CONDITIONAL MODELS

COMP 4680/8650
Justin Domke
comp8650@anu.edu.au

Reading: Murphy, §19.6, 9.3-9.4

Conditional Models

- Suppose we have some exponential family model of an input x and and output y:

$$p(x, y|\theta) = \exp\left(\theta^T \phi(x, y) - A(\theta)\right)$$

$$A(\theta) = \log \textstyle\sum_{x,y} \exp(\theta^T \phi(x, y))$$

- We want to use it to predict y given x. Write conditional distribution as

$$p(y|x;\theta) = \exp\left(\theta^T \phi(x, y) - A(x;\theta)\right)$$

$$A(x;\theta) = \log \textstyle\sum_{y} \exp(\theta^T \phi(x, y))$$

- Given a dataset $(x^1, y^1), (x^2, y^2), ..., (x^D, y^D)$ how do we learn θ and how do we do inference?

Conditional Models

$$p(y|x;\theta) = \exp\left(\theta^T \phi(x,y) - A(x;\theta)\right)$$
$$A(x;\theta) = \log \sum_y \exp(\theta^T \phi(x,y))$$

- Traditional approach:
  - Fit θ to maximize $\frac{1}{D}\sum_{d=1}^{D} \log p(x^d, y^d|\theta)$
  - At test time, do inference using $p(y|x;\theta)$
- Fine, but…
  - We are learning a distribution over x that we never actually use!
- New approach:
  - Fit θ to maximize $\frac{1}{D}\sum_{d=1}^{D} \log p(y^d|x^d;\theta)$
  - At test time, do inference using $p(y|x;\theta)$

Conditional Models

$$\frac{1}{D}\sum_{d=1}^{D}\log p(y^d|x^d;\theta) \text{ vs. } \frac{1}{D}\sum_{d=1}^{D}\log p(x^d,y^d|\theta)$$

- We can always write that:

$$\frac{1}{D}\sum_{d=1}^{D}\log p(x^d,y^d|\theta) = \frac{1}{D}\sum_{d=1}^{D}\log p(y^d|x^d;\theta) + \frac{1}{D}\sum_{d=1}^{D}\log p(x^d|\theta)$$

- Switching from joint to conditional likelihood just means dropping this term.

Model Specification

$$\frac{1}{D}\sum_{d=1}^{D}\log p(x^d, y^d|\theta) = \frac{1}{D}\sum_{d=1}^{D}\log p(y^d|x^d;\theta) + \frac{1}{D}\sum_{d=1}^{D}\log p(x^d|\theta)$$

- If the model is not well-specified the conditional likelihood will tend to be better, given enough data.

- Joint likelihood converges to

$$\arg\min_\theta KL(p_0(x,y)||p(x,y|\theta)) = -\sum_{x,y} p_0(x,y)\log\frac{p_0(x,y)}{p(x,y|\theta)}$$

- Conditional likelihood converges to

$$\arg\min_\theta KL(p_0(y|x)\ ||\ p(y|x;\theta)) = -\sum_{x} p_0(x,y)\log\frac{p_0(y|x)}{p(y|x;\theta)}$$

Partition-Function Computation

- Computing the joint likelihood requires

$$A(\theta) = \log \sum_{x,y} \exp(\theta^T \phi(x, y))$$

  or sometimes

$$A(\theta) = \log \int_x \sum_y \exp(\theta^T \phi(x, y)) dx$$

- The conditional likelihood only requires

$$A(x; \theta) = \log \sum_y \exp(\theta^T \phi(x, y))$$

Over-Fitting

- With a well-specified model, the joint likelihood will tend to over-fit less.

- Why? Both of these are minimized by true θ.

$$\frac{1}{D}\sum_{d=1}^{D} \log p(y^d | x^d; \theta)$$

$$\frac{1}{D}\sum_{d=1}^{D} \log p(x^d | \theta)$$

- With finite data, you face a trade-off.

Clamped log-partition derivatives

$$
\begin{aligned}
A(x;\theta) &= \log \sum_y \exp(\theta^T \phi(x,y)) \\
\frac{dA(x;\theta)}{d\theta} &= \frac{1}{\sum_y \theta^T \phi(x,y)} \sum_y \frac{d}{d\theta} \exp(\theta^T \phi(x,y)) \\
&= \frac{1}{\sum_y \theta^T \phi(x,y)} \sum_y \exp(\theta^T \phi(x,y)) \frac{d}{d\theta} \theta^T \phi(x,y) \\
&= \frac{1}{\sum_y \theta^T \phi(x,y)} \sum_y \exp(\theta^T \phi(x,y)) \phi(x,y) \\
&= \sum_y p(y|x;\theta) \phi(x,y)
\end{aligned}
$$

Conditional Moment Matching

$$\frac{dL}{d\theta} = \frac{1}{D}\sum_{d=1}^{D}\frac{d}{d\theta}\log p(y^d|x^d;\theta)$$

$$= \frac{1}{D}\sum_{d=1}^{D}\frac{d}{d\theta}(\theta^T\phi(x^d,y^d) - A(x^d;\theta))$$

$$= \frac{1}{D}\sum_{d=1}^{D}\left(\phi(y^d,x^d) - \sum_{y} p(y|x^d;\theta)\phi(x^d,y)\right)$$

$$= \hat{\mathbb{E}}_{X,Y}\phi(Y,X) - \hat{\mathbb{E}}_X\mathbb{E}_{p(Y|X)}\phi(Y,X)$$

Must do inference D times!

Joint vs. Conditional Likelihood

- Advantages of conditional likelihood:
  - With infinite data, at least as good. (Better if mis-specified)
  - Only need to compute $dA(x^d;\theta)/d\theta$ instead of $dA(\theta)/d\theta$.

- Advantage of joint likelihood:
  - Better generalization with finite data.
  - Only need to compute $dA(\theta)/d\theta$ once, rather than $dA(x^d;\theta)/d\theta$ for each datum.

- Alas, the real world tends to have finite data and mis-specified models.

Conditional Random Fields

- A conditional undirected model.

$$p(y|x,w) = \frac{1}{Z(x,w)} \prod_c \psi_c(y_c|x,w)$$

- Typically have log-linear potentials

$$\psi_c(y_c|x,w) = \exp(w_c^T \phi(x,y_c))$$

  - Can be seen as a conditional exponential family

$$p(y|x,w) = \exp(w^T \phi(x,y) - A(x,w))$$

$$\phi(x,y) = \{\phi(x,y_c) \forall c\}$$

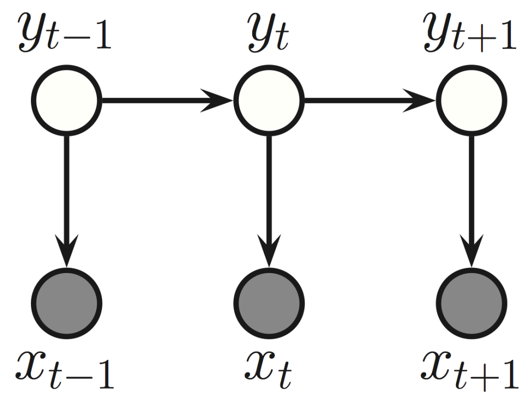
Hidden Markov Models

- Consider modeling an input sequence with a corresponding output sequence

$$(x_1, x_2, \ldots, x_T) \qquad (y_1, y_2, \ldots, y_T)$$

Traditionally done with a hidden Markov model.

$$p(x, y|w) = \prod_{t=1}^{T} p(y_t|y_{t-1}, w)p(x_t|y_t, w)$$

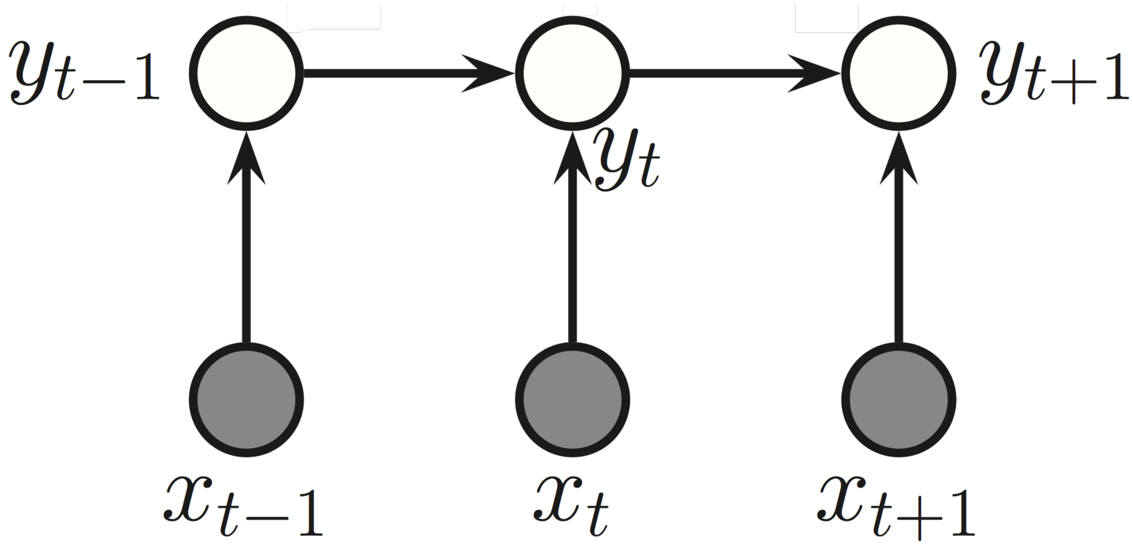


What if we don't want to model x?

Maximum Entropy Markov Models

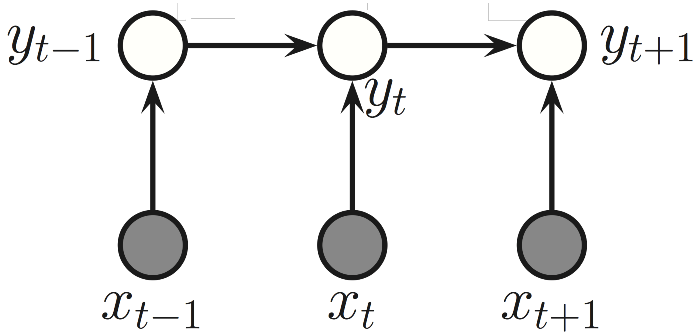
- Simplest Solution: Reverse the arrows!

$$p(y|x, w) = \prod_{t=1}^{T} p(y_t | y_{t-1}, x_t, w)$$



Note: The book has different (worse?) notation for this

The Label-Bias Problem

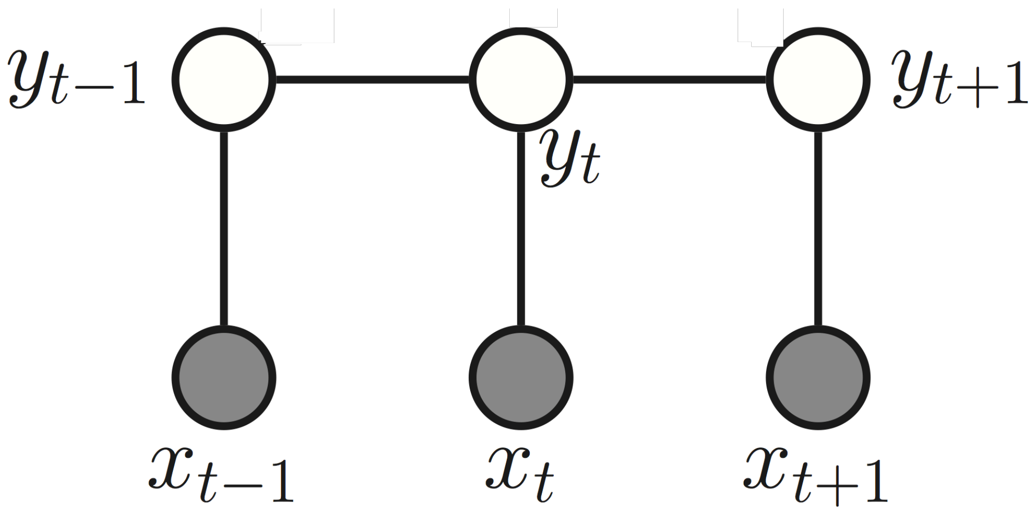


- Maximum entropy Markov models are little used because of the "label-bias" problem.
  - x_{t+2} cannot influence y_t .

Really, just a form of model-misspecification. Conditional independencies asserted by a MEMM don't hold in many applications.

Chain Structured CRF

$$p(y|x,w) = \frac{1}{Z(x,w)} \prod_{t=1}^{T} \psi(y_t|x,w) \prod_{t=1}^{T-1} \psi(y_t, y_{t+1}|x,w)$$



Global normalization allows later features to influence earlier ones.

Summary of models

- Hidden Markov Model (Ancient history – 2000)

$$p(x, y|w) = \prod_{t=1}^{T} p(y_t|y_{t-1}, w)p(x_t|y_t, w)$$

- Maximum Entropy Markov Model (2000 – 2001)

$$p(y|x, w) = \prod_{t=1}^{T} p(y_t|y_{t-1}, x_t, w)$$

- Conditional Random Field (2001 – present)

$$p(y|x, w) = \frac{1}{Z(x, w)} \prod_{t=1}^{T} \psi(y_t|x, w) \prod_{t=1}^{T-1} \psi(y_t, y_{t+1}|x, w)$$

Handwriting Recognition

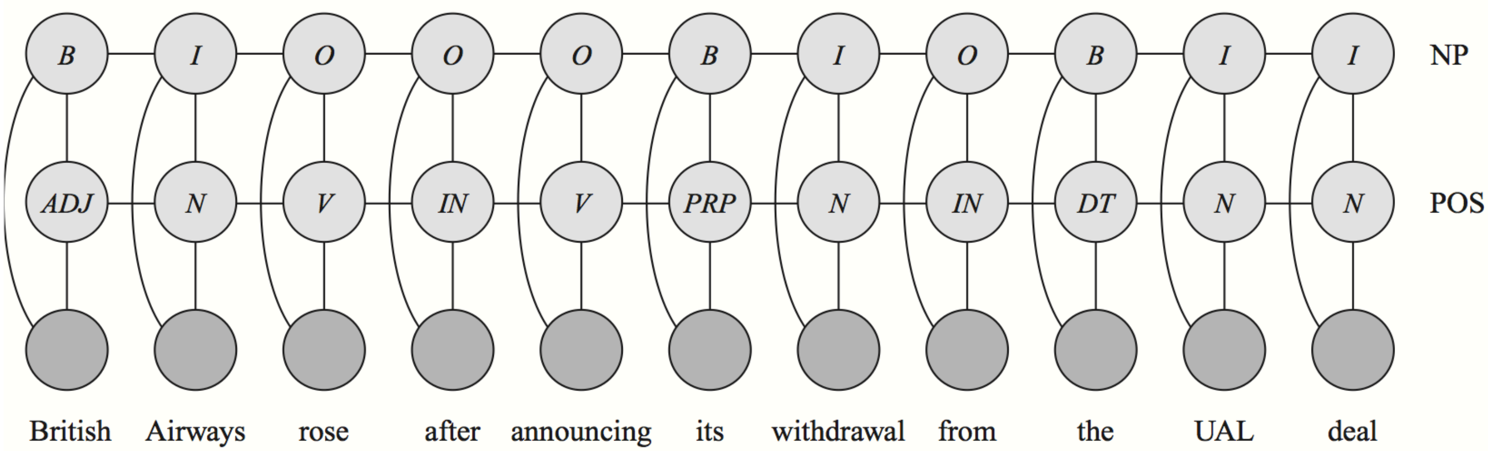
$$p(y|x,w) = \frac{1}{Z(x,w)} \prod_{t=1}^{T} \psi(y_t|x,w) \prod_{t=1}^{T-1} \psi(y_t, y_{t+1}|x,w)$$

Typically, $\psi(y_t|x,w)$ would be a probabilistic classifier, e.g. a neural network.

Noun Phrase Chunking

```
   B      I          O     O      O          B    I         O    B   I   I
(British Airways) rose after announcing (its withdrawl) from (the UAI deal)
```

Standard approach: convert each word into a POS, then convert POS tags into Noun Phrases. **Problem**: Errors propagate.

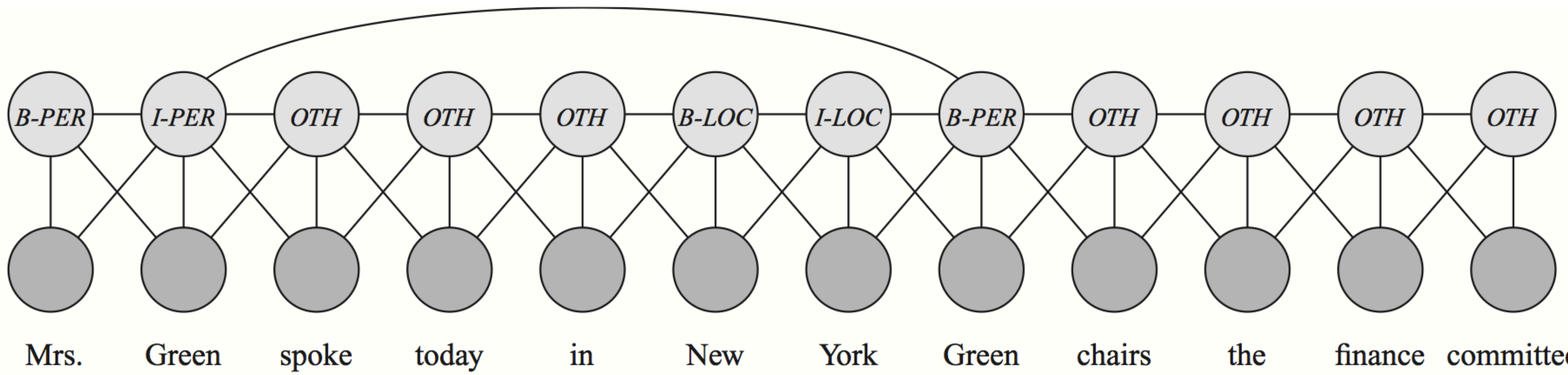


KEY

B	Begin noun phrase	*V*	Verb
I	Within noun phrase	*IN*	Preposition
O	Not a noun phrase	*PRP*	Possesive pronoun
N	Noun	*DT*	Determiner (e.g., a, an, the)
ADJ	Adjective		

Named-Entity Recognition

- Distinguish person vs. location entities



KEY

B-PER	Begin person name	*I-LOC*	Within location name
I-PER	Within person name	*OTH*	Not an entitiy
B-LOC	Begin location name		

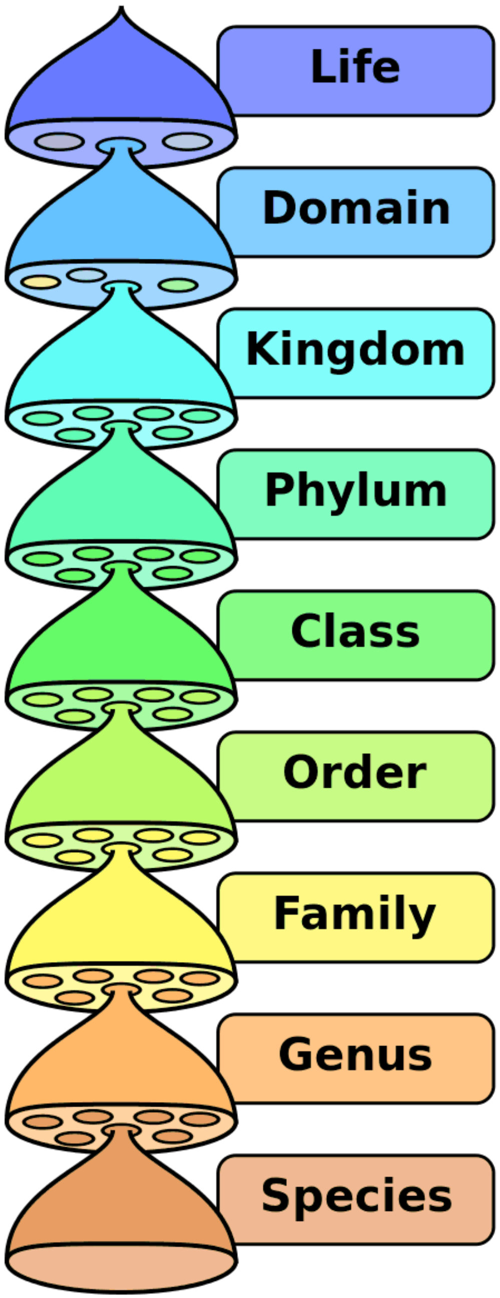
Long-range connections: "skip chain" CRF

- Typically use 1,000-10,000 features per node. Cost?

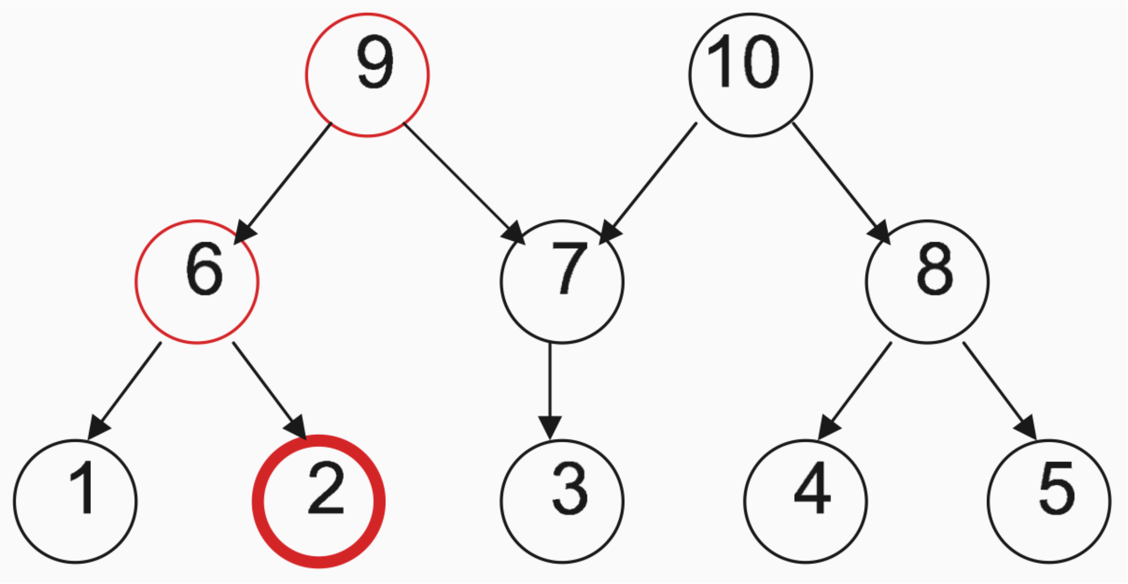
Hierarchical Classification

- Suppose we want to classify an organism's species.
  - Harder the further we go down.
- Idea: create $\phi(y)$ with one component for each domain/phylum/species. Values are positive for all relevant categories.

$$\phi(x, y) = \phi(x) \otimes \phi(y)$$



Hierarchical Classification



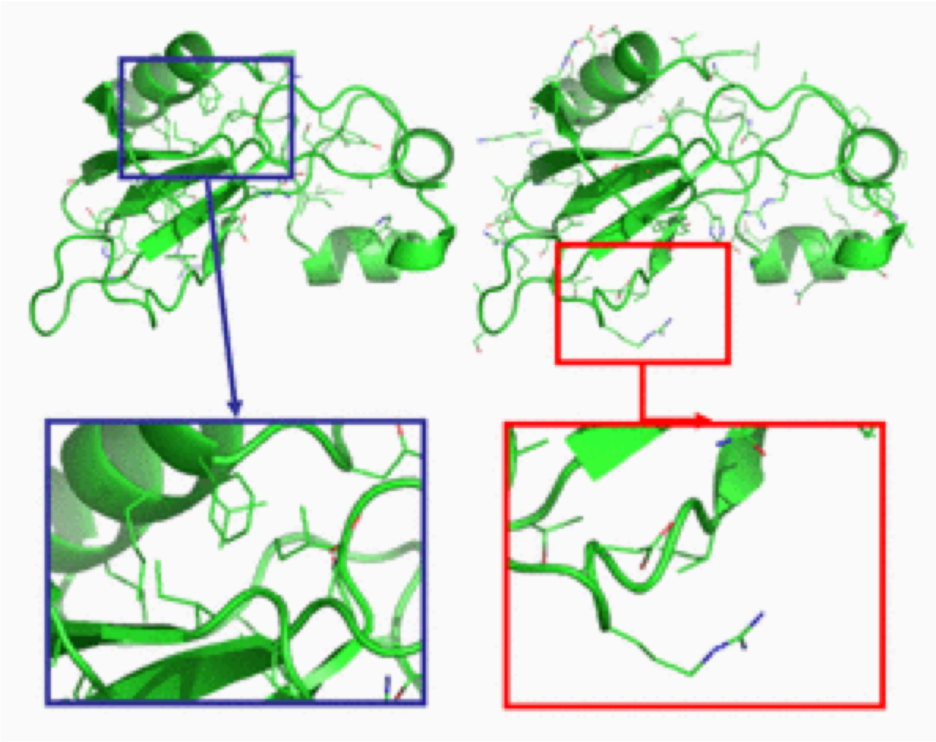
$$\Lambda(2) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad \Psi(\mathbf{x}, 2) = \begin{pmatrix} 0 \\ \mathbf{x} \\ 0 \\ 0 \\ 0 \\ \mathbf{x} \\ 0 \\ 0 \\ \mathbf{x} \\ 0 \end{pmatrix}$$

$$\langle \mathbf{w}, \Psi(\mathbf{x}, 2) \rangle = \langle w_2, \mathbf{x} \rangle + \langle w_6, \mathbf{x} \rangle + \langle w_9, \mathbf{x} \rangle$$

Protein Side-Chain Prediction

- Input series of amino acids x.
- Want to predict discrete sequence of angles y.

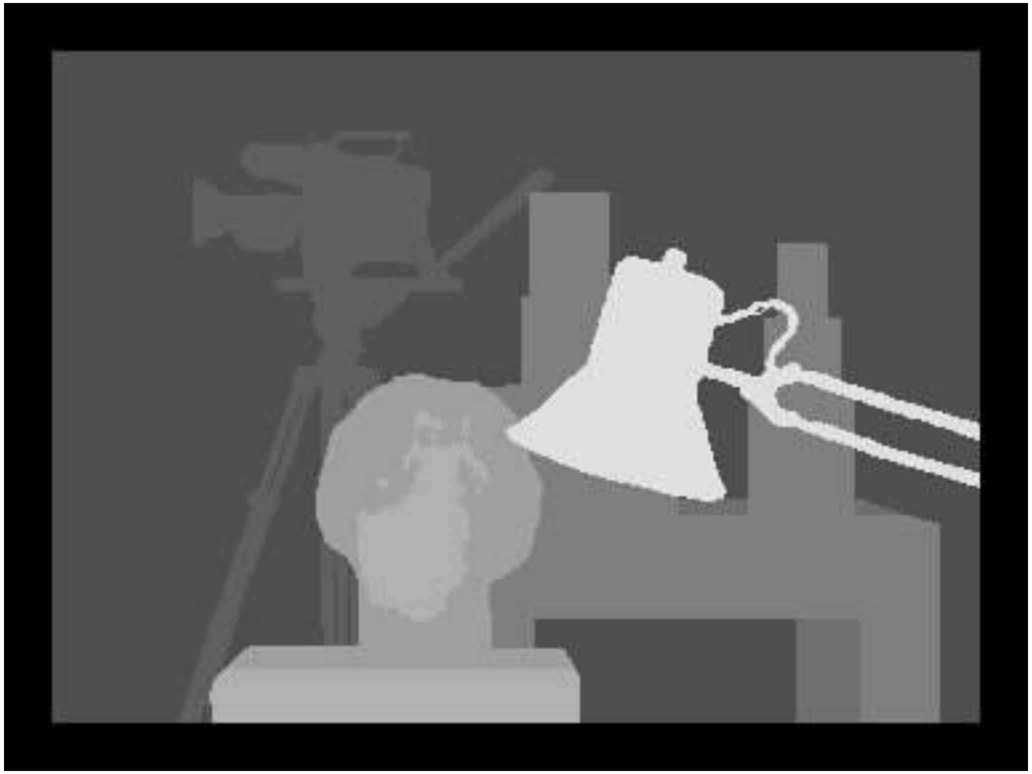
$$E(x, y, \theta) = \sum_{j=1}^{D} \theta_j E_j(x, y)$$



- Different E_j model different:
  - Electrostatic charges
  - Hydrogen bonding potentials
  - Etc.
- Also a skip-chain.

Stereo Vision

Stereo Vision



Stereo Vision

- Given a pair of images x, want to predict disparities y.

$$p(y|x) \propto \prod_s \psi(y_s|x) \prod_{st} \psi_{st}(y_s, y_t)$$

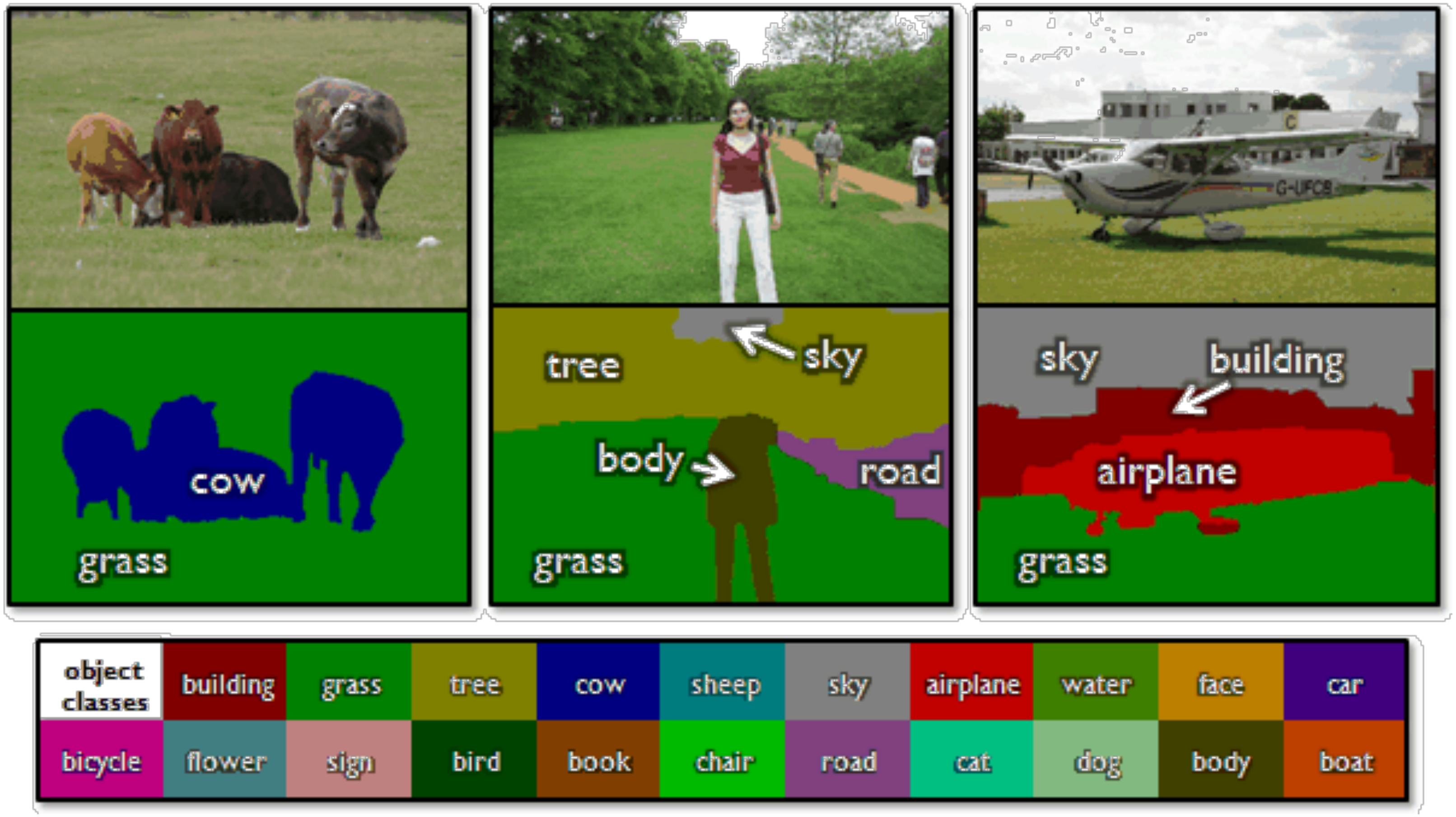
- First term ensures consistency with images:

$$\psi_s(y_s|x) \propto \exp\left(\frac{1}{2\sigma^2}(x_L(i_s, j_s) - x_R(i_s + y_s, j_s))^2\right)$$

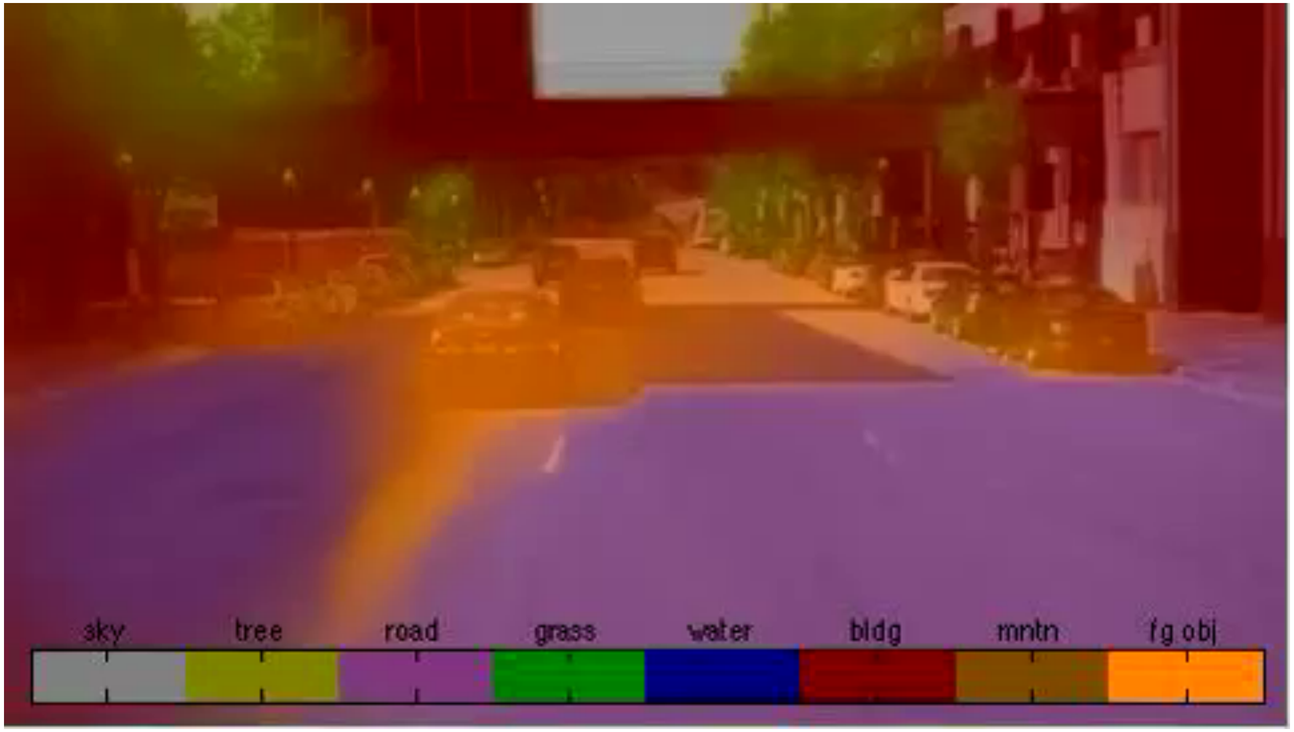
- Second term ensures smoothness:

$$\psi_{st}(y_s, y_t) \propto \exp\left(-\frac{1}{2\gamma^2}\min((y_s - y_t)^2, \delta_0^2\right)$$

Semantic Segmentation



Semantic Segmentation



Semantic Segmentation

- Given an image x, want to predict set of labels y.

$$p(y|x;w) \propto \prod_s \psi_s(y_s|x) \prod_{st} \psi_{s,t}(y_s, y_t|x)$$

- First term models "how much does pixel s locally look like class y_s"?
- Second term models "how much does class y_s like to be above class y_t"?
  - Not symmetric
  - Depends on x!

What to take home

- Conditional exponential family (CEF).
- Conditional Random Field (CRF) as CEFs.
- In a CRF, derivative of log-partition function is conditional marginals.
- Tradeoffs between “bias” and “variance” in generative vs. discriminative models.
- CRFs have lots of applications.