Readings: K&F 20.3, 20.4, 20.6, 20.7

Learning undirected Models

Lecture 18 – June 1, 2011
CSE 515, Statistical Methods, Spring 2011

Instructor: Su-In Lee
University of Washington, Seattle

Is the energy functional convex in the parameters of Q?
- entropy x log(x) is concave in x
- xy is jointly convex in (x,y)

one over a small set of variables.

$$F[P_F, Q] = \sum_{\phi \in F} E_Q[\ln \phi] + H_Q(\mathbf{U})$$

$$E_Q[\ln \phi] = \sum_{\mathbf{u}_\phi} Q(\mathbf{u}_\phi) \ln \phi(\mathbf{u}_\phi) = \sum_{\mathbf{u}_\phi} \left(\prod_{X_i \in \mathbf{u}_\phi} Q(x_i)\right) \ln \phi(\mathbf{u}_\phi)$$

$$H_Q(\mathbf{U}) = \sum_i H_Q(X_i)$$

- The complexity of this expression depends on the size of the factors in P_F, and not on the topology of the network.

2

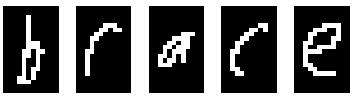
Learning Undirected Graphs

- The likelihood function
- Learning parameters
- Collective classification with HMM, MEMM, CRF
  - Generative vs. discriminative models
  - Directed vs. undirected models
- Learning with incomplete data
- Learning with Priors
  - Maximum A Priori (MAP) estimation
- Learning with alternative objectives
  - Pseudo likelihood objective
  - Max-margin learning
- Structure Learning

3

Collective Classification

- Taking a set of interrelated instances and jointly labeling them
  - Sequential labeling: labeling instances organized in a sequence
  - Example: handwriting recognition



x *A sequence of observations (feature)*

- Use local information
- Exploit correlations

b r a c e

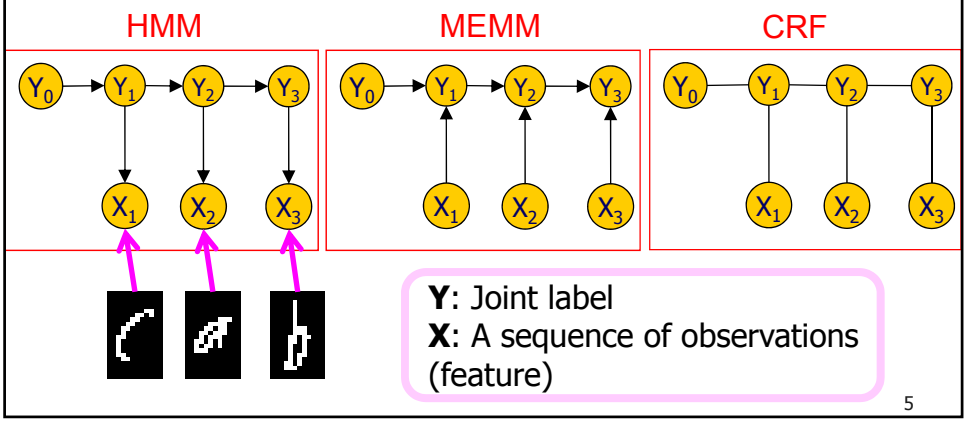
y *Label them with some joint label*

- Model-based approach
  - Training data: Fully labeled (both **Y** and **X** are observed)
  - Test data: only **X** is observed

4

Collective Classification

- Trade-offs between different models
  - Hidden Markov Model (HMM)
  - Maximum Entropy Markov Model (MEMM)
  - Conditional Random Field (CRF)

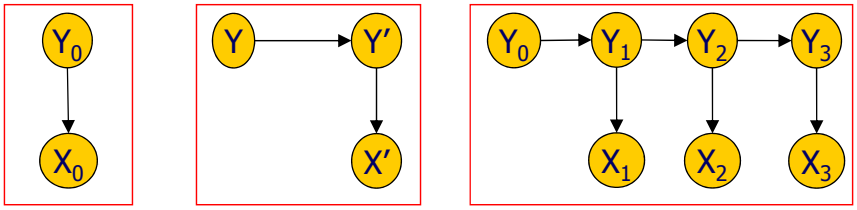


Y: Joint label
X: A sequence of observations (feature)

5

Hidden Markov Model

- For each classification task,
  - Single (hidden) state variable Y (e.g. label)
  - Single (observed) observation variable X (e.g. image)
- Observation probability P(X|Y)
  - For example, P(X=  | Y='b')
- Transition probability P(Y'|Y)
  - Statistical dependencies between the neighboring Y's

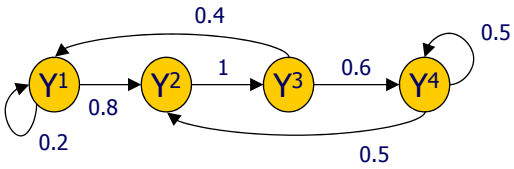


G_0 $G_{\rightarrow}$ Unrolled network

6

Hidden Markov Model

- For each classification task,
  - Single (hidden) state variable Y
  - Single (observed) observation variable X
- Observation probability P(X|Y)
- Transition probability P(Y'|Y) assumed to be sparse
  - Usually encoded by a state transition graph



State transition representation

P(Y'|Y)

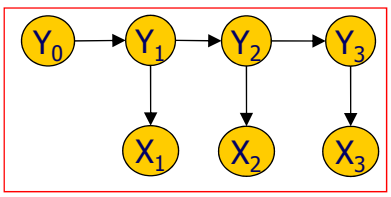
	y^1	y^2	y^3	y^4
y^1	0.2	0.8	0	0
y^2	0	0	1	0
y^3	0.4	0	0	0.6
y^4	0	0.5	0	0.5

7

Learning: Hidden Markov Model

- Generative models
  - Define a joint probability P(**Y**,**X**) over paired label **Y** and observation **X**
  - Parameters trained to maximize the joint log-likelihood log P(**Y**,**X**)
- Joint distribution
  - P(**X**,**Y**) = ?

HMM



- We can label new observations **x** by inferring P(**Y**|**X**=**x**)
  - To make inference tractable, there are typically no long-range dependencies (Markov assumption)

8

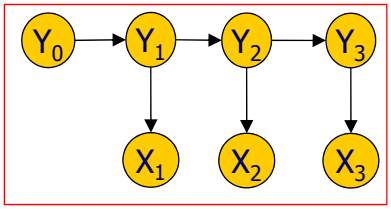
Discriminative (Conditional) Models

- Specifies the probability of possible label sequences given the observations, P(**Y**|**X**)
  - **X** is always observed
- Key advantage:
  - Does not "waste" parameters on modeling P(**X**)
  - Distribution over **Y** can depend on non-independent features **X** without modeling feature dependencies
  - Transition probabilities can depend on past and future
- Two representations
  - Maximum Entropy Markov Models (MEMMs)
  - Conditional Random Fields (CRFs)

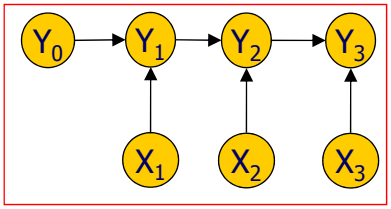
9

Max Entropy Markov Models

- Models the probability over the next state given the previous state and the observations
- Discriminative model: Provides a model for P(**Y**|**X**)
- Weakness: label bias problem
  - $(Y_i \perp X_j \mid \mathbf{X}_{-j})$ for any j>i : an observation from later in the sequence has absolutely no effect on the probability of the current state



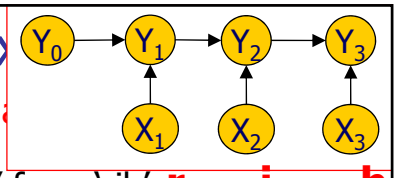
HMM



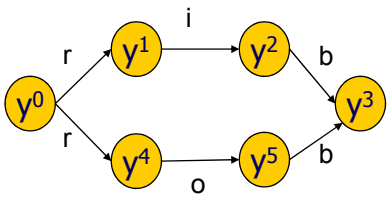
MEMM

10

Label-Bias Problem: Ex



- Weakness: current label is not a observation
- A model for distinguishing 'rob' from 'rib' **r i b**
- Suppose we get an input sequence **X**='rib'
  - First step, 'r' matches both possible states equally likely
  - Next, 'i' is observed, but since both y^1 and y^4 have one outgoing state, they both give probability 1 to the next state
  - Note: if one word is more likely in train data, it will win
  - Does not happen in HMMs



State transition representation

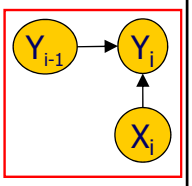
P(Y'|Y)

	y^0	y^1	y^2	y^3	y^4	y^5
y^0	0	0.5	0.5	0	0	0
y^1	0	0	**1**	0	0	0
y^2	0	0	0	1	0	0
y^3	0	0	0	1	0	0
y^4	0	0	0	0	0	**1**
y^5	0	0	0	1	0	0

11

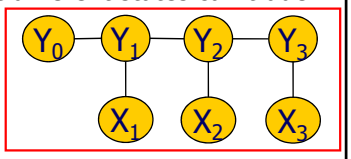
Conditional Random Fields

MEMM



- Advantages of MEMMs without the label bias problem
- Key difference
  - MEMMs use per-state model for conditional probabilities of next state given current state
  - CRFs have a single model for the joint probability of the entire sequence of labels given the observations
    - Thus, weights of different features at different states can trade off against each other

CRF



- CRF training
  - Maximum likelihood estimation or MAP (a little later)
  - Objective function is concave, guaranteeing convergence to global optimum

12

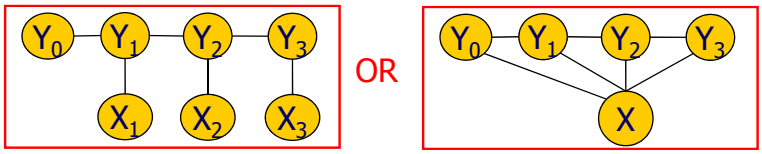
Conditional Random Fields

- Let G=(V,E) be a graph with vertices V and edges E, such that $\boldsymbol{Y} = (\boldsymbol{Y}_v)_{v \in V}$
- Then (**X**,**Y**) is a CRF if the random variables $\mathbf{Y}_v$ obey the Markov property with respect to the graph:

$$P(\mathrm{Y}_v \mid \mathrm{X}, \mathrm{Y}^i, i \neq v) = P(\mathrm{Y}_v \mid \mathrm{X}, \mathrm{Y}^j, i \sim v)$$

  - where $\mathbf{Y}^j$ is the set of Y neighbors of Y^i
- And if it models only P(**Y**|**X**)

CRF



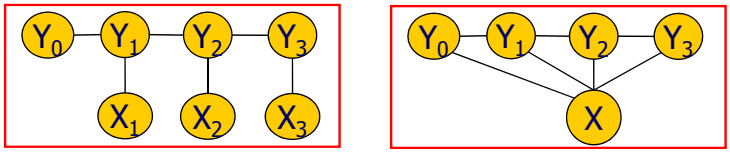
Conditional Random Fields

- Joint probability distribution for trees over **Y**
  - Cliques (and thus potentials) are the edges and vertices

$$p_\theta(\boldsymbol{y} \mid \boldsymbol{x}) \propto \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, \boldsymbol{y}[e], \boldsymbol{x}) + \sum_{v \in V,k} \mu_k g_k(v, \boldsymbol{y}[v], \boldsymbol{x}) \right)$$

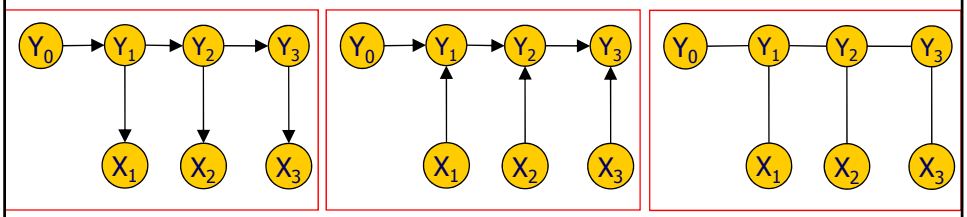
  - **x** are the observed variables
  - **y** are the state variables
  - **y**[S] is the components of **Y** associated with vertices in S
  - f_k is an edge feature with weight λ_k
  - g_k is a vertex feature with weight μ_k
  - Note that features can be over all of variables in **x**

CRF



Comparison 1/3

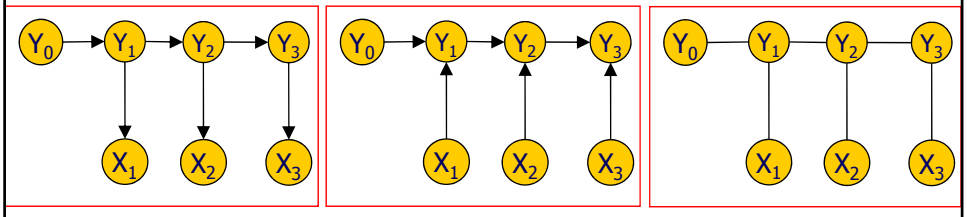
- Computational perspective
  - Purely directed models – HMMs and MEMMs – are much more easily learned
  - Their parameters can be computed in closed form using MLE or Bayesian estimation
  - CRF requires an iterative gradient-based approach and inference must be run for every training instance



15

Comparison 2/3

- Ability to use a rich feature set
  - Success in a classification task often depends strongly on the quality of our features
  - In an HMM, we must explicitly model the distribution over features **X**, including the interactions between them
    - Depending on features, this type of model is very hard and often impossible to construct correctly
  - MEMM, CRF are both discriminative models and so they avoid this challenge entirely



16

Comparison: Summary

- Independence assumptions made by the model
  - In MEMMs, $(Y_i \perp X_j | \mathbf{X}_{-j})$ for any j>i : current label is not affected by the future observation (label bias problem)
- Summary
  - In cases where there are many correlated features, discriminative models are probably better
  - If only limited data are available, the stronger bias of the generative model (modeling P(**X**)) may dominate and allow learning with fewer samples
  - Among the discriminative models, MEMMs should probably be avoided in cases where many transitions are close to deterministic (label bias problem)
  - In many cases, CRFs are likely to be a safer choice, but the computational cost may be prohibitive for large datasets

17

Learning Undirected Graphs

- The likelihood function
- Learning parameters
- Collective classification with HMM, MEMM, CRF
  - Generative vs. discriminative models
  - Directed vs. undirected models
- Learning with incomplete data
- Learning with Priors
  - Maximum A Priori (MAP) estimation
- Learning with alternative objectives
  - Pseudo likelihood objective
  - Max-margin learning
- Structure Learning

18

Learning with Missing Data

- In MLE with complete data, the gradient is

$$\frac{\partial}{\partial \theta_i} l(\boldsymbol{\theta} : D) = M\mathbf{E}_D[\, f_i[\mathbf{d}_i]] - M\mathbf{E}_{\boldsymbol{\theta}}[\, f_i]$$

Number of times feature f_i is true in data D

Expected number of times feature f_i is true according to model

- Gradient of likelihood is now difference of expectations
  - Y: hidden, X: observed

$$\frac{\partial}{\partial \theta_i} l(\boldsymbol{\theta} : D) = M\mathbf{E}_{\boldsymbol{\theta}}[\, f_i[\mathbf{y}_i \mid \mathbf{x}_i]] - M\mathbf{E}_{\boldsymbol{\theta}}[\, f_i]$$

Expected number of times feature f_i is true given observed data

Expected number of times feature f_i is true according to model

- Can use gradient descent or EM

19

Learning Undirected Graphs

- The likelihood function
- Learning parameters
- Collective classification with HMM, MEMM, CRF
  - Generative vs. discriminative models
  - Directed vs. undirected models
- Learning with incomplete data
- Learning with Priors
  - Maximum A Priori (MAP) estimation
- Learning with alternative objectives
  - Pseudo likelihood objective
  - Max-margin learning
- Structure Learning

20

Maximum A Priori (MAP) estimation

- Introducing a prior distribution $P(\theta)$ over the model parameters
- Bayesian approach
  - Given D={**x**[1],…,**x**[M]},

$$P(\mathbf{x}[M+1] \mid D) = \int_{\theta} P(\mathbf{x}[M+1] \mid \theta) P(\theta \mid D) d\theta$$

- Maximum a Priori (MAP) estimation

$$\arg\max_{\theta} P(\theta \mid D) = \arg\max_{\theta} P(\theta) P(D \mid \theta)$$

- Maximum likelihood estimation (MLE)

$$\arg\max_{\theta} P(D \mid \theta)$$

21

Gaussian Prior

- MAP estimation

$$\arg\max_{\theta} P(\theta \mid D) = \arg\max_{\theta} P(\theta) P(D \mid \theta)$$

$$\log P(\theta \mid D) = \log P(D \mid \theta) + \log P(\theta)$$

- Gaussian prior

$$P(\boldsymbol{\theta} \mid \sigma^2) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{\theta_i^2}{2\sigma^2} \right\}$$

  - Converting to log-space $-\frac{1}{2\sigma^2} \sum_{i=1}^{k} \theta_i^2$
  - L2 regularization

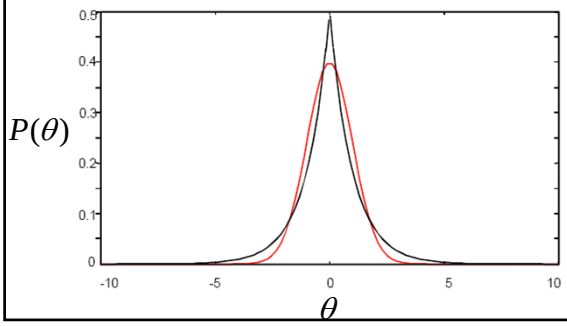
22

Laplacian Prior

- Laplacian prior

$$P_{\text{Laplacian}}(\boldsymbol{\theta} \mid \beta) = \prod_{i=1}^{k} \frac{1}{2\beta} \exp\left\{-\frac{|\theta_i|}{\beta}\right\}$$

  - Converting to log-space $-\frac{1}{\beta}\sum_{i=1}^{k}|\theta_i|$
  - L1 regularization



Laplacian distribution (β=1) and Gaussian distribution (σ^2=1)

23

Why Regularization?

- Both forms of regularization penalize parameters whose magnitude is large
  - Why is a bias in favor of parameters of low magnitude a reasonable one?
  - A prior often serves to pull the distribution toward an "uninformed" one, smoothing out fluctuations in the data

- A distribution is "smooth" if the probabilities assigned to different assignments are not radically different.
  - Consider two assignments ξ and ξ'
  - Log of their relative probability is

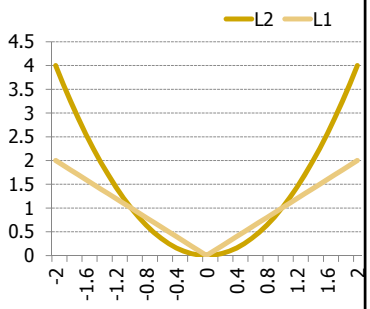
$$\ln \frac{P(\xi)}{P(\xi')} = \sum_{i=1}^{k} \theta_i f_i(\xi) - \sum_{i=1}^{k} \theta_i f_i(\xi') = \sum_{i=1}^{k} \theta_i \left(f_i(\xi) - f_i(\xi')\right)$$

  - When all Θ's have small magnitude, this log-ratio is also bounded, resulting in a smooth distribution.

24

L1 vs L2 Regularization

- Gaussian prior (L2): $-\frac{1}{2\sigma^2}\sum_{i=1}^{k}\theta_i^2$
- Laplacian prior (L1): $-\frac{1}{\beta}\sum_{i=1}^{k}|\theta_i|$



- Key differences:
  - In L2, the penalty grows quadratically with the parameter magnitude.
  - In L1, the penalty is linear in the parameter magnitude.
  - In L2, as the parameters get close to 0, the effect of the penalty diminishes, whereas in L1 case, the penalty is linear all the way until the parameter value is 0.
- The models learned with an L1 regularization tend to be much sparser than the L2 case.
  - The strength depends on the hyper-parameter β

25

Learning Undirected Graphs

- The likelihood function
- Learning parameters
- Collective classification with HMM, MEMM, CRF
  - Generative vs. discriminative models
  - Directed vs. undirected models
- Learning with incomplete data
- Learning with Priors
  - Maximum A Priori (MAP) estimation
- Learning with alternative objectives
  - Pseudo likelihood objective
  - Max-margin learning
- Structure Learning

26

Why Alternative Objectives?

- The log-likelihood objective, on the case of a single data instance ξ

$$l(\boldsymbol{\theta}:\xi) = \ln \tilde{P}(\xi \mid \boldsymbol{\theta}) - \ln Z(\boldsymbol{\theta}) = \ln \tilde{P}(\xi \mid \boldsymbol{\theta}) - \ln\left(\sum_{\xi'} \tilde{P}(\xi' \mid \boldsymbol{\theta})\right)$$

  - MLE can be viewed as aiming to increase the distance between the log of the un-normalized probability (log-measure) of ξ and the aggregate of the measures of all instances.
- Key difficulty: the 2nd term involves a summation over the exponentially many instances in Val(**X**).
  - In MLE, we have to compute the log-likelihood in every iteration (approximate inference)
- Alternative objectives
  - Aim to increase the difference between the log-measure of the data instance and a more tractable set of other instances ("Contrastive" objectives)

27

Pseudo-likelihood

- For a data instance ξ, using the chain rule, we can write

$$P(\xi) = \prod_{j=1}^{n} P(x_j \mid x_1, \ldots, x_{j-1})$$

- We can approximate this formulation by replacing each term by the conditional probability x_j given all other variables $\mathbf{x}_{-j}$

$$P(\xi) \approx \prod_{j=1}^{n} P(x_j \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$$

- This approximation leads to the pseudolikelihood objective: Given D with M training instances,

$$l_{PL}(\boldsymbol{\theta}:D) = \frac{1}{M} \sum_{\text{Each instance } m} \sum_{\text{Each variable } j} \ln P(x_j[m] \mid \mathbf{x}_{-j}[m], \boldsymbol{\theta})$$

28

Gradient of Pseudolikelihood

- Pseudolikelihood objective:

$$l_{PL}(\boldsymbol{\theta}:D)=\frac{1}{M}\sum_{\text{Each instance } m}\sum_{\text{Each variable } j}\ln P(x_j[m]\,|\,\mathbf{x}_{-j}[m],\boldsymbol{\theta})$$

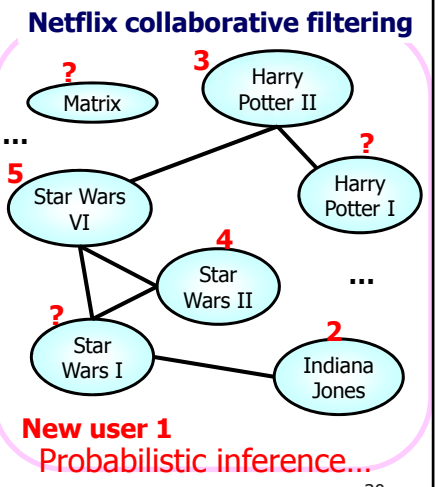
- Each term is a log-conditional likelihood term over a single var X_j, conditional on all the remaining vars

$$\ln P(x_j\,|\,\mathbf{x}_{-j})=\left(\sum_{i:\mathrm{X}_j\in\mathrm{Scope}[\mathrm{f}_i]}\theta_i\mathrm{f}_i[x_j,\mathbf{u}_j]\right)-\ln\left(\sum_{x_j'}\exp\left\{\sum_{i:\mathrm{X}_j\in\mathrm{Scope}[\mathrm{f}_i]}\theta_i\mathrm{f}_i[x_j',\mathbf{u}_j]\right\}\right)$$

  - The 2nd term involves a summation over values on only a single var X_j (does not require inference at each step)
- Widely used in vision, spatial statistics, etc.
- Jointly concave over all parameters
- Consistent estimator
  - As the number of data instances M goes to infinity, with probability 1, MLE of the log-likelihood objective Θ* (the true parameter) is a global optimum of the pseudolikelihood objective

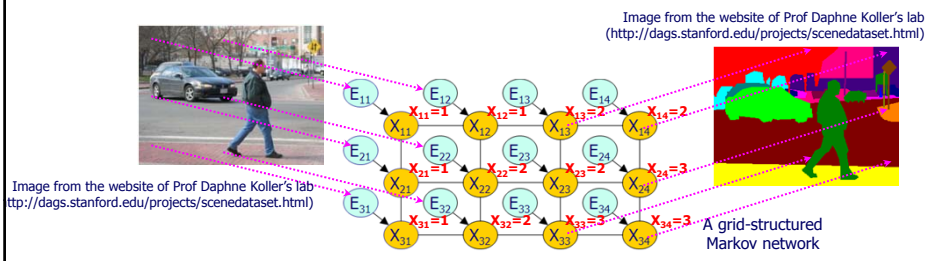
Pseudolikelihood vs Likelihood

- When the pseudolikelihood does not work well?
  - Depends on the types of queries for which we intend to use the model
- Psuedolikelihood objective is a better training objective
  - If we plan to run queries where we condition on most of the variables and query the values of only a few, the pseudolikelihood objective is a very close match to the type of predictions we would like to make
  - Any example?



Pseudolikelihood vs Likelihood

- Likelihood objective is a better training objective
  - If a typical query involves most or all of the variables in the model, the likelihood objective is more appropriate.
  - E.g. Given **E**=image what is P(**X**=labels|**E**=image)=?



- However, even in cases where the likelihood is the more appropriate objective, we may have to resort to pseudolikelihood for computational reasons
  - In many cases, this objective performs surprisingly well.

31

Max-margin Training

- Say that we want to use the model for predicting a MAP assignment
  - (E.g. image segmentation)
  - In this setting, our training set consists of a set of pairs $D=\{(\mathbf{y}[m],\mathbf{x}[m])\}_{m=1,\ldots,M}$.
  - Given an observation $\mathbf{x}[m]$, we want our learned model to give the highest probability to $\mathbf{y}[m]$.
  - In other words, we want the probability $P_{\theta}(\mathbf{y}[m]|\mathbf{x}[m])$ to be higher than any other probability $P_{\theta}(\mathbf{y}|\mathbf{x}[m])$ for $\mathbf{y}\neq\mathbf{y}[m]$.
- To increase our confidence in the prediction, we would like to increase the log-probability gap as much as possible by increasing

$$\ln P_{\theta}(\mathbf{y}[m]\,|\,\mathbf{x}[m]) - \left[\max_{\mathbf{y}\neq\mathbf{y}[m]} \ln P_{\theta}(\mathbf{y}\,|\,\mathbf{x}[m])\right]$$

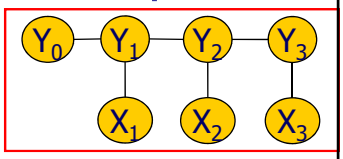
  - This difference between the log-probability of the target assignment $\mathbf{y}[m]$ and that of the "next best" assignment is called the margin.
  - The higher the margin, the more confidence the model is

32

Handwriting Recognition Example

- Margin

$$\ln P_{\theta}(\mathbf{y}[m] \mid \mathbf{x}[m]) - \left[\max_{\mathbf{y} \neq \mathbf{y}[m]} \ln P_{\theta}(\mathbf{y} \mid \mathbf{x}[m]) \right]$$



CRF

- We want:

$\arg\max_{\mathbf{y}} \boldsymbol{\theta}^{\top} \mathbf{f}(\text{  }, \mathbf{y}) = \text{"brace"}$

- Equivalently:

$\boldsymbol{\theta}^{\top} \mathbf{f}(\text{  }, \text{"brace"}) > \boldsymbol{\theta}^{\top} \mathbf{f}(\text{}, \text{"aaaaa"})$

$\boldsymbol{\theta}^{\top} \mathbf{f}(\text{  }, \text{"brace"}) > \boldsymbol{\theta}^{\top} \mathbf{f}(\text{}, \text{"aaaab"})$

...

$\boldsymbol{\theta}^{\top} \mathbf{f}(\text{}, \text{"brace"}) > \boldsymbol{\theta}^{\top} \mathbf{f}(\text{}, \text{"zzzzz"})$

a lot!

33

Learning Undirected Graphs

- The likelihood function
- Learning parameters
- Collective classification with HMM, MEMM, CRF
  - Generative vs. discriminative models
  - Directed vs. undirected models
- Learning with incomplete data
- Learning with Priors
  - Maximum A Priori (MAP) estimation
- Learning with alternative objectives
- Structure Learning
  - Structure learning via L1 regularization

37

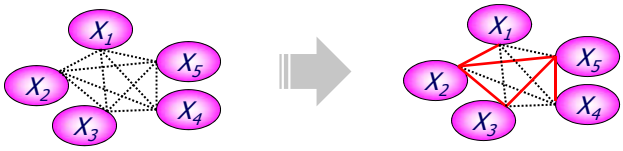
Structure Learning

- Start with atomic features
- Greedily conjoin features to improve score
- Problem: Need to re-estimate weights for each new candidate
- Approximation: Keep weights of previous features constant

38

Structure Learning via Regularization*

- Treat the structure learning problem as a parameter estimation problem in a fully connected network
- L1 regularization to obtain a sparse representation



- Likelihood or pseudolikelihood objective
- Convex optimization problem

*Lee 07, Wainwright 07, Hoefling 09

39