BTRY 6790, Probabilistic Graphical Models

Oct. 24, 2013

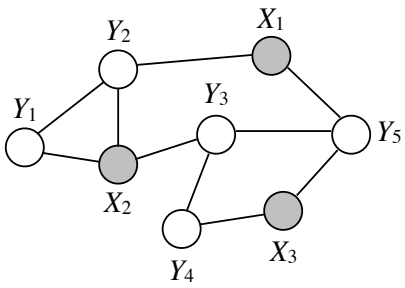
Plan for Today

- Conditional random fields (CRFs)

Conditional Random Fields

- CRFs are undirected graphical models in which a set of variables Y is conditioned on another set of (observed) variables X
- As usual, X and Y correspond to nodes in an undirected graph, which defines a factorized joint probability
- In this case, however, X consists of *input variables* or *covariates*, and the conditional distribution $p(Y \mid X)$ is of interest
- CRFs are useful for *discriminative* learning

Example



Goal: compute $p(Y \mid X)$

Recall: Major Strategies

Generative

- Full model for classes & data
- Classes inferred by Bayes' rule
- Complete description can be useful
- But accurate classification requires modeling the data

Discriminative

- Directly estimate the class distribution given the data
- May be harder to interpret in some cases
- But will be optimized directly for the task of interest

Factorization

- Let $\mathcal{G} = (V, E)$ be a *factor graph* over Y with factors $F = \{\psi_C\}$ for subsets $C \subseteq Y$
- The graph $\mathcal{G}$ defines a CRF if, for any fixed x, $p(y \mid x)$ factorizes according to F:

$$p(y|x) = \frac{1}{Z(x)} \prod_C \psi_C(y_C, x)$$

- Suppose the factors have (unnormalized) exponential family forms:

$$\psi_C(y_C, x) = \exp\left\{\sum_k \lambda_{Ck} f_{Ck}(y_C, x_{Ck})\right\}$$

- Then:

$$p(y|x) = \frac{1}{Z(x)} \exp\left\{\sum_C \sum_{k=1}^{K(C)} \lambda_{Ck} f_{Ck}(y_C, x)\right\}$$

Features

- Typically each f_{Ck} picks out particular "features" of X. Let x_{Ck} in X be the subset needed for f_{Ck}. Thus,

$$\psi_C(y_C, x) = \exp\left\{\sum_k \lambda_{Ck} f_{Ck}(y_C, x_{Ck})\right\}$$

- The framework is very general. The f_{Ck} functions may be parametric or non-parametric. Delta functions are common, e.g., $f_{Ck}(y_C, x_{Ck}) = I(y_C \text{ is proper noun}, x_{Ck} \text{ is upper case})$
- Note that, if the f_{Ck} functions are held fixed, then $p(y \mid x)$ is concave

Logistic Regression

- Recall the case of logistic (softmax) regression: multinomial Y conditioned on m-dimensional X, with $p(y \mid x)$ given by:

$$p(y|x) = \frac{\exp(\theta_y^T x)}{\sum_{\tilde{y}} \exp(\theta_{\tilde{y}}^T x)}$$

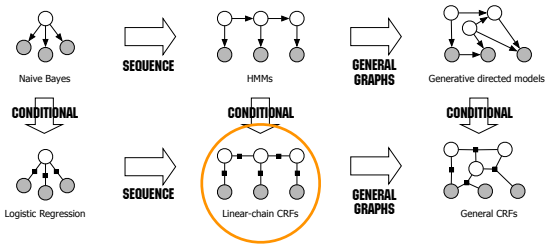
- This can be shown to be an exponential-family CRF, with $\{C_1, ..., C_m\}$ all equal to $\{Y\}$, $K(C_i) = m$, $\lambda_{Cik} = \theta_{ik}$, and $f_{Cik}(y_{Ci}, x_{Cik}) = x_k \delta(y, i)$

$$p(y|x) = \frac{1}{Z(x)} \exp\left\{\sum_C \sum_{k=1}^{K(C)} \lambda_{Ck} f_{Ck}(y_C, x_{Ck})\right\} = \frac{\exp(\sum_{k=1}^M \theta_{yk} x_k)}{\sum_{\tilde{y}} \sum_{k=1}^M \exp(\theta_{\tilde{y}k} x_k)}$$

Applications

- CRFs are broadly applicable
- They are most appropriate for supervised learning problems, typically in a batch setting
- Examples: document classification, part-of-speech tagging, information extraction, computer vision, bioinformatics
- Special cases of interest: linear-chain CRFs, dynamic CRFs, relational Markov networks

Taxonomy of Models



Sutton & McCallum, in *Introduction to Statistical Relational Learning*, 2006

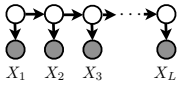
Linear-Chain CRFs

- Linear-chain CRFs generalize HMMs. They differ from HMMs in three key ways:
  - They are undirected rather than directed models (not locally normalized)
  - They use generalized "emission" and "transition" potentials that consider arbitrarily defined features of one or more sequences, as well as other covariates
  - They are applied in a discriminative rather than a generative setting (conditioned on input sequence)

Recall: HMMs

- The joint probability of a sequence and a path (the complete data likelihood) is:

$$p(x, z|\theta) = p(z_1|\pi) \prod_{i=2}^{L} p(z_i|z_{i-1}, A) \prod_{i=1}^{L} p(x_i|z_i, \eta)$$
$$= \pi_{z_1} \prod_{i=2}^{L} a_{z_{i-1}, z_i} \prod_{i=1}^{L} p(x_i|z_i, \eta)$$



- The incomplete data likelihood is:

$$p(x|\theta) = \sum_z \pi_{z_1} \prod_{i=2}^{L} a_{z_{i-1}, z_i} \prod_{i=1}^{L} p(x_i|z_i, \eta)$$
$$= \sum_{z_1} \sum_{z_2} \cdots \sum_{z_L} \pi_{z_1} \prod_{i=2}^{L} a_{z_{i-1}, z_i} \prod_{i=1}^{L} p(x_i|z_i, \eta)$$

Limitations of HMMs

- As generative classification models, HMMs are trained to optimize

$$p(x, z|\theta) = p(z|x, \theta)p(x|\theta)$$

rather than

$$p(z|x, \theta)$$

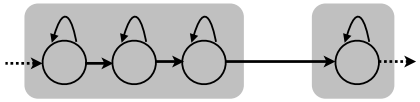
- Furthermore, they have a restricted ability to model x—no long-range dependencies, limited interactions between features, etc
- Finally, local normalization can lead to a problem called "label bias" when states have different numbers of outgoing arcs

Label Bias

- Suppose two parts of an HMM compete, but part 1 allows for many possible paths and part 2 allows for few
- Even if the total probability of part 1 is greater, the Viterbi algorithm will tend to choose part 2
- The problem is essentially that HMM states can decide *where* to pass their probability mass but not *how much*
- This leads to an implicit bias toward states in which paths are concentrated

Lafferty, McCallum & Pereira, ICML 2001

Example



Exon *Intron*

Linear-Chain CRFs

- Let F be the set of consecutive pairs in Y, $F = \{y_{i-1}, y_i: i = 1, ..., L\}$ (with y_0=B), and let G be the set of nodes in Y, $G = \{y_i: i = 1, ..., L\}$
- Let J factors, $f_1, ..., f_J$, with coefficients $\lambda_1, ..., \lambda_J$, be associated with every $C \in F$
- Let K factors, $g_1, ..., g_K$, with coefficients $\mu_1, ..., \mu_K$, be associated with every $D \in G$
- The conditional probability factors as:

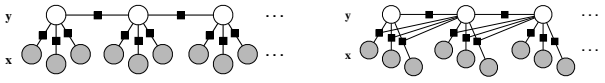
$$p(y|x) = \frac{1}{Z(x)} \exp\left\{\left(\sum_{C\in F}\sum_{j=1}^{J}\lambda_j f_j(y_C, x)\right) + \left(\sum_{D\in G}\sum_{k=1}^{K}\mu_k g_k(y_D, x)\right)\right\}$$

$$= \frac{1}{Z(x)} \exp\left\{\sum_{i=1}^{L}\left(\sum_{j=1}^{J}\lambda_j f_j(y_{i-1}, y_i, x)\right) + \left(\sum_{k=1}^{K}\mu_k g_k(y_i, x)\right)\right\}$$

Linear-Chain CRF, cont

$$p(y|x) = \frac{1}{Z(x)} \exp\left\{\sum_{i=1}^{L}\left(\sum_{j=1}^{J}\lambda_j f_j(y_{i-1}, y_i, x)\right) + \left(\sum_{k=1}^{K}\mu_k g_k(y_i, x)\right)\right\}$$

- If $J = K = 1$, $\lambda_1 = \mu_1 = 1$, $f_1(y_{i-1}, y_i, x) = \log p(y_i \mid y_{i-1})$, and $g_1(y_i, x) = \log p(x_i \mid y_i)$ this model is equivalent to an HMM
- It is a *locally unnormalized*, *conditional* generalization of an HMM, which makes use of x in a highly flexible, general way



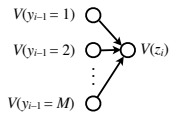
Lafferty, McCallum & Pereira, ICML 2001

Viterbi

- Despite the richness of the model, the Viterbi algorithm is essentially unchanged
- Let $V(y_i)$ be the log weight of the most likely path prefix that ends in state y_i
- Base case: $V(y_0 = B) = 0$, $V(y_i = B) = -\infty$ for $i > 0$, $V(y_0) = -\infty$ for $y_0 \neq B$
- Recurrence:

$$V(y_i) = \max_{y_{i-1}}\left\{V(y_{i-1}) + \left(\sum_{j=1}^{J}\lambda_j f_j(y_{i-1}, y_i, x)\right) + \left(\sum_{k=1}^{K}\mu_k g_k(y_i, x)\right)\right\}$$

- Termination: $V(y_L) = \max_{y_L} V(y_L)$



Forward/Backward

- The forward and backward (α and β) algorithms follow in a similar way
- The termination steps now compute $Z(x)$ rather than $p(x)$
- As before, marginal state probabilities can be obtained by locally normalizing the products: $p(y_i \mid x) = \alpha(y_i)\,\beta(y_i) / q_i$, where $q_i = \sum \alpha(y_i)\,\beta(y_i)$

General Inference

- In general, conditioning on X does not fundamentally change the inference problem
- When Y is defined by a polytree, a slightly adapted sum-product algorithm can be used
- For more general graphs, the junction tree algorithm, approximate methods, or sampling can be used

Parameter Estimation

- Parameter estimation is accomplished by conditional maximum likelihood
- It is complicated by a $-\log Z(x)$ term in the conditional log likelihood
- *Regularization* (≈ prior on coefficients) is typically needed to avoid overfitting
- Maximization is typically accomplished by a quasi-Newton or conjugate gradients method
- Iterative scaling algorithms are also available (Lafferty et al., 2001)
- See Sutton & McCallum, 2006

Limitations

- Sometimes HMMs are still preferred
  - CRF parameters can be hard to interpret
  - CRFs cannot be used in an unsupervised setting
  - They cannot be used to generate data
  - HMMs can be more natural in an online setting
- CRFs are inherently linear; extensions to neural nets, kernel regression are needed to capture nonlinearities
- Some advantages of CRFs can be achieved with directed models (MEMMs, IO-HMMs). Even HMMs can be trained by conditional ML.

That's All

- See articles on CRFs (website)
- Homework #4 due Mon Nov 4
- Project info on website and project proposal coming