Conditional Training of Undirected Models

Sargur Srihari
srihari@cedar.buffalo.edu

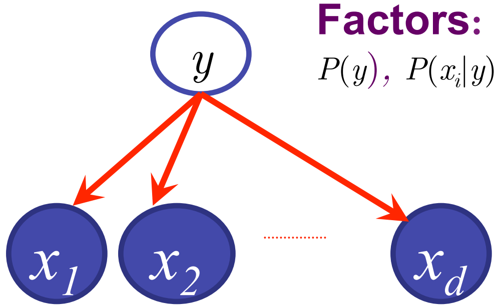
Topics

- Generative BN vs Conditional MN
- Conditionally Trained Models
- Log-conditional likelihood
- Conditional Training Complexity
- Generative and discriminative models for sequence training

Parameters: Gen. BN vs. Disc. MN

Classification Problem: Features $\boldsymbol{x} = \{x_1, .. x_d\}$ and two-class label y

Naïve Bayes (Generative BN): CPD parameters for $p(x_i|y)$



Factors:
$P(y)$, $P(x_i|y)$

Joint Probability:

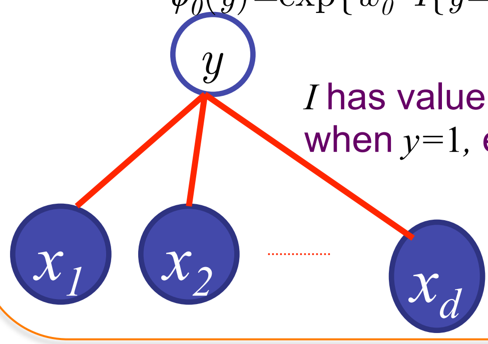
$$P(y,\mathrm{x}) = P(y)\prod_{i=1}^{d} P(x_i \mid y)$$

From joint infer $P(y|\boldsymbol{x})$

Learning: If each x_i is discrete with k values independently estimate $d(k\text{-}1)$ parameters
But independence is false
For sparse data generative is better
C-class problem: $d(k\text{-}1)(C\text{-}1)$ parameters

Logistic Regression (Conditional MN): feature parameters w_i

Factors (log-linear w. features):
$D_i = \{x_i, y\}$ $f_i(D_i) = x_i\, I(y)$
$\phi_i(x_i, y) = \exp\{w_i x_i\, I\{y=1\}\}$,
$\phi_0(y) = \exp\{w_0\ I\{y=1\}\}$

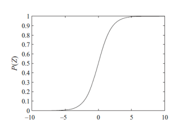


I has value 1 when $y=1$, else 0

Conditional Probability:

Unnormalized $\tilde{P}(y=1 \mid \mathrm{x}) = \exp\left\{w_0 + \sum_{i=1}^{d} w_i x_i\right\}$ $\tilde{P}(y=0 \mid \mathrm{x}) = \exp\{0\} = 1$

Normalized $P(y=1 \mid \mathrm{x}) = sigmoid\left\{w_0 + \sum_{i=1}^{d} w_i x_i\right\}$ where $sigmoid(z) = \dfrac{e^z}{1+e^z} = \dfrac{1}{1+e^{-z}}$



sigmoid

Z has term 1 because $\tilde{P}(y=0 \mid \mathrm{x})=1$

Learning: Jointly optimize d parameters w_i
High dimensional estimation
but correlations accounted for
Can use much richer features:
Edges, image patches sharing same pixels

C-class

$$p(y_c \mid \mathrm{x}) = \frac{\exp(\mathrm{w}_c^T \mathrm{x})}{\sum_j^C \exp(\mathrm{w}_j^T \mathrm{x})}$$

$C \times d$ parameters

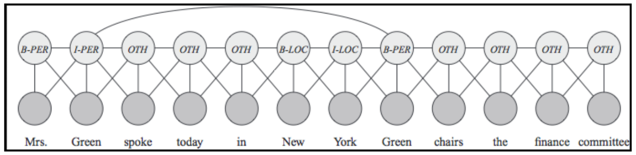
Conditionally Trained Models

- Often we want to perform a particular inference
  - where we have a known set of variables, or features, X
- We want to query a pre-determined set of variables Y
- We prefer to use *discriminative training*
- Train the network as a *Conditional Random Field* (*CRF*) that encodes a conditional distribution $P(Y|X)$

Log-Conditional Likelihood

- Train the network as a CRF that encodes a conditional distribution $P(Y|X)$
  - Training set consists of M pairs

$$D=\{\boldsymbol{y}[m],\ \boldsymbol{x}[m]\},\ m=1,..,\ M$$



Example:
$\boldsymbol{y}[m]$=word category, $B\text{-}PER$
$\boldsymbol{x}[m]$=word, $Mrs.$

- Objective Function: Log-Conditional likelihood

$$\boxed{E_{(\boldsymbol{x},\boldsymbol{y})\sim P^*}\left[\log \tilde{P}\left(\boldsymbol{y} \mid \boldsymbol{x}\right)\right]}$$

We are not interested in the distribution of $\boldsymbol{x}$ variables; only predicting $\boldsymbol{y}$ given $\boldsymbol{x}$

- Log-Conditional likelihood is

$$\boxed{\ell_{Y|X}\left(\boldsymbol{\theta}:D\right)=\ln P\left(\boldsymbol{y}[1,..,M] \mid \boldsymbol{x}[1,..,M],\boldsymbol{\theta}\right)=\sum_{m=1}^{M}\ln P\left(\boldsymbol{y}[m] \mid \boldsymbol{x}[m],\boldsymbol{\theta}\right)}$$

- In this objective, we are optimizing the likelihood of each observed assignment $\boldsymbol{y}[m]$ *given* observed assignment $\boldsymbol{x}[m]$
- Summation is over the M samples

Log-conditional likelihood is concave

- Each of the terms $\boxed{\ln P\left(\boldsymbol{y}[1,..,M] \mid \boldsymbol{x}[1,..,M],\boldsymbol{\theta}\right)}$
  - is a log-likelihood of a MN model with a different set of factors– the factors of the original network reduced by the observation $\boldsymbol{x}[1,\ldots,M]$ and its own partition function
- Because the sum of concave functions is concave, the log-likelihood is concave
- Implies that the function has a global optimum, not necessarily unique
- Gradient ascent can be used

Gradient of Conditional Likelihood

- A reduced MN is itself an MN.
- We use log-linear representation with features f_i and parameters θ
  - Analogous to gradient for full MN

$$\boxed{\frac{\partial}{\partial \theta_i}\frac{1}{M}\ell(\theta : D) = E_D[f_i(\chi)] - E_\theta[f_i]}$$

we can write gradient for reduced MN

$$\boxed{\frac{\partial}{\partial \theta_i}\ell_{\mathbf{Y}|\mathbf{X}}\left(\boldsymbol{\theta} : D\right) = \sum_{m=1}^{M}\left(f_i\left(\boldsymbol{y}[m], \boldsymbol{x}[m]\right) - E_{\boldsymbol{\theta}}\left[f_i \mid \boldsymbol{x}[m]\right]\right)}$$

First term is empirical count conditioned on $x[m]$

Second is based on running inference on each data case

Comparison with unconditional case

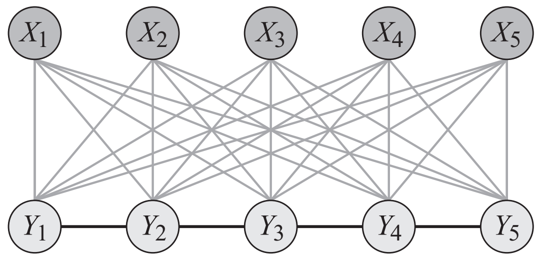
- The solution $\boxed{\frac{\partial}{\partial \theta_i} \ell_{\mathbf{Y}|\mathbf{X}}(\boldsymbol{\theta}:D) = \sum_{m=1}^{M} \left(f_i(\boldsymbol{y}[m], \boldsymbol{x}[m]) - E_{\boldsymbol{\theta}}\left[f_i \mid \boldsymbol{x}[m] \right] \right)}$ looks deceptively similar to $\boxed{\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta:D) = E_D[f_i(\chi)] - E_\theta[f_i]}$
  - Indeed if we aggregate the first component in each of the summands, we obtain precisely the empirical count of f_i in the data set D
- Key difference:
  - In the unreduced MN the expected feature counts
    - are computed relative to a single model
  - In the case of conditional MN th expected counts
    - are computed as the summation of counts in ensemble of models defined by conditioning variables x[m]
  - The difference has significant computational issues

Conditional Training Complexity

- In the unconditional case, each gradient step required only a single execution of inference
- When training CRF we must execute inference for every single data case, conditioning on $\boldsymbol{x}[m]$
- On the other hand inference is executed on a simpler model
  - Since conditioning on evidence can only reduce computational cost

Ex: Simplification due to Conditioning

- Very densely connected CRF for sequence labeling



- Full MN encodes

$$\tilde{P}(\boldsymbol{X},\boldsymbol{Y}) = \prod_{i=1}^{4} \phi_i \left(Y_i, Y_{i+1}\right) \prod_{i=1}^{5} \phi_i \left(Y_i, X_1, X_2, X_3, X_4, X_5\right)$$

  – It is densely connected

- Edges disappear in a reduced Markov network
  – After conditioning on $\boldsymbol{X}$

$$\tilde{P}(\boldsymbol{Y} \mid \boldsymbol{X}) = \prod_{i=1}^{5} \phi_i \left(Y_i, Y_{i+1} \mid X_1, X_2, X_3, X_4, X_5\right)$$

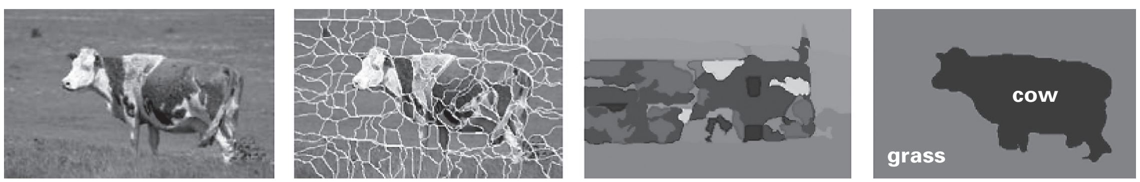
  – Remaining edges form a simple chain, allowing linear-time inference

Benefit of discriminative training

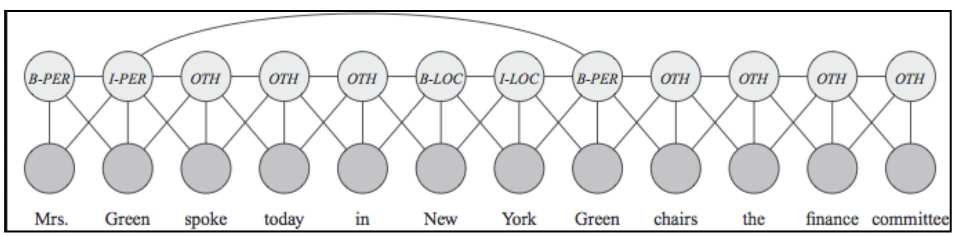
- Beneficial when the domain of $\boldsymbol{X}$ is very large or even infinite
- Ex: image classification task where we want to assign labels to pixels when features are given
  - Partition function in a generative setting involves summation over the space of all possible images
    - If we have an $N \times N$ image where each pixel takes 256 values the resulting space has 256^N values
    - Highly intractable inference problem even using approximate inference methods

Generative and Discriminative Models for Sequence Labeling

- A main task of PGMs: taking a set of inter-related instances and jointly labeling them
- Also called collective classification



- Super-pixel labeling
- A non-sequential task



- Named entity recognition
- A sequential labeling task

- We look at trade-offs in using different models for instances organized sequentially

The sequence labeling task

- Given: sequence of observations $\boldsymbol{X}=\{X_1,..X_k\}$
- Need: a joint label $\boldsymbol{Y}=\{Y_1,..Y_k\}$
- Text Analysis task:
  - sequence of words each of which we want to label with some label
- Activity Recognition task:
  - obtain a sequence of images and label each frame
  - With the activity taking place in it,
  - e.g., running, jumping, walking
- Assume that we want to construct a model for this task
  - and to train it using fully labeled training data,
  - where both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are observed

Three Models for Sequence Labeling

Given: sequence of observations $\boldsymbol{X}=\{X_1,..X_k\}$.
Need: a joint label $\boldsymbol{Y}=\{Y_1,..Y_k\}$

1. HMM is a directed *generative* model
   That needs joint probability $P(X,Y)$

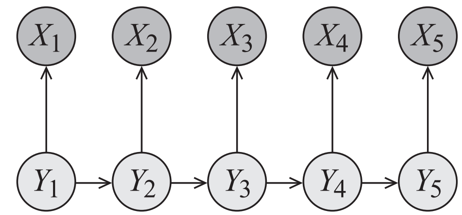
2. MEMM is also directed
   - But a *discriminative* model
   - Represents conditional distribution $P(Y|X)$

     $Y_1 \perp X_2$ if not given Y_2, by *D-separation*
     More generally, $Y_i \perp X_j \mid \boldsymbol{X}_{-j} \quad j > I$

3. CRF is a *discriminative* model

The three models present interesting trade-offs:
In their expressive power and learnability

HMM

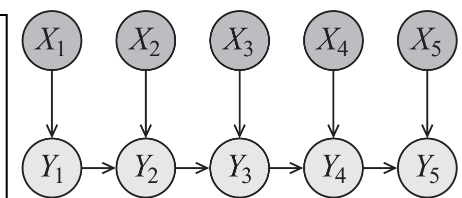


Needs joint distribution

$$P(\boldsymbol{X},\boldsymbol{Y}) = \prod_{i=1}^{k} P(X_i / Y_i) P(Y_i \mid Y_{i-1})$$

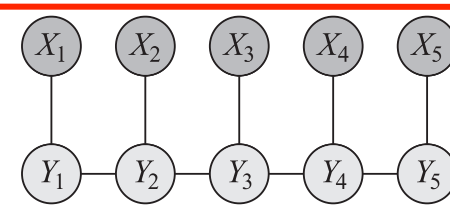
$$P(\boldsymbol{Y}/\boldsymbol{X}) = \frac{P(\boldsymbol{X},\boldsymbol{Y})}{P(\boldsymbol{X})}$$

MEMM
Does not model distribution over $\boldsymbol{X}$



$$P(\boldsymbol{Y} \mid \boldsymbol{X}) = \prod_{i=1}^{k} P(Y_i \mid X_i, Y_{i-1})$$

CRF
Directly obtains $P(\boldsymbol{Y}|\boldsymbol{X})$



Note: $Z(\boldsymbol{X})$ is marginal of un-normalized measure

$$P(\boldsymbol{Y} \mid \boldsymbol{X}) = \frac{1}{Z(X)} \tilde{P}(Y,X)$$

$$\tilde{P}(\boldsymbol{Y},\boldsymbol{X}) = \prod_{i=1}^{k-1} \phi_i(Y_i, Y_{i+1}) \prod_{i=1}^{k} \phi_i(Y_i, X_i)$$

$$Z(\boldsymbol{X}) = \sum_{Y} \tilde{P}(\boldsymbol{Y},\boldsymbol{X})$$

Comparison of Sequential Models

- Trade-offs: Training, Expression, Independence

1. Computational perspective: Training Effort
   - HMM, MEMM easily learned (they are BNs)
   - CRF: gradient-based inference for every sequence
     difficult with large data sets
2. Expressibility: use of a rich feature set
   - Performance strongly dependent on feature set
   - In HMM: explicitly model distribution over features
     - This type of model is very hard, almost impossible to correctly construct
   - MEMM , CRF are discriminative
     - hence avoid the challenge entirely

Independence Assumptions made by the model

- MEMM makes the assumption:

  $(Y_i \perp X_j \mid \boldsymbol{X}_{-j})$ for any $j > i$

  – Thus an observation later in the sequence has no effect on posterior probability of current state
    - i.e., model does not allow for any smoothing

  – Implications can be severe in many settings
    - In activity recognition in video sequence: frames are labeled as running/walking.
      – Earlier frames may be blurry but later ones clearer

  – Called the label bias problem

Summary of Trade-offs

- Trade-offs between these different models are subtle and non-definitive
- In cases where we have many correlated features, discriminative models are better
- If only limited data is available, the stronger bias of the generative model dominates and allow learning with fewer samples
- CRFs are a safe choice but computational cost is prohibitive for large data sets