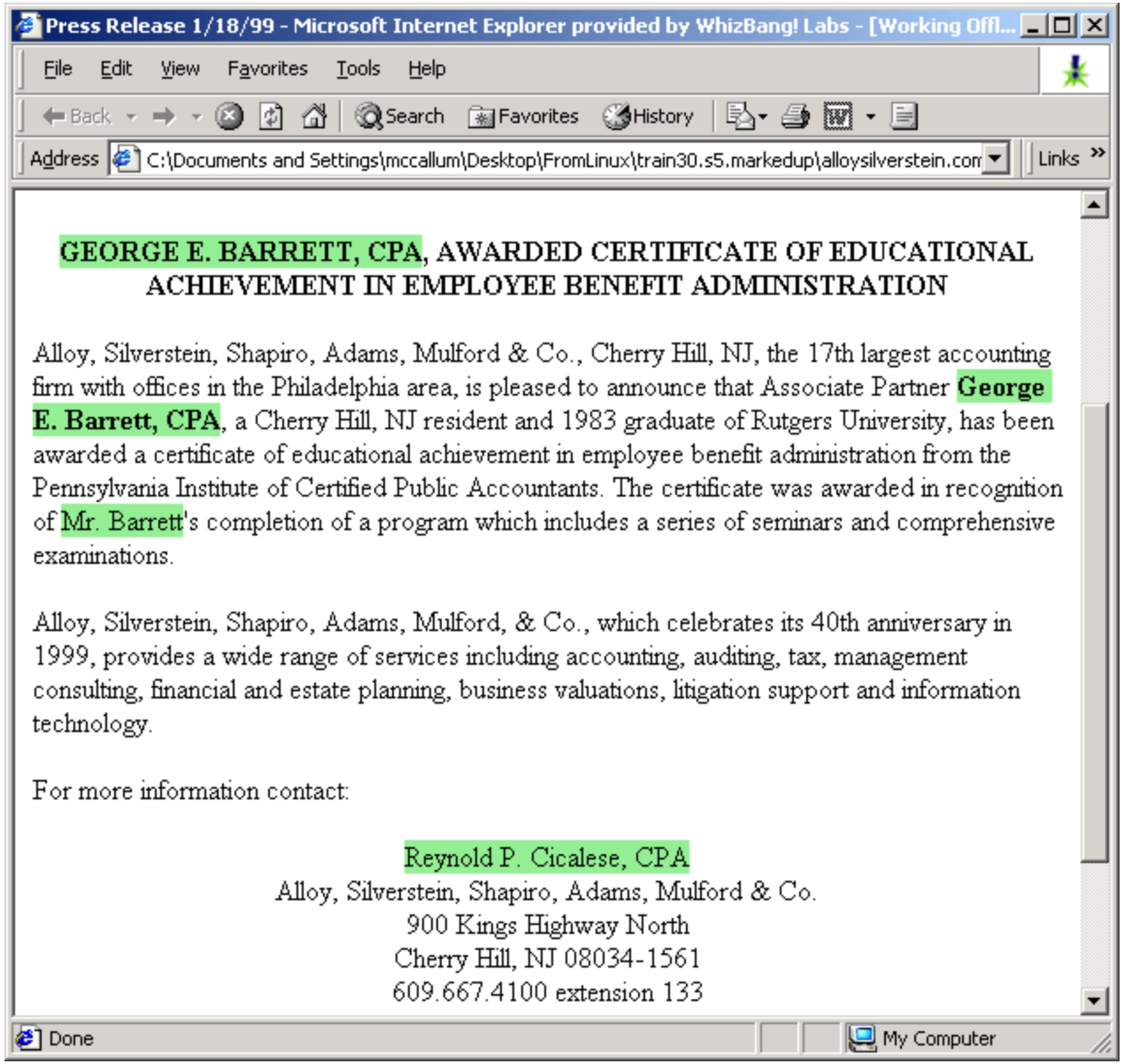
slide by Daniel Khashabi

CRF for Information Extraction

- CRF gives us P(label | obs, model)
  - Extraction: Find most probable label sequence (y's), given an observation sequence (x's)
    - 2001 Ford Mustang GT V-8 Convertible - $12700
  - No more independence assumption
    - Conditionally trained for whole label sequence (given input)
      - "long range" features (future/past states)
      - Multi-features
    - Now we can use better features!
      - What are good features for identifying a price?

Person name Extraction

[McCallum 2001, unpublished]



Features in Experiment

Capitalized	Xxxxx
Mixed Caps	XxXxxx
All Caps	XXXXX
Initial Cap	X….
Contains Digit	xxx5
All lowercase	xxxx
Initial	X
Punctuation	.,:;!(), etc
Period	.
Comma	,
Apostrophe	‘
Dash	-
Preceded by HTML tag	

Character n-gram classifier says string is a person name (80% accurate)

In stopword list (the, of, their, etc)

In honorific list (Mr, Mrs, Dr, Sen, etc)

In person suffix list (Jr, Sr, PhD, etc)

In name particle list (de, la, van, der, etc)

In Census lastname list; segmented by P(name)

In Census firstname list; segmented by P(name)

In locations lists (states, cities, countries)

In company name list (“J. C. Penny”)

In list of company suffixes (Inc, & Associates, Foundation)

Hand-built FSM person-name extractor says yes, (prec/recall ~ 30/95)

Conjunctions of all previous feature pairs, evaluated at the current time step.

Conjunctions of all previous feature pairs, evaluated at current step and one step ahead.

All previous features, evaluated two steps ahead.

All previous features, evaluated one step behind.

Total number of features = ~200k

Training and Testing

- Trained on 65469 words from 85 pages, 30 different companies’ web sites.
- Training takes 4 hours on a 1 GHz Pentium.
- Training precision/recall is 96% / 96%.

- Tested on different set of web pages with similar size characteristics.
- Testing precision is 92 – 95%,
  recall is 89 – 91%.

References

- J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. ICML01, 2001.
- Charles Elkan, “Log-linear Models and Conditional Random Field,” Notes for a tutorial at CIKM, 2008.
- Charles Sutton and Andrew McCallum, “An Introduction to Conditional Random Fields for Relational Learning,” MIT Press, 2006
- Slides: An Introduction to Conditional Random Field, Ching-Chun Hsiao
- Hanna M. Wallach , Conditional Random Fields: An Introduction, 2004
- Sutton, Charles, and Andrew McCallum. *An introduction to conditional random fields for relational learning*. Introduction to statistical relational learning. MIT Press, 2006.
- Sutton, Charles, and Andrew McCallum. "An introduction to conditional random fields." *arXiv preprint arXiv:1011.4088* (2010).
- *B. Majoros,* Conditional Random Fields, for eukaryotic gene prediction

slide by Daniel Khashabi