# Structured Learning

Jun Zhu

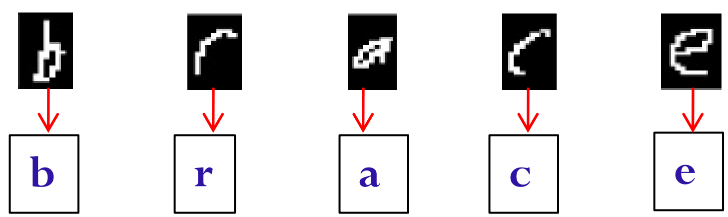
# Supervised learning

- Given a set of I.I.D. training samples $\mathcal{D} = \{(\mathrm{x}^i, y^i)\}_{i=1}^N$

$$\mathrm{x}^i = (x_1^i, x_2^i, \cdots, x_d^i)^\top \qquad y^i \in C \triangleq \{c_1, c_2, \cdots, c_L\}$$

- Learn a prediction function

$$h: \mathcal{X} \rightarrow \mathcal{Y}$$



# Supervised learning (cont'd)

- Many different choices

  – Logistic Regression
    - Maximum likelihood estimation

$$\max_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{w}) \triangleq \sum_{i=1}^{N} \log p(y^i | \mathrm{x}^i)$$

$$p(y|\mathrm{x}) = \frac{\exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x}, y)\}}{\sum_{y'} \exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x}, y')\}}$$

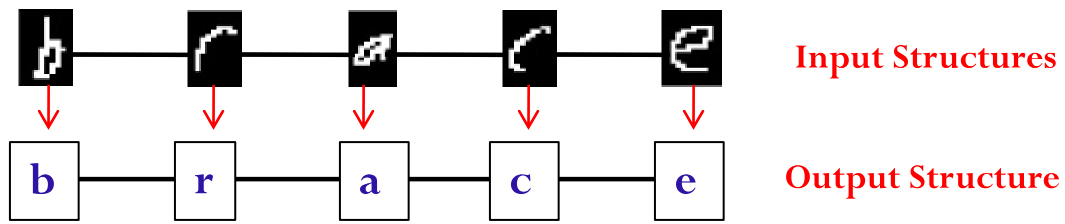
  – Support Vector Machines (SVM)
    - Max-margin learning

$$\min_{\mathbf{w}, \xi} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^{N} \xi_i;$$

$$\text{s.t.} \quad \mathbf{w}^\top \Delta \mathbf{f}_i(y) \geq 1 - \xi_i, \ \forall i, \forall y \neq y^i.$$

# Real problems usually come with structures

- OCR – sequence




- Image annotation – regular/irregular 2D layout

- Much richer structures are not uncommon…

# Structured learning

- A set of models, learning methods and theories to consider structured inputs and/or structured outputs and or structured models

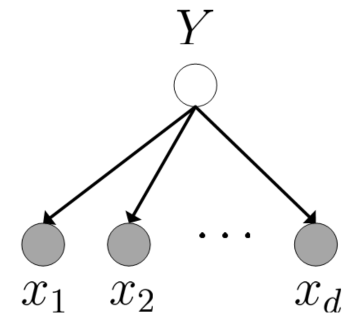
- Learning with structured outputs come with various names
  - Structured output learning
  - Structured prediction
  - Collective prediction/classification
  - Relational learning
  - …

- We don't discuss model structures
  - Sparsity, structured sparsity, hierarchical models, etc.

# Structured inputs

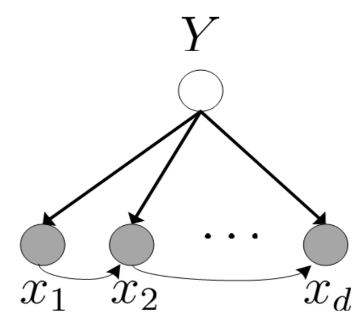
- Naïve Bayes (generative models)
  - Strict conditional independence assumption on inputs

$$p(x_1, \ldots, x_d | y) = \prod_{i=1}^{d} p(x_i | y)$$



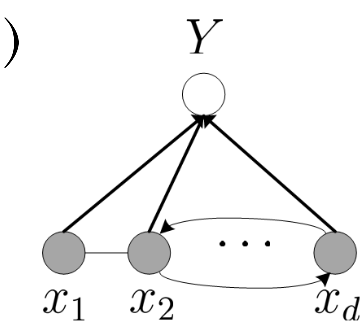

- Tree-augmented NB (generative models)
  - Introduce sparse edges between input variables

$$p(x_1, \ldots, x_d | y) = p(x_1 | y) \prod_{i=2}^{d} p(x_i | x_{i-1}, y)$$




- Logistic regression (conditional/discriminative models)
  - Allow arbitrary structures in inputs

$$p(y | \mathrm{x}) = \frac{\exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x}, y)\}}{\sum_{y'} \exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x}, y')\}}$$




**Discriminative SVM deals with rich input structures using kernels**

# Conditional vs. Generative

- Vapnik: "one should solve the problem directly and never solve a more general problem as an intermediate step"

- No one always wins!

- Theoretically, two regimes exist
  - Conditional models have lower asymptotic error
  - … but, generative models have better sample complexity, i.e., converge faster to asymptotic error

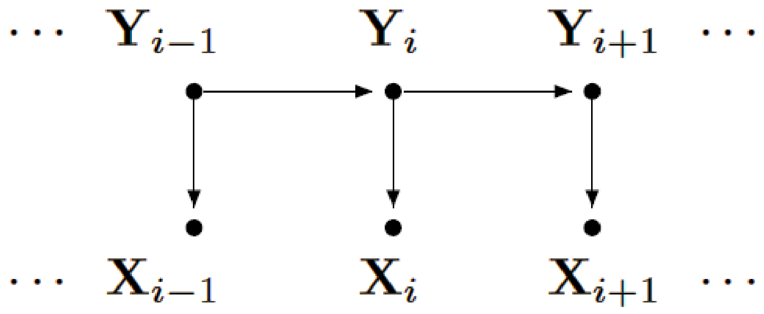
- Empirical results verify

# Structured outputs

- We consider sequential labeling
  - Application in computational linguistics & computer science
    - Text and speech processing, including topic segmentation, part-of-speech (POS) tagging
    - Information extraction
    - Syntactic disambiguation
  - Application in computational biology
    - DNA and protein sequence alignment
    - Sequence homolog searching in databases
    - Protein secondary structure prediction
    - RNA secondary structure analysis
- … but the ideas generalize to rich structures (difficulty lies in inference)

# Generative models

- Hidden Markov models (HMMs)
  - Assign a joint probability to paired observation and label sequences
  - The parameters typically trained to maximize the joint likelihood of train examples




$$P(\mathbf{X}, \mathbf{Y}) = \prod_i P(\mathbf{X}_i \,|\, \mathbf{Y}_i)\, P(\mathbf{Y}_i \,|\, \mathbf{Y}_{i-1})$$

  - Inference is done with forward-backward message passing

# Generative models (cont'd)

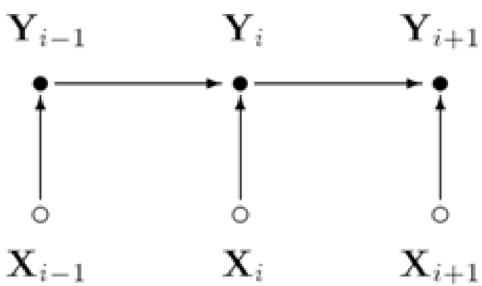
- Difficulties and disadvantages
  - Need to enumerate all possible observation sequences
  - Data sparsity
  - Not practical to represent multiple interacting features or long-range dependencies of the observations
  - Very strict independence assumptions on the observations

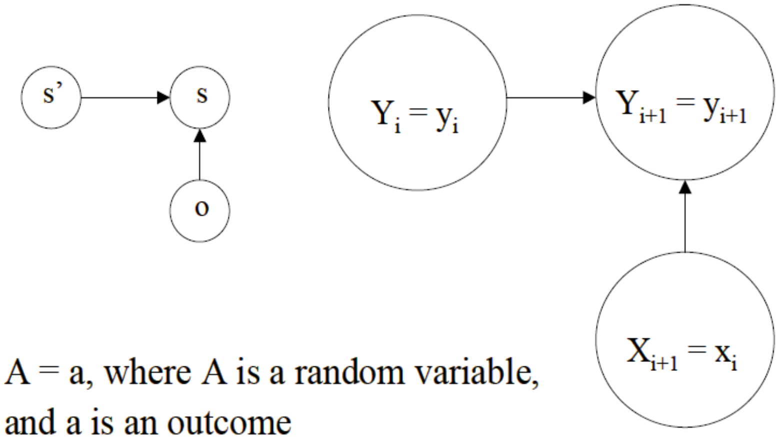
# Conditional models

- Conditional probability *P(label sequence* **y** *| observation sequence* **x***)* rather than joint probability *P*(**y, x**)
  - Specify the probability of possible label sequences given an observation sequence
- Allow arbitrary, non-independent features on the observation sequence X
- The probability of a transition between labels may depend on past and future observations
  - Relax strong independence assumptions in generative models

# Maximum entropy Markov models (MEMMs)

- Given training set X with label sequence Y:
  - Train a model θ that maximizes p(Y|X, θ)
  - For a new data sequence **x**, the predicted label **y** maximizes p(**y**|**x**, θ)



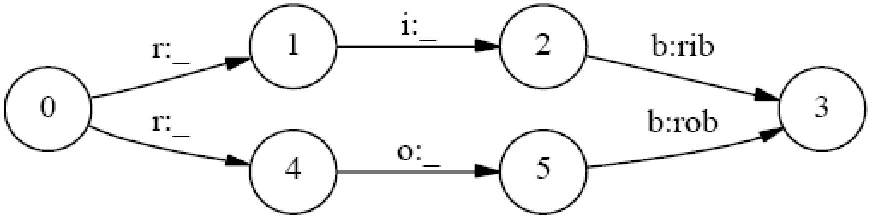

$$P(y' \mid y, x) = \frac{1}{Z(y,x)} \exp\left( \sum_k \underbrace{\lambda_k}_{\text{weight}} \underbrace{f_k(x, y, y')}_{\text{feature}} \right)$$

- Note: per-state/local normalization

# MEMMs (cont'd)

- MEMMs have all the advantages of conditional models
- But, it's subject to "label bias problem"
  - Bias toward states with fewer outgoing transitions
  - Due to per-state normalization:
    - all the mass that arrives at a state must be distributed among the possible successor states ("conservation of score mass")

# Label bias problem




since $p(2|1,x) = 1$ and $p(5|4,x) = 1, \quad \forall x$ (per-state normalization)

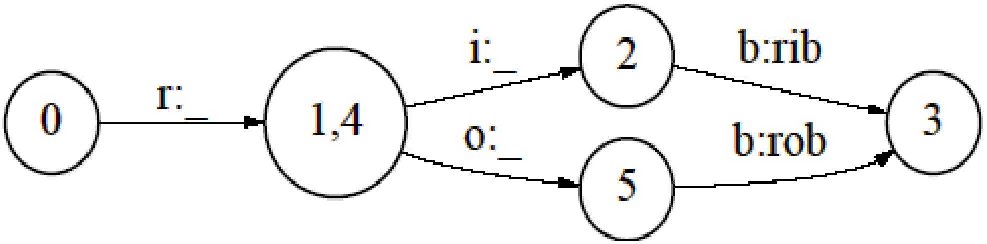
$$p(1,2|r,i) = p(1|r)p(2|1,i) = p(1|r)$$

$$p(4,5|r,i) = p(4|r)p(5|4,i) = p(4|r)$$

- The probability doesn't depend on the second observation
  - If one path is slightly more often in training, it always wins in testing!
- Does HMM has the label bias problem?

# Solve the label bias problem

- Change the state-transition structure of the model




  - Not always practical to change the set of states

- Start with a fully-connected model and let the training procedure figure out a good structure
  - Prelude the use of prior, which is very valuable (e.g. in information extraction)

# Conditional Random Fields (CRFs)

- CRFs have all the advantages of MEMMs without label bias problem
  - MEMM uses per-state exponential model for the conditional probabilities of next states given the current state
  - CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence

- Undirected graphs

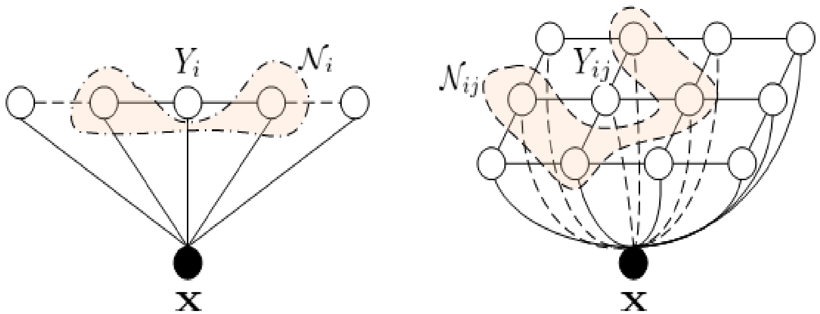
- Allow some transitions “vote” more strongly than others depending on the corresponding observations

# Definition of CRFs

**Definition.** *Let $G = (V, E)$ be a graph such that $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$, so that $\mathbf{Y}$ is indexed by the vertices of $G$. Then $(\mathbf{X}, \mathbf{Y})$ is a conditional random field in case, when conditioned on $\mathbf{X}$, the random variables $\mathbf{Y}_v$ obey the Markov property with respect to the graph: $p(\mathbf{Y}_v \,|\, \mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v \,|\, \mathbf{X}, \mathbf{Y}_w, w \sim v)$, where $w \sim v$ means that $w$ and $v$ are neighbors in $G$.*

- A random field model conditioned on inputs

- Examples:



# Conditional distribution

- If the graph $G = (V, E)$ of Y is a chain, the conditional distribution over the label sequence y, given x is:

$$p_\theta(\mathrm{y}\,|\,\mathrm{x}) = \frac{1}{Z(\mathrm{x})}\exp\left(\sum_{e\in E,k}\lambda_k f_k(e,\mathrm{y}|_e,\mathrm{x}) + \sum_{v\in V,k}\mu_k g_k(v,\mathrm{y}|_v,\mathrm{x})\right)$$

  - $f_k$ and $g_k$ are given and fixed. $g_k$ is a Boolean vertex feature; $f_k$ is a Boolean edge feature
  - $k$ is the number of features
  - $\theta = (\lambda_1, \lambda_2, \cdots, \lambda_n; \mu_1, \mu_2, \cdots, \mu_n)$; $\lambda_k$ and $\mu_k$ are parameters to be estimated
  - $\mathrm{y}|_e$ is the set of components of y defined by edge $e$
  - $\mathrm{y}|_v$ is the set of components of y defined by vertex $v$
  - Z(x) is a normalization over the data sequence x

# Parameter estimation for CRFs

- Lafferty et al., presented iterative scaling algorithms
- But it's very inefficient

$$\log p_\theta(y \mid x) = \sum_{e \in E,k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V,k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) - \log Z(\mathrm{x})$$

- More efficient learning algorithms
  - LBFGS with approximate Hessian

$$\frac{\partial \log p_\theta(y \mid x)}{\partial \theta} = \frac{\partial}{\partial \theta}\left( \sum_{e \in E,k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V,k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) - \log Z(\mathrm{x}) \right)$$

    - depending on graph structures, log Z(x) and its derivative can be hard
  - Other optimization algorithms apply

- Note: standard MCLE over-fits, 2-norm regularization saves a lot!

# Discriminative Learning from unstructured to structured …

– Logistic Regression
  - maximum likelihood estimation

$$\max_{\mathbf{w}} \mathcal{L}(\mathcal{D};\mathbf{w}) \triangleq \sum_{i=1}^{N} \log p(y^i|\mathrm{x}^i)$$

$$p(y|\mathrm{x}) = \frac{\exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x},y)\}}{\sum_{y'} \exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x},y')\}}$$

– Conditional Random Fields: CRFs
  - maximum likelihood estimation

$$\max_{\mathbf{w}} \mathcal{L}(\mathcal{D};\mathbf{w}) \triangleq \sum_{i=1}^{N} \log p(\mathbf{y}^i|\mathbf{x}^i)$$

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x},\mathbf{y})\}}{\sum_{\mathbf{y}'} \exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x},\mathbf{y}')\}}$$

– Support Vector Machines (SVM)
  - max-margin learning

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_{i=1}^{N}\xi_i;$$

$$\text{s.t.} \quad \mathbf{w}^\top \Delta\mathbf{f}_i(y) \geq 1 - \xi_i, \;\; \forall i, \forall y.$$

?

# Discriminative Learning
# from unstructured to structured …

- Logistic Regression
  - maximum likelihood estimation

$$\max_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{w}) \triangleq \sum_{i=1}^{N} \log p(y^i | \mathrm{x}^i)$$

$$p(y|\mathrm{x}) = \frac{\exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x}, y)\}}{\sum_{y'} \exp\{\mathbf{w}^\top \mathbf{f}(\mathrm{x}, y')\}}$$

- Conditional Random Fields: CRFs
  - maximum likelihood estimation

$$\max_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{w}) \triangleq \sum_{i=1}^{N} \log p(\mathbf{y}^i | \mathbf{x}^i)$$

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})\}}{\sum_{\mathbf{y}'} \exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}')\}}$$

- Support Vector Machines (SVM)
  - max-margin learning

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^{N} \xi_i;$$
$$\text{s.t.} \quad \mathbf{w}^\top \Delta \mathbf{f}_i(y) \geq 1 - \xi_i, \ \forall i, \forall y.$$

- Max-margin Markov Networks: M3Ns
  - max-margin learning

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} \xi_i$$
$$\text{s.t.} \quad \mathbf{w}^\top \Delta \mathbf{f}_i(\mathbf{y}) \geq \Delta \ell_i(\mathbf{y}) - \xi_i, \ \forall i, \forall \mathbf{y} \ ,$$

where $\mathbf{w}^\top \Delta \mathbf{f}_i(\mathbf{y})$ denotes the margin and $\Delta \ell_i(\mathbf{y})$ is a loss function.

# Max-margin Markov Networks

- Generalize the ideas of max-margin classifiers to structured output learning
- Like CRFs, it has a Markov graph structure
- But it doesn't define a normalized conditional distribution
- Instead, it directly learns a prediction model by doing opt.

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{s.t.} \quad \mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i, \ \forall i, \forall \mathbf{y} \ ,$$

# Learning M3Ns

- Many algorithms
  - Sequential minimal optimization (SMO)
  - Stochastic sub-gradient descent
  - Cutting-plane methods
  - Bundle methods
  - …

- Compare with SVM, the difficulty lies on inference!

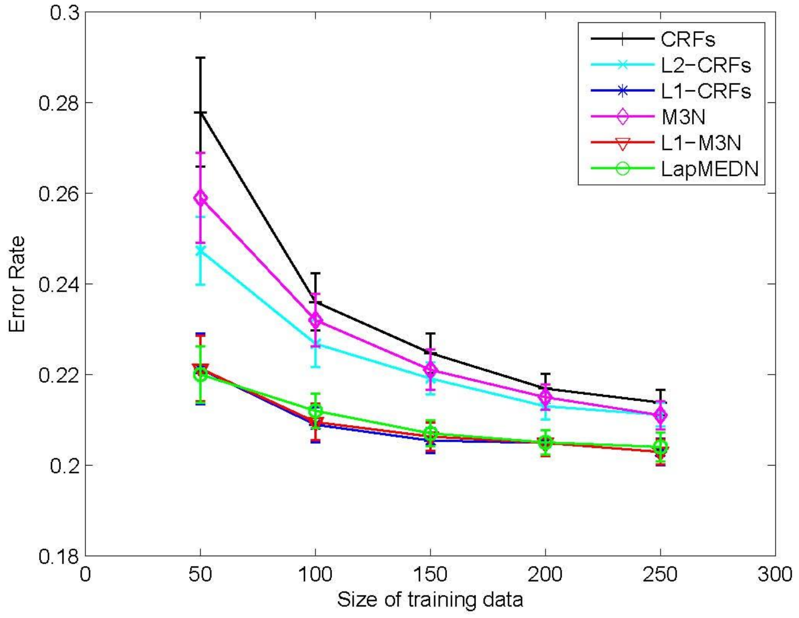
# CRFs vs. M3N

- Commons
  - have a Markov network to encode output structures
  - discriminative models dealing with arbitrary inputs
  - the kernel trick applies
  - can use various regularizors in learning

- Differences
  - Log-loss versus structured hinge loss
  - Probabilistic versus non-probabilistic (normalization matters!)

# Empirical comparison

- Synthetic datasets with 30 relevant features + 70 irrelevant features



# Other developments

- Direct task-dependent loss minimization

$$\theta^* = \operatorname*{argmin}_{\theta} \mathbb{E}\Big[L(\mathbf{y}, \mathbf{y}_\theta(\mathbf{x}))\Big]$$

- Problem:
  - task loss is typically non-convex, no polynomial algorithms with performance guarantees
  - convex surrogate (struct-SVM) is inconsistent
  - CRF maximizes likelihood, not related to task loss.

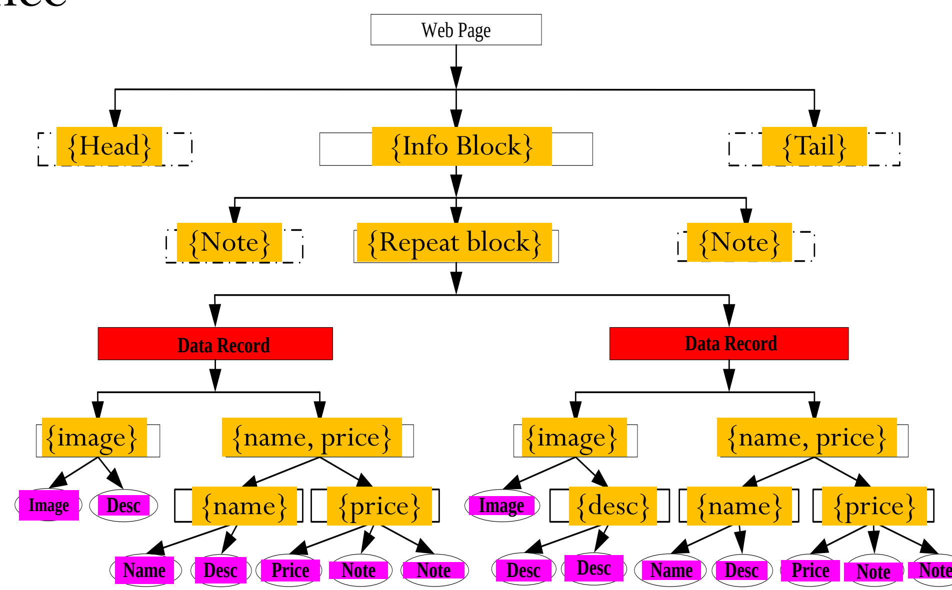
- A perceptron-like learning rule is constructed, whose expected update direction approaches the gradient of task loss
  - related to stochastic sub-gradient descent of struct-SVM.

# Other developments (cont'd)

- Markov logic networks
  - Use logic formula as dependence templates to construct a Markov network
  - Each formula is "softened" by associating with a weight
  - Generative or discriminative training
  - Structure learning
  - Many applications
- Open source:
  - http://alchemy.cs.washington.edu/

# Other developments (cont'd)

- Learning with structured latent variables
  - Hidden CRFs for object detection
  - Latent structural SVMs
  - Markov logic networks with latent variables
- Different strategies to consider latent variables
  - Mode, expectation, variance

# Other developments (cont'd)

- Discriminative training of generative models
  - Perceptron algorithm for HMMs
  - Max-margin learning for HMMs
  - Latent maximum entropy discrimination (MED)
  - MED Markov Networks
  - Nonparametric latent max-margin models

# Summary

- Structured learning
  - A suit of models, algorithms, and theories to deal with structured inputs, structured outputs, and structured models
- Conditional structured learning models
  - Conditional random fields
  - Max-margin Markov networks
  - …

# References


- Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Lafferty et al., ICML 2001 (Test-of-Time Award, 2011)
- Max-margin Markov Networks. Taskar et al., NIPS 2003
- Direct Loss Minimization for Structured Output Learning, McAllester et al., NIPS 2010
- On Discriminative vs. Generative classifies: A comparison of logistic regression and naïve Bayes. Ng & Jordan, NIPS 2001