Statistical Learning Models for Text and Graph Data Sequence Labeling and Structured Output Learning: Constraint Modeling

Yangqiu Song

Hong Kong University of Science and Technology

yqsong@cse.ust.hk

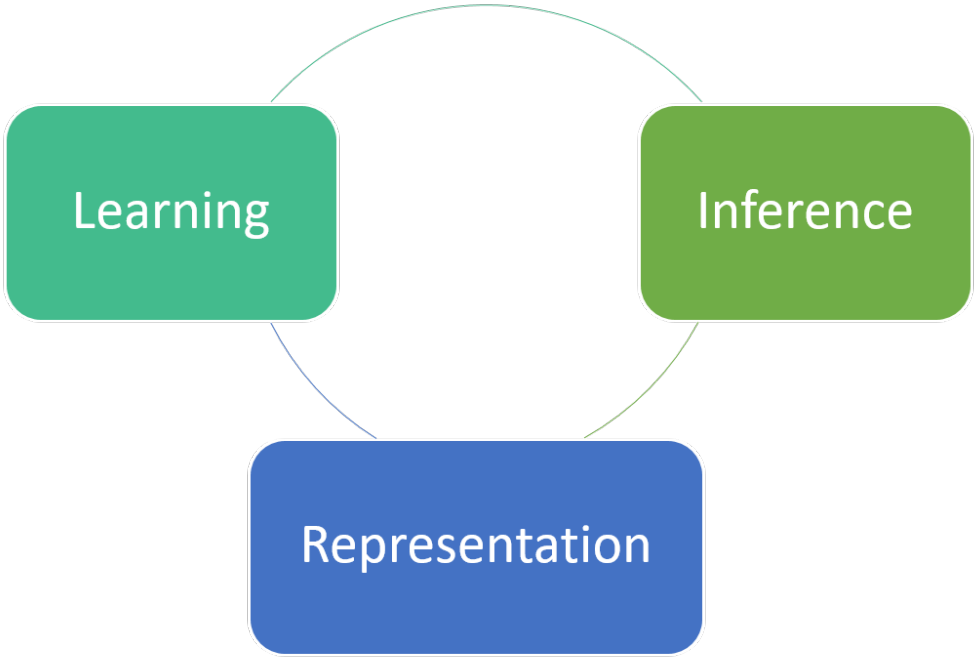
November 1, 2019

∗Contents are based on materials created by Dan Roth, Xiaojin (Jerry) Zhu

Reference Content

- Dan Roth. NAACL tutorial on Structured Predictions in NLP: Constrained Conditional Models and Integer Linear Programming. `http://l2r.cs.illinois.edu/tutorials.html`
- Dan Roth. CS546: Machine Learning and Natural Language . `http://l2r.cs.uiuc.edu/~danr/Teaching/CS546-16/`
- Xiaojin (Jerry) Zhu. CS 769: Advanced Natural Language Processing. `http://pages.cs.wisc.edu/~jerryzhu/cs769.html`

Course Topics



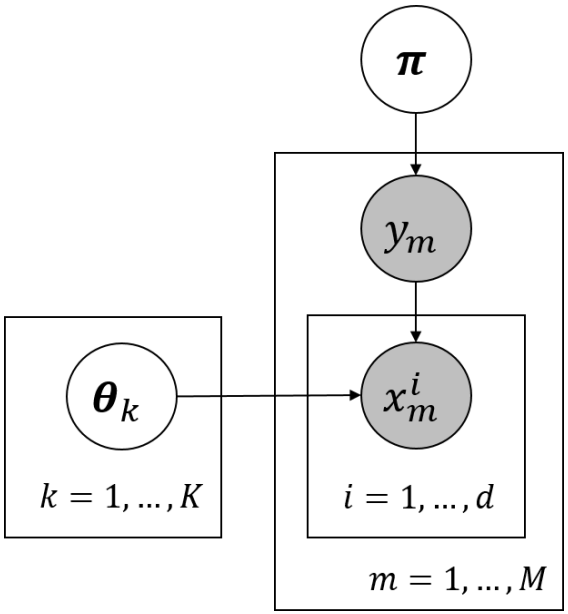
- Representation: language models, word embeddings, topic models, knowledge graphs
- Learning: supervised learning,unsupervised learning, semi-supervised learning, distant supervision, indirect supervision, sequence models, deep learning, optimization techniques
- Inference: constraint modeling, joint inference, search algorithms

Overview

1. Semi-supervised Mixture Models

2. Posterior Regularization
   - Motivation
   - Algorithm

Recall Naive Bayes Classifier: A Generative View

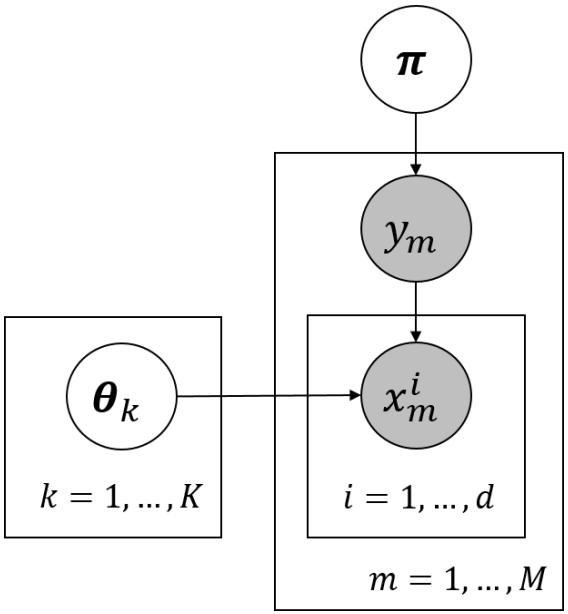


Both y_m and $\mathbf{x}_m = (x_m^1, \ldots, x_m^d)^T$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

Naive Bayes from Class Conditional Unigram Model

- For $m = 1, \ldots, M$
  - Choose $y_m \sim Multinomial(y_m|1, \boldsymbol{\pi})$
  - Choose $N_m = \sum_j^d x_m^j \sim Poisson(\xi)$
  - For $n = 1, \ldots, N_m$
    - Choose $v \sim Multinomial(v|1, \boldsymbol{\theta}_{*|y_m}) = \prod_{j=1}^d (\theta_{*|y_m}^j)^{v=j}$

Parameter Estimation (based on Multinomial)



Both y_m and $\mathbf{x}_m = x_m^1, \ldots, \mathbf{x}_m^d$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

Maximum likelihood of the training set:

$$\begin{aligned}\mathcal{J} &= \log \prod_{m=1}^{M} P_{\boldsymbol{\pi},\{\boldsymbol{\theta}_k\}}(\mathbf{x}_m, y_m) \\ &= \sum_{m=1}^{M} \log P_{\boldsymbol{\pi},\{\boldsymbol{\theta}_k\}}(\mathbf{x}_m, y_m) \\ &= \sum_{m=1}^{M} \log P(y_m|\boldsymbol{\pi}) P(\mathbf{x}_m|y_m, \boldsymbol{\theta}_{*|y_m})\end{aligned}$$

We can formulate a constrained optimization problem

$$\begin{aligned}\max \quad & \mathcal{J} \\ s.t. \quad & \sum_{k=1}^{K} \pi_k = 1 \\ & \sum_{j=1}^{d} \theta_k^j = 1 (k = 1, \ldots, K)\end{aligned}$$

It's easy to solve with Lagrange multiplier and arrive at:

$$\begin{aligned}\pi_k &= \frac{|\{y_m = k\}|}{M} \\ \theta_k^j &= \frac{\sum_{m, y_m = k} x_m^j}{\sum_{m, y_m = k} \sum_{j=1}^{d} x_m^j}\end{aligned}$$

What if the documents are not labeled?

In naive Bayes, both y_m and $\mathbf{x}_m = (x_m^1, \ldots, x_m^d)^T$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

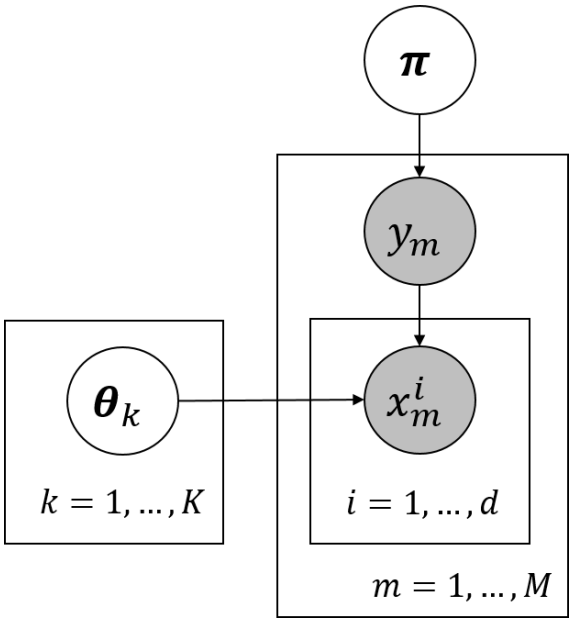


Figure: Native Bayes

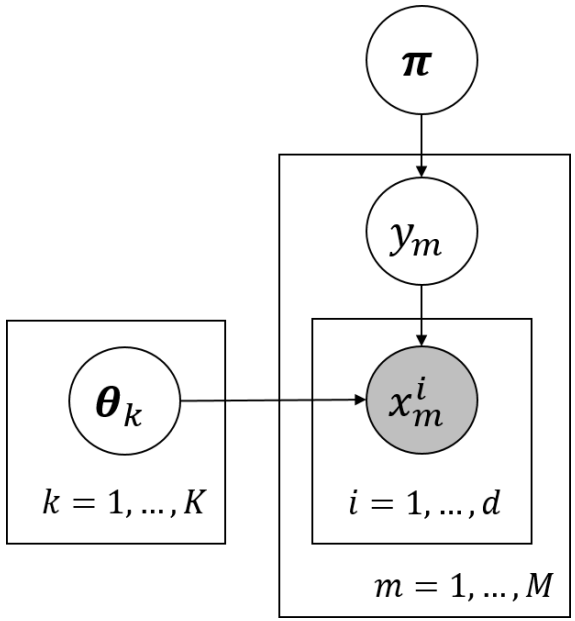


Figure: Mixture Model

However, in clustering problems, y_m is not observed (labeled before feeding into machine learning algorithm)

Expectation Maximization (EM) Algorithm

- We instead maximize the marginal log likelihood:

$$\mathcal{J}(\Theta) = \log P(\{\mathbf{x}_m\}_{m=1}^M | \Theta)$$

- Then the lower bound can be derived:

$$\begin{aligned}
\mathcal{J}(\Theta^t) &= \sum_{m=1}^M \log \sum_{y=1}^K P(\mathbf{x}_m, y | \Theta^t) \\
&= \sum_{m=1}^M \log \sum_{y=1}^K q_{\mathbf{x}_m,y}(\Theta) \frac{P(\mathbf{x}_m, y | \Theta^t)}{q_{\mathbf{x}_m,y}(\Theta)} \\
&\geq \sum_{m=1}^M \sum_{y=1}^K q_{\mathbf{x}_m,y}(\Theta) \log \frac{P(\mathbf{x}_m, y | \Theta^t)}{q_{\mathbf{x}_m,y}(\Theta)} \\
&\doteq Q(\Theta, \Theta^t)
\end{aligned}$$

where $\sum_{y=1}^K q_{\mathbf{x}_m,y}(\Theta) = 1$ is some distribution

EM Algorithm

- Repeat
  - E-step: compute posterior of hidden variables

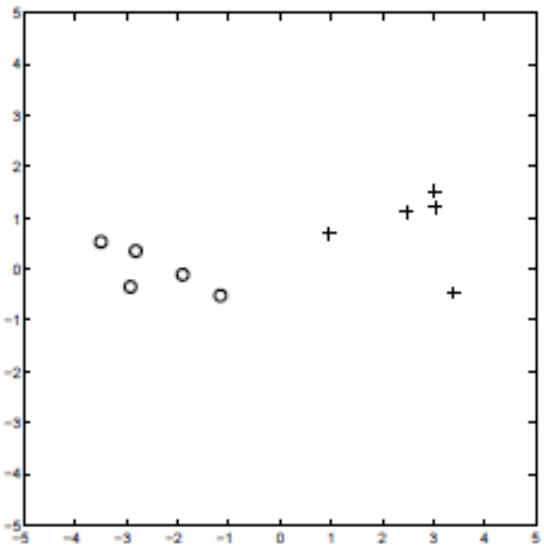
$$q_{\mathbf{x}_m,y} = P(y|\mathbf{x}_m, \Theta)$$

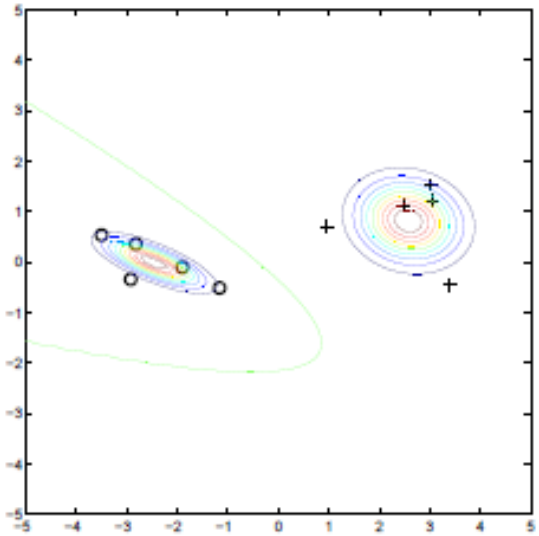
  - M-step: parameter estimation by maximizing the lower bound

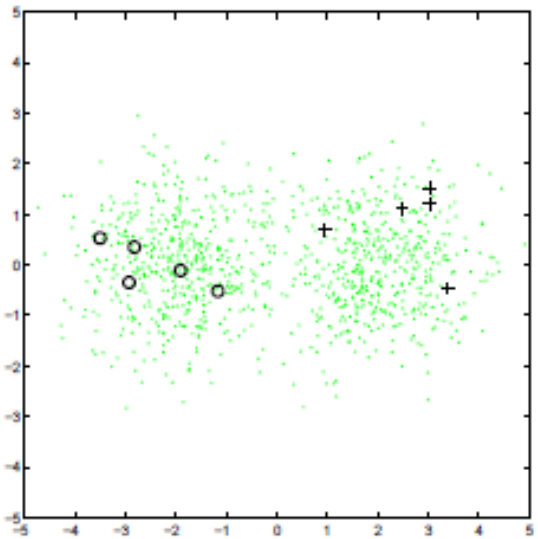
$$\pi_k = \frac{\sum_m q_{\mathbf{x}_m,y}}{M}$$
$$\theta_k^j = \frac{\sum_m q_{\mathbf{x}_m,y} x_m^j}{\sum_m \sum_{j=1}^d q_{\mathbf{x}_m,y} x_m^j}$$

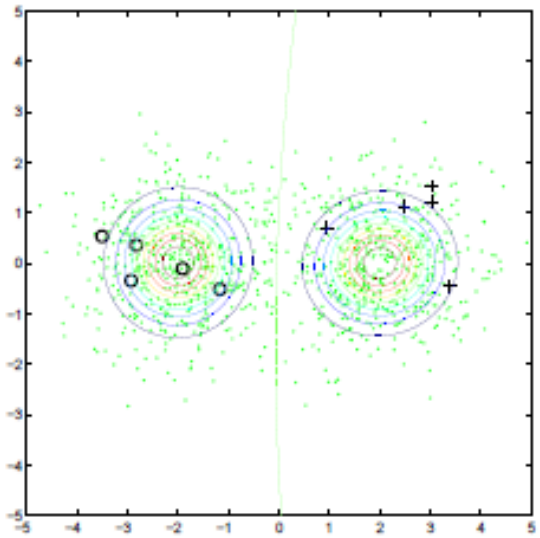
- Until the convergence of the objective function

What if the Data is Semi-supervised?









Semi-supervised Text Classification (Nigam et al. (2000))

- We modify the log-likelihood as

$$\mathcal{J}(\Theta) = \sum_{m=1}^{M_l} \log P(\mathbf{x}_m, y_m|\Theta) + \lambda \sum_{m=1}^{M_u} \log \sum_{y=1}^{K} P(\mathbf{x}_m, y|\Theta)$$

- When $\lambda = 0$, it the supervised learning case
- We still need to perform EM algorithm since there is a sum inside log
  - In E-step, we estimate $q_{\mathbf{x}_m,y} = P(y|\mathbf{x}_m, \Theta)$ for unlabeled data
  - In M-step, we modify the algorithm based both labeled and unlabeled data

$$\pi_k = \frac{\sum_m^{M_l} I(y_m=k) + \lambda \sum_m^{M_u} q_{\mathbf{x}_m,y}}{M_l + M_u}$$

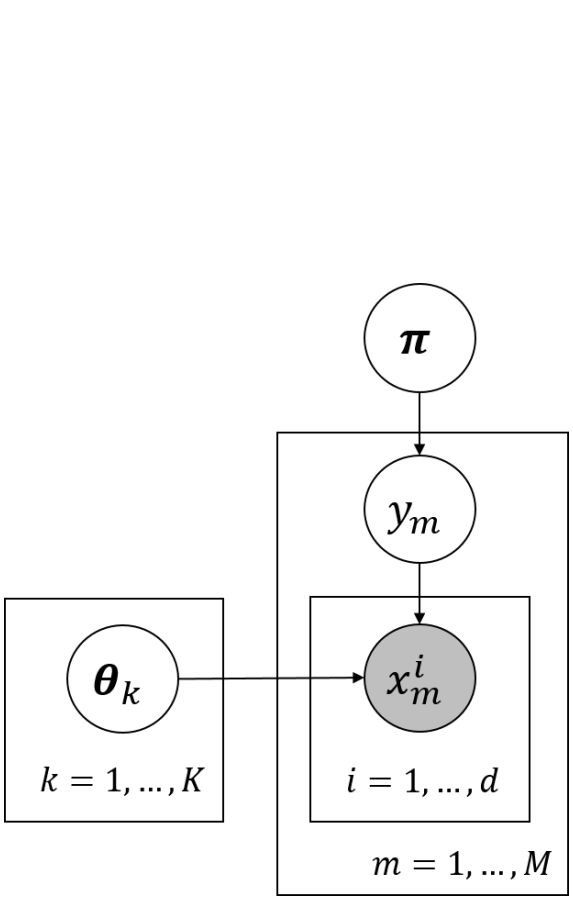
$$\theta_k^j = \frac{\sum_{m,y_m=k}^{M_l} x_m^j + \lambda \sum_m^{M_u} q_{\mathbf{x}_m,y} x_m^j}{\sum_{m,y_m=k}^{M_l} \sum_{j=1}^{d} x_m^j + \lambda \sum_m^{M_u} \sum_{j=1}^{d} q_{\mathbf{x}_m,y} x_m^j}$$

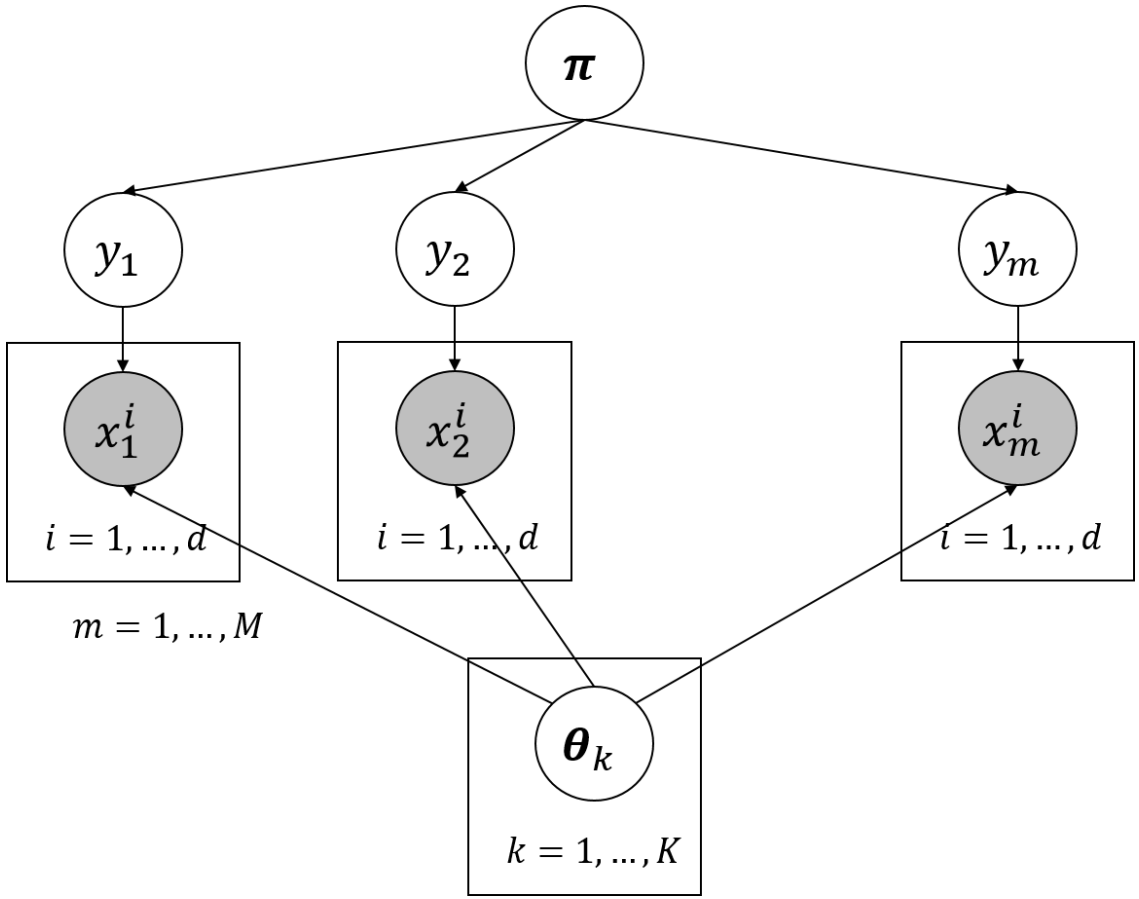
What Kind of Other Supervision Can We Use?

- Pairwise constraints
  - Must-link: two data samples must be in the same class
  - Cannot-link: two data samples cannot be in the same class
- Constrained clustering (Basu et al. (2004))
  - Still consider mixture modeling
  - Add pairwise constraints to labels

Graphical Model for Constrained Clustering

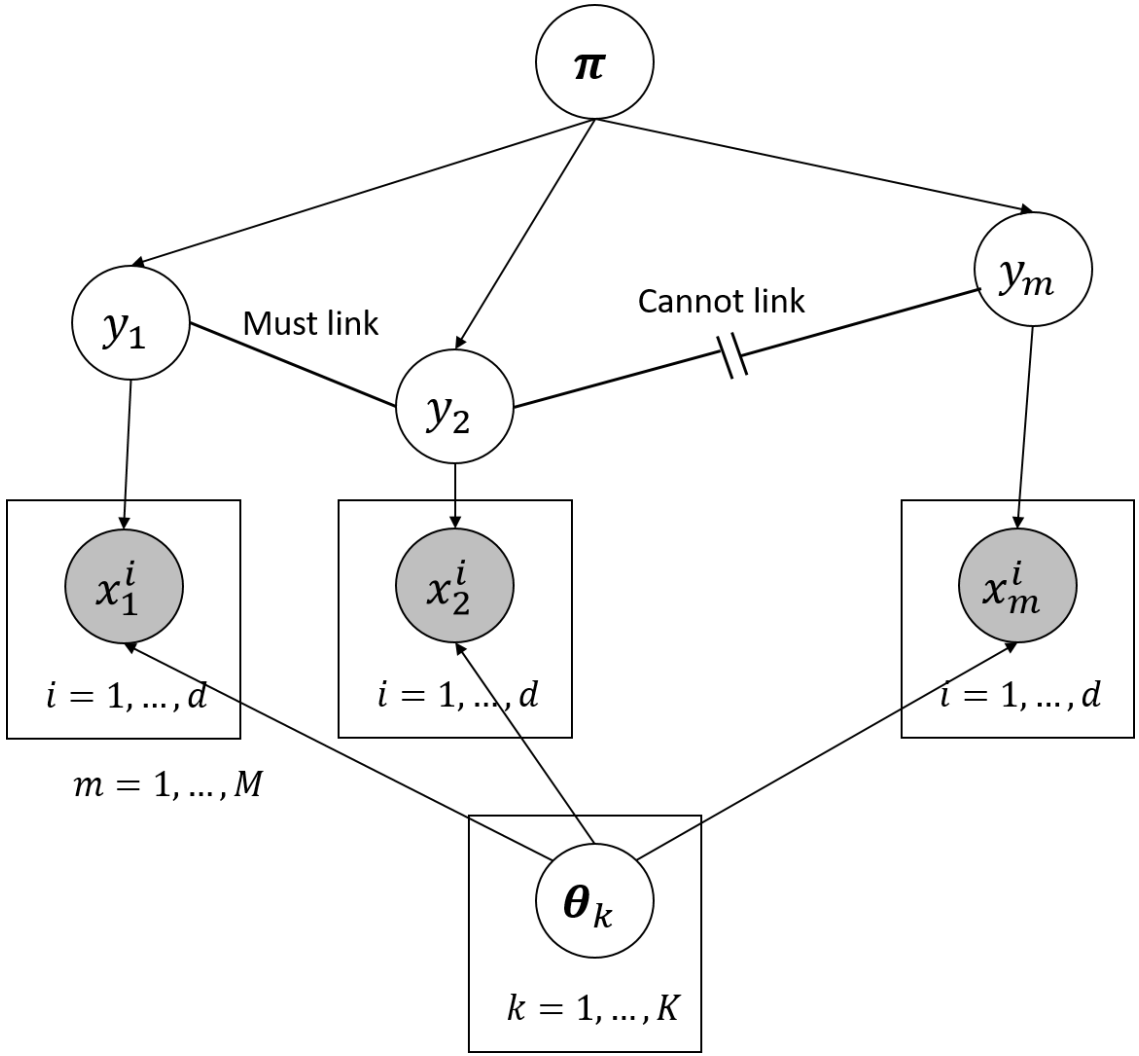
Recall





Graphical Model for Constrained Clustering

A hidden Markov random field model over labels



Hidden Markov Modeling

- Emission probability: $P(\mathbf{x}|y)$ is a multinomial distribution for document classification
  - As what we did in naive Bayes classifier or unsupervised/semi-supervised mixture models
- Markov random field over labels $P(\mathcal{Y}) \doteq \frac{1}{Z}\exp(-\frac{E}{T})$
  - A Gibbs distribution defined based on some energy function E
  - Z is a normalization constant
- For must-links

$$E_M(y_i, y_j) \propto I(y_i \neq y_j)$$

  which penalize two examples with different estimated labels but with a must-link constraint
- For cannot-links

$$E_C(y_i, y_j) \propto I(y_i = y_j)$$

  which penalize two examples with the same estimated label but with a cannot-link constraint

Constrained Clustering

- Objective function:

$$\begin{aligned}
&\mathcal{J}(\Theta) \\
&= \sum_{m=1}^{M} \log \sum_{y=1}^{K} P(\mathbf{x}_m, y|\Theta) - \sum_{i,j\in\mathcal{M}} E_M(y_i, y_j) - \sum_{i,j\in\mathcal{C}} E_C(y_i, y_j) \\
&\geq \sum_{m=1}^{M} \sum_{y=1}^{K} P(y|\mathbf{x}_m, \Theta) \log \frac{P(\mathbf{x}_m, y|\Theta^t)}{P(y|\mathbf{x}_m, \Theta)} \\
&- \sum_{i,j\in\mathcal{M}} E_M(y_i, y_j) - \sum_{i,j\in\mathcal{C}} E_C(y_i, y_j)
\end{aligned}$$

- Recall: When we consider a finite mixture model, and draw just one sample at each E-step
  - This is called stochastic EM
  - In the E-step, a sample of y is taken from the posterior distribution $P(y|\mathbf{x}, \Theta^t)$
  - This effectively makes a hard assignment of each data point to one of the components in the mixture
- If Gibbs sampling is used
  - Instead of drawing a sample from the corresponding conditional distribution, we make a point estimate of the variable given by the maximum of the conditional distribution
  - Then we obtain the iterated conditional modes (ICM) algorithm
  - For finite mixture models, it's similar to K-means

Constrained Clustering (Cont'd)

- Objective function:

$$
\begin{aligned}
&\mathcal{J}(\Theta) \\
&= \sum_{m=1}^{M} \log \sum_{y=1}^{K} P(\mathbf{x}_m, y|\Theta) + \sum_{i,j\in\mathcal{M}} E_M(y_i, y_j) + \sum_{i,j\in\mathcal{C}} E_C(y_i, y_j) \\
&\geq \sum_{m=1}^{M} \sum_{y=1}^{K} P(y|\mathbf{x}_m, \Theta) \log \frac{P(\mathbf{x}_m, y|\Theta^t)}{P(y|\mathbf{x}_m, \Theta)} \\
&+ \sum_{i,j\in\mathcal{M}} E_M(y_i, y_j) + \sum_{i,j\in\mathcal{C}} E_C(y_i, y_j)
\end{aligned}
$$

- In E-step, we use iterated conditional modes (ICM) algorithm
  - We re-assign the cluster labels due to the maximization of the objective function
- In M-step, we maximize the parameter Θ exactly the same as mixture models
  - Given the cluster labels for each example, we re-calculate cluster centers (like K-means)

Summary of Semi-supervised Clustering

- Semi-supervised learning with seeds
  - Supervision is coming from the prior of individual labels
  - Augment maximum likelihood with supervised learning likelihood
  - Modify the M-step with both labeled and unlabeled data
- Semi-supervised learning with constraints
  - Supervision is coming from the pairwise labels
  - Augment maximum likelihood with hidden Markov random field model
  - Modify the E-step with both unlabeled data and constraints
- Can we generalize both ideas?
  - For the individual supervision, there have been a lot of semi-supervised learning algorithm with assumptions such as
    - Continuity assumption: Points which are close to each other are more likely to share a label; yields a preference for decision boundaries in low-density regions
    - Cluster assumption: The data tend to form discrete clusters, and points in the same cluster are more likely to share a label
    - Manifold assumption: The data lie approximately on a manifold of much lower dimension than the input space
  - For the pairwise supervision: today's lecture

Overview

1 Semi-supervised Mixture Models

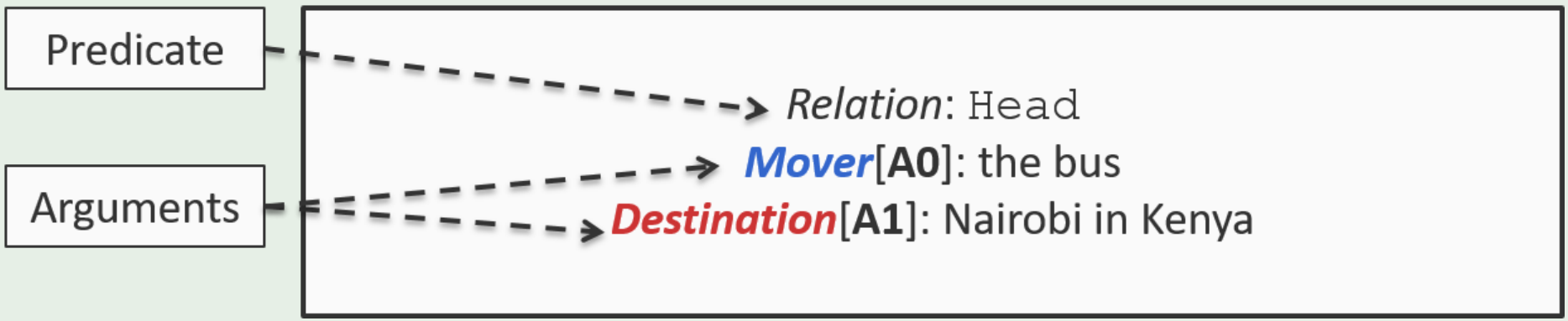
2 Posterior Regularization
- Motivation
- Algorithm

Semantic Role Labeling

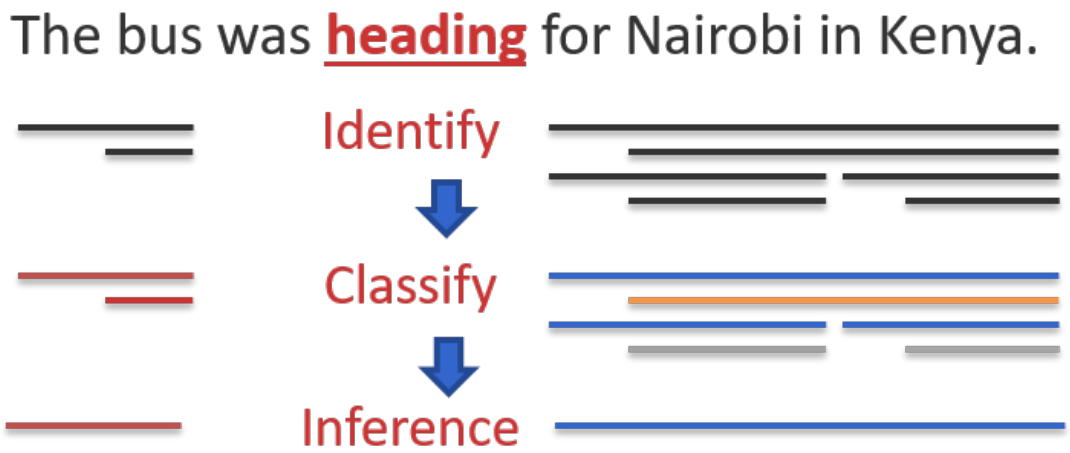
- Based on the dataset PropBank (Palmer et al. (2005))
  - Large human-annotated corpus of verb semantic relations
- The task: To predict arguments of verbs

Example ("The bus was **heading** for Nairobi in Kenya")

Given the sentence, identifies who does what to whom, where and when.

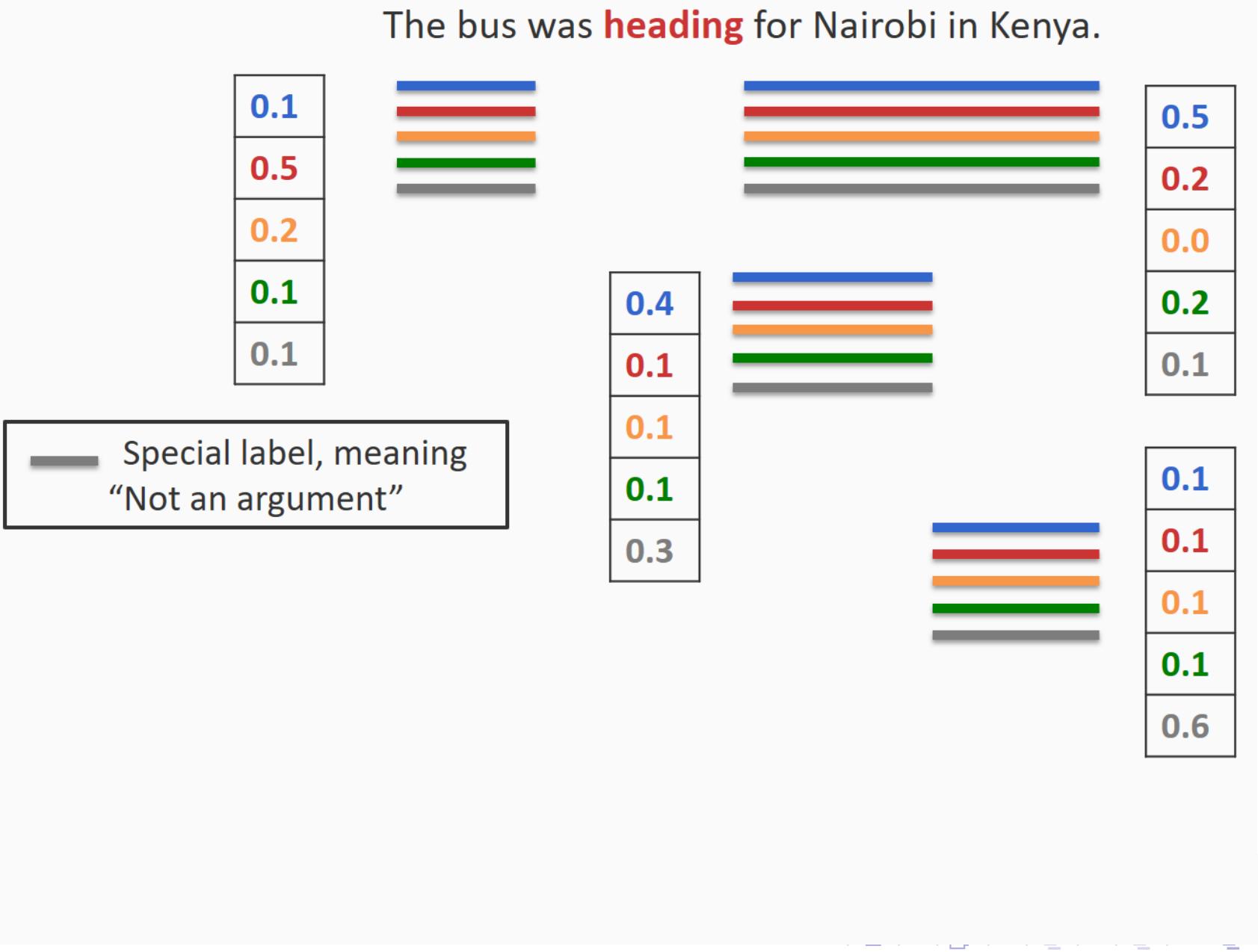


Predicting Verb Arguments

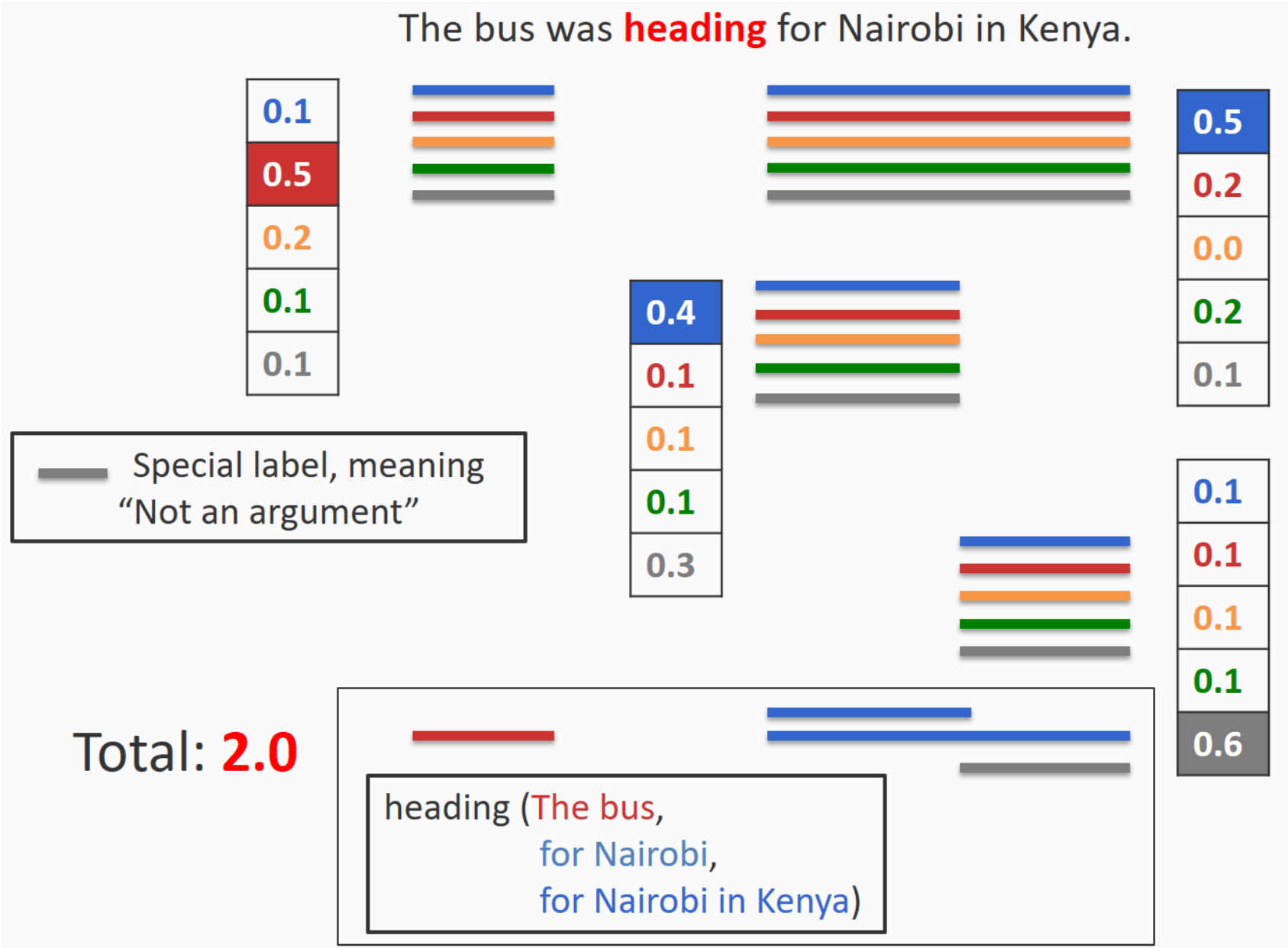


- Identify candidate arguments for verb using parse tree
  - Filtered using a binary classifier
- Classify argument candidates
  - Multi-class classifier (one of multiple labels per candidate)
- Inference
  - Using probability estimates from argument classifier
  - Must respect structural and linguistic constraints, e.g., no overlapping arguments

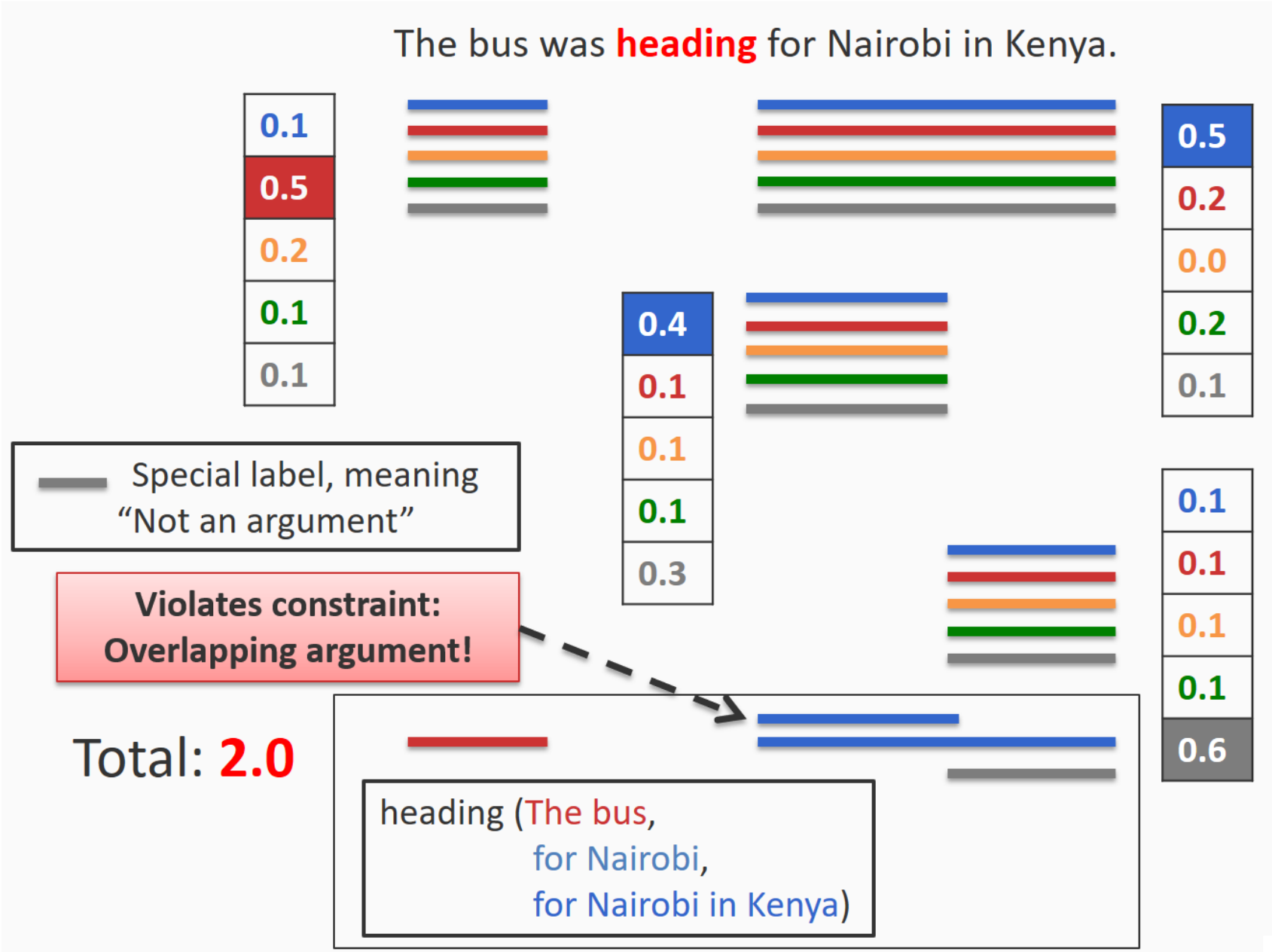
Inference: Verb Arguments



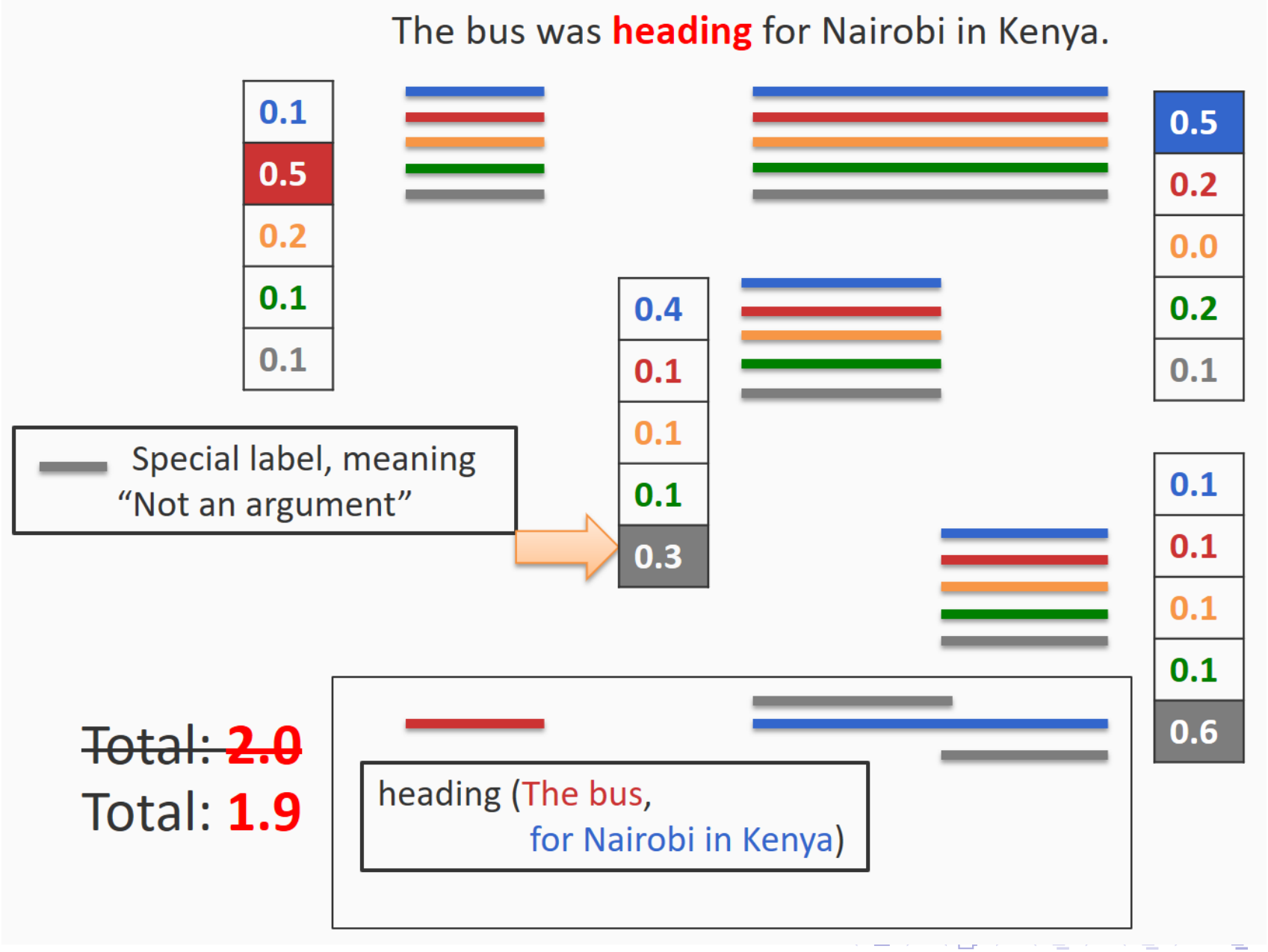
Inference: Verb Arguments



Inference: Verb Arguments



Inference: Verb Arguments



Many Other Such Constraints in NLP

- Recognizing entities and relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

other	0.05
per	**0.50**
loc	0.45

Dole 's wife, Elizabeth , is a native of N.C.

irrelevant	0.05
spouse_of	0.45
born_in	**0.50**

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

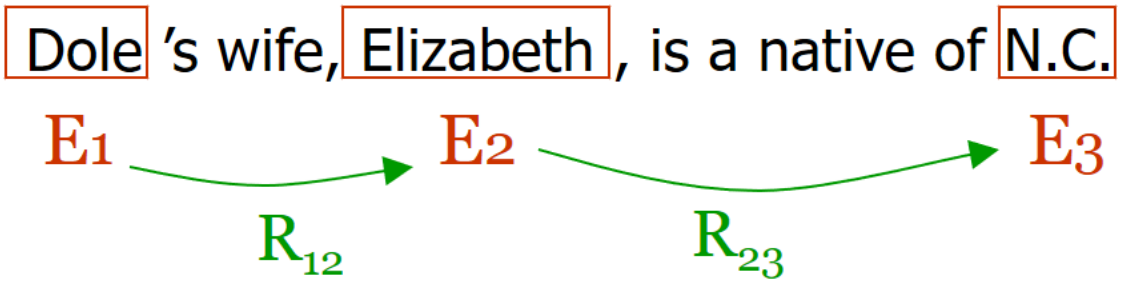
Many Other Such Constraints in NLP

- Recognizing entities and relations

other	0.05
per	**0.85**
loc	0.10

other	0.10
per	**0.60**
loc	0.30

other	0.05
per	0.50
loc	**0.45**



irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

- This helps improve 2-5% over no inference (Roth and Yih (2004))

Many Other Such Constraints in NLP

- Word alignment: Symmetric: link is used by source$\rightarrow$target model and target$\rightarrow$source model
- Multi-view learning: both view should predict the same label
- Part-of-speech tagging: each sentence should have at least one verb and at least one noun

Overview

1 Semi-supervised Mixture Models

2 Posterior Regularization
- Motivation
- Algorithm

A Running Example

- The task is part-of-speech (POS) tagging with limited or no training data.
- Suppose we know that each sentence should have at least one verb and at least one noun,
- and would like our model to capture this constraint on the unlabeled sentences.
- The model we will be using is a first-order hidden Markov model (HMM).

Running Example

- In the POS tagging example from above, we would use

$$\log P_\Theta(\mathbf{x}_{1:N}, y_{1:N}) = \sum_n \log P_\Theta(y_n|y_{n-1}) + P_\Theta(\mathbf{x}_n|y_n)$$

  as the joint probability
- Θ represents the multinomial distributions
- The log-likelihood (+ log-prior) is

$$\mathcal{J}(\Theta) = \sum_i^{M_L} \log P_\Theta(\mathbf{x}^{(i)}_{L,1:N}, y^{(i)}_{L,1:N}) + \sum_i^{M_U} \log \sum_{y_{1:N}} P_\Theta(\mathbf{x}^{(i)}_{1:N}, y_{1:N}) + \log P(\Theta)$$

  which is a general MAP setting for semi-supervised learning

Regularization via Posterior Constraints

- The goal of the posterior regularization framework is to
  - restrict the space of the model posteriors on unlabeled data to guide the model towards desired behavior
- Here we want to bias learning so that each sentence is labeled to contain at least one verb
- We define $\phi(\mathbf{x}_{1:N}, y_{1:N})$ as "negative number of verbs in $y_{1:N}$"
- Now the constraint over the corpus is

$$Q_{\mathbf{x}} = \{q_{\mathbf{x}}(y_{1:N}) : \quad \mathbb{E}_{q_{\mathbf{x}}}[\phi(\mathbf{x}_{1:N}, y_{1:N})] \leq -1\}$$

- More generally, we can define the constraint set to be

$$Q_{\mathbf{x}} = \{q_{\mathbf{x}}(y_{1:N}) : \quad \exists \xi, \mathbb{E}_{q_{\mathbf{x}}}[\phi(\mathbf{x}_{1:N}, y_{1:N})] - \mathbf{b} \leq \xi; \ ||\xi||_{\beta} \leq \epsilon\}$$

Recall: A More General View of EM

- One can introduce an arbitrary distribution over hidden variables $Q(Y)$

$$\begin{aligned}
\mathcal{J}(\Theta) &= \log P(X|\Theta) = \log \textstyle\sum_Y P(X,Y|\Theta) \\
&= \textstyle\sum_Y Q(Y) \log P(X|\Theta) \\
&= \textstyle\sum_Y Q(Y) \log \frac{P(X|\Theta)Q(Y)P(X,Y|\Theta)}{P(X,Y|\Theta)Q(Y)} \\
&= \textstyle\sum_Y Q(Y) \log \frac{P(X,Y|\Theta)}{Q(Y)} + \sum_Y Q(Y) \log \frac{P(X|\Theta)Q(Y)}{P(X,Y|\Theta)} \\
&= \textstyle\sum_Y Q(Y) \log \frac{P(X,Y|\Theta)}{Q(Y)} + \sum_Y Q(Y) \log \frac{Q(Y)}{P(Y|X,\Theta)} \\
&= F(Q,\Theta) + KL[Q(Y)||P(Y|X,\Theta)]
\end{aligned}$$

- Note $F(Q,\Theta)$ is the right hand side of Jensen's inequality
  - If $KL > 0$, $F(Q,\Theta)$ is a lower bound of $\mathcal{J}(\Theta)$
- First consider the maximization of F on Q with Θ^t fixed
  - $F(Q,\Theta)$ is maximized by $Q(Y) = P(Y|X,\Theta^t)$ since $\mathcal{J}(\Theta)$ is fixed and KL attends its minimum zero (E-Step)
- Next consider the maximization of F on Θ with Q fixed as above
  - Note in this case $F(Q,\Theta) = Q(\Theta^t,\Theta)$ (M-Step)

Illustration of EM

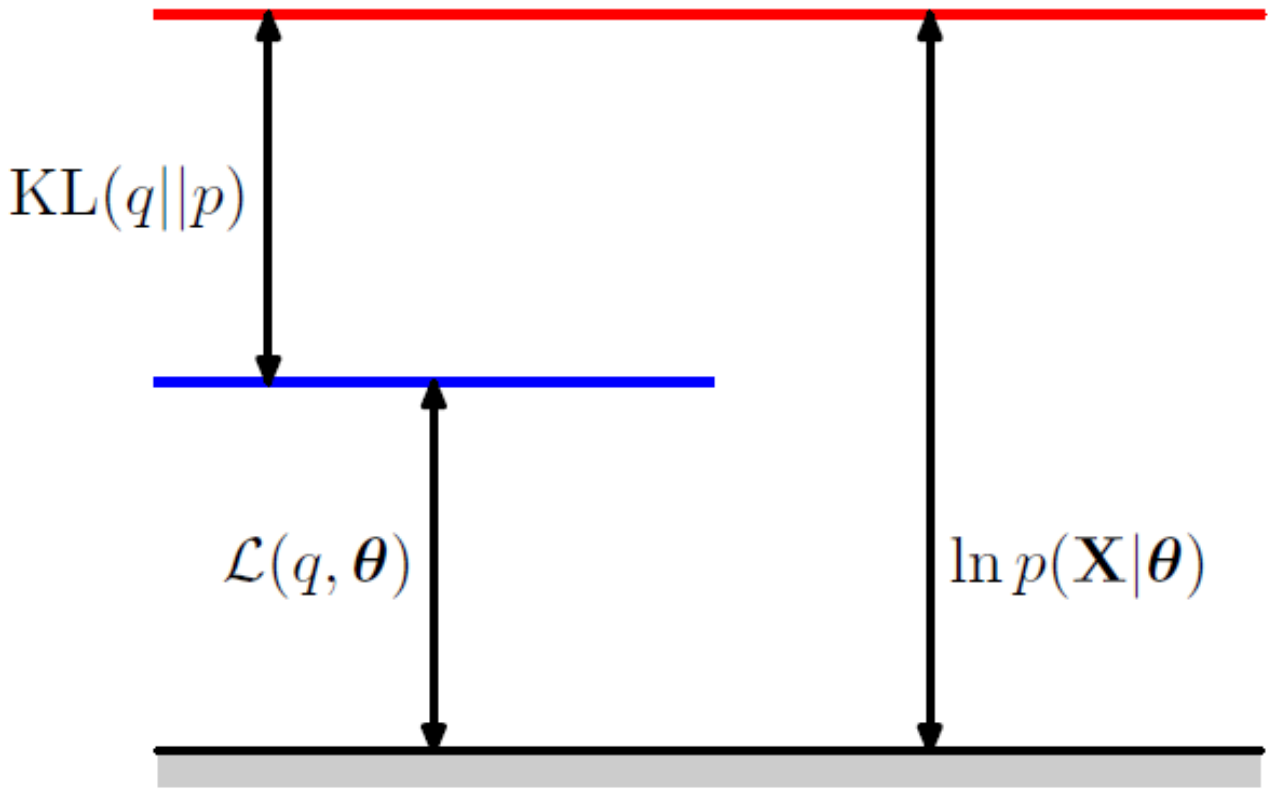


Figure: EM Algorithm

Illustration of EM

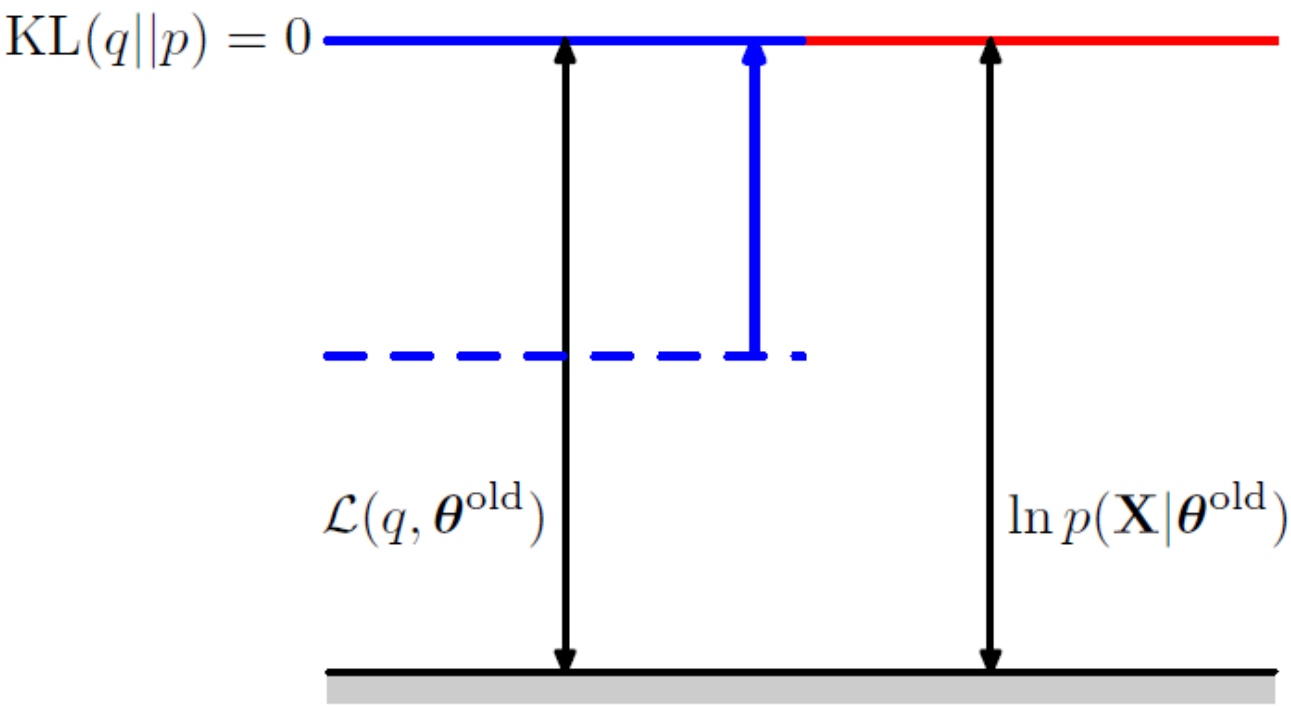


Figure: EM Algorithm

Illustration of EM

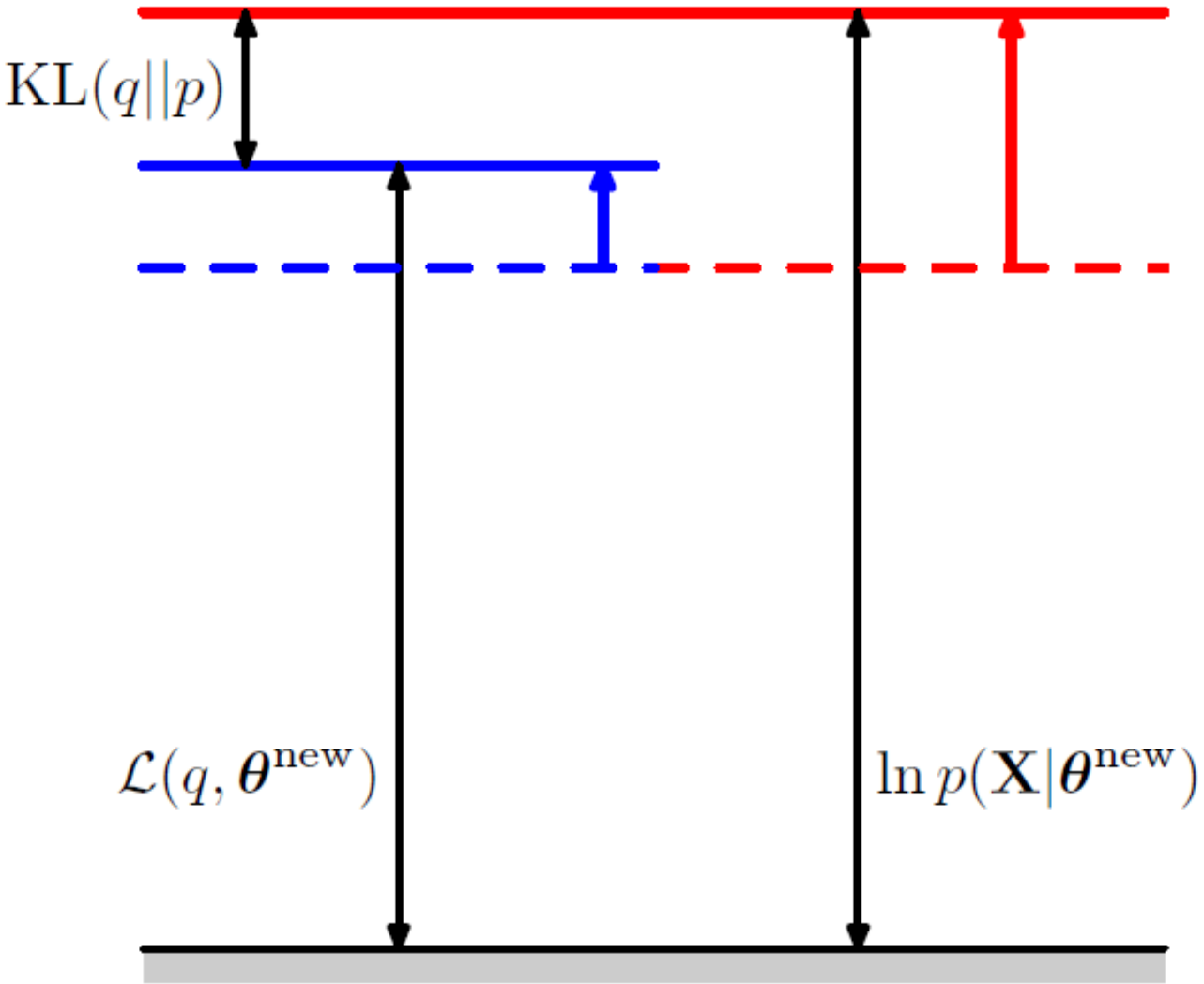
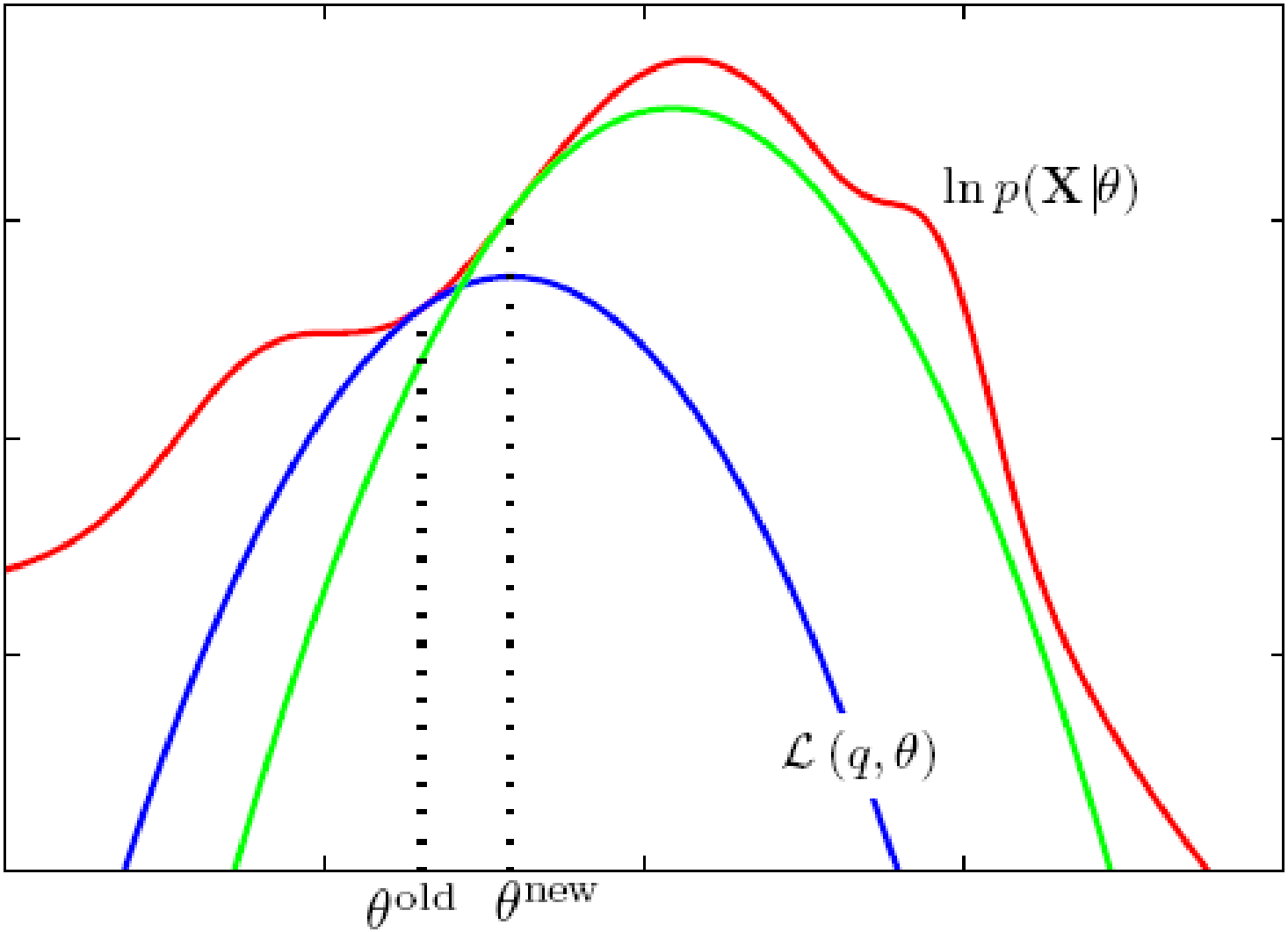


Figure: EM Algorithm

EM Algorithm: General Idea



EM Algorithm for Posterior Regularization

- Note that we constrain q as

$$Q_{\mathbf{x}} = \{q_{\mathbf{x}}(y_{1:N}) : \quad \exists \xi, \mathbb{E}_{q_{\mathbf{x}}}[\phi(\mathbf{x}_{1:N}, y_{1:N})] - \mathbf{b} \leq \xi; \ ||\xi||_\beta \leq \epsilon\}$$

- So in E-step, we will possibly find a sub-optimal solution which is not exactly the posterior:

$$\begin{array}{ll} \min_{q_{\mathbf{x}},\xi} & \sum_i^M KL[q_{\mathbf{x}}(y_{1:N}^{(i)})||P_{\Theta^t}(y_{1:N}^{(i)}|\mathbf{x}_{1:N}^{(i)})] \\ s.t. & \mathbb{E}_{q_{\mathbf{x}}}[\phi(\mathbf{x}_{1:N}^{(i)}, y_{1:N}^{(i)})] - \mathbf{b} \leq \xi; \ ||\xi||_\beta \leq \epsilon \ \text{ for all } \ i \end{array}$$

  - This is corresponding to maximizing $F(Q, \Theta^t) = \mathcal{J}(\Theta^t) - KL[Q(Y)||P(Y|X, \Theta^t)]$ w.r.t. Q to obtain Q^{t+1}
  - In traditional way, we minimize $KL[Q(Y)||P(Y|X, \Theta^t)]$
- In the M-step, we maximize $F(Q^{t+1}, \Theta)$ with respect to Θ, which is

$$F(Q^{t+1}, \Theta) = \sum_Y Q^{t+1}(Y) \log \frac{P(X,Y|\Theta)}{Q^{t+1}(Y)} = \mathbb{E}_{Q^{t+1}(Y)}[\log P(X, Y|\Theta)] + const$$

which is exactly the same as traditional EM algorithm

Objective Function for PR

- Comparing the original cost function used in EM

$$\begin{aligned}
\mathcal{J}(\Theta) &= \log P(X|\Theta) = \log \textstyle\sum_Y P(X,Y|\Theta) \\
&= \textstyle\sum_Y Q(Y) \log P(X|\Theta) \\
&= \textstyle\sum_Y Q(Y) \log \frac{P(X|\Theta)Q(Y)P(X,Y|\Theta)}{P(X,Y|\Theta)Q(Y)} \\
&= \textstyle\sum_Y Q(Y) \log \frac{P(X,Y|\Theta)}{Q(Y)} + \sum_Y Q(Y) \log \frac{P(X|\Theta)Q(Y)}{P(X,Y|\Theta)} \\
&= \textstyle\sum_Y Q(Y) \log \frac{P(X,Y|\Theta)}{Q(Y)} + \sum_Y Q(Y) \log \frac{Q(Y)}{P(Y|X,\Theta)} \\
&= F(Q,\Theta) + KL[Q(Y)||P(Y|X,\Theta)]
\end{aligned}$$

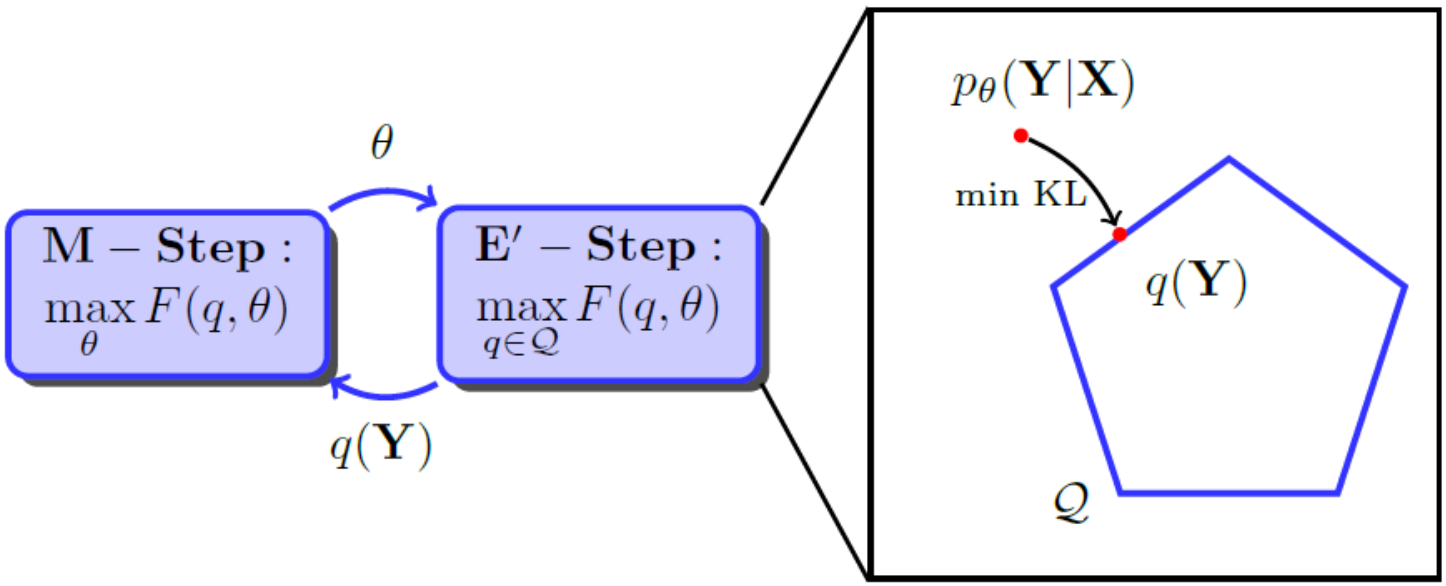
- The actual objective function for PR is

$$\mathcal{J}(\Theta) - KL[Q(Y)||P(Y|X,\Theta)] = F(Q,\Theta)$$

  by alternatively optimizing Q and Θ in E-step and M-step

- The difference is in E-step, we optimize Q with constraints

Illustration



PR for Discriminative Models

- For a discriminative model, we directly optimize

$$\mathcal{J}(\Theta) = \log P_\Theta(Y|X) + \log P(\Theta)$$

- We define the same object function

$$\mathcal{J}(\Theta) - KL[Q(Y)||P(Y|X,\Theta)]$$

- For labeled data part, we can still use
  $\mathcal{J}(\Theta) = \log P_\Theta(Y|X) + \log P(\Theta)$
- Here we define the lower bound function
  $F'(Q,\Theta) = -KL[Q(Y)||P(Y|X,\Theta)]$ for the unlabeled data
- Then in E-step, we maximize

$$F'(Q,\Theta^t) = -KL[Q(Y)||P(Y|X,\Theta^t)]$$

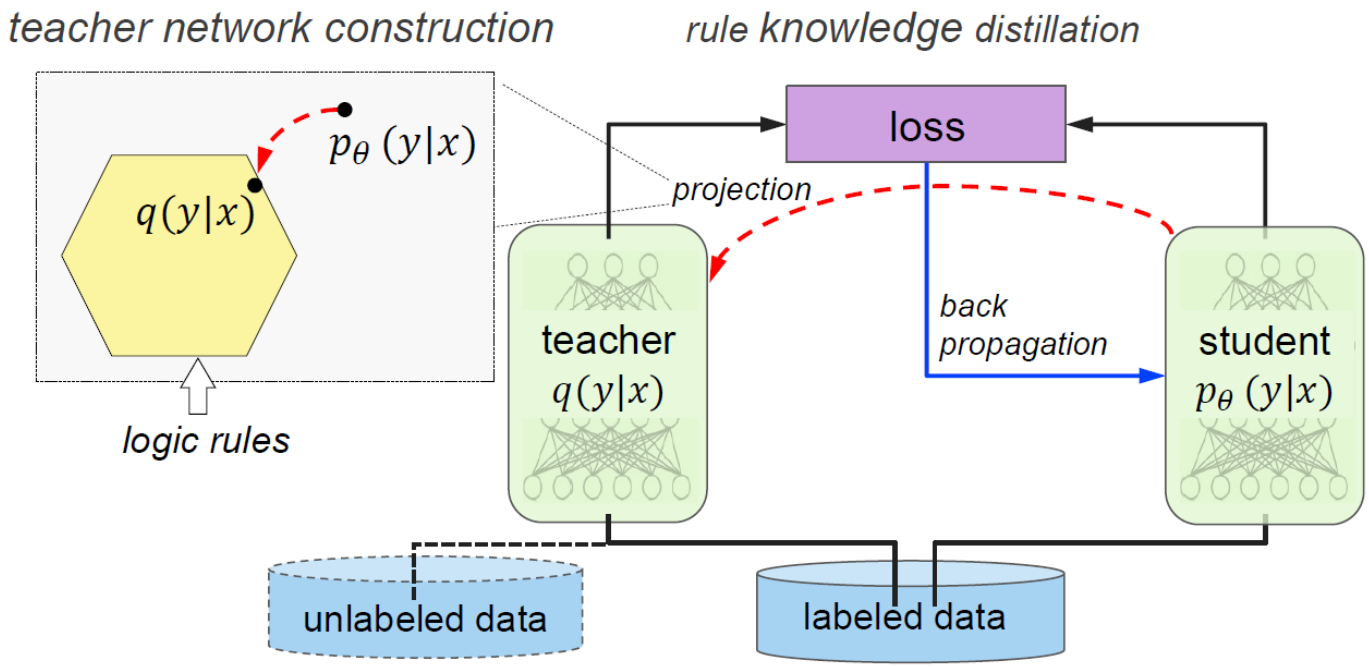
  w.r.t. Q to obtain Q^{t+1}
- In the M-step, we maximize

$$F'(Q^{t+1},\Theta) = -KL[Q^{t+1}(Y)||P(Y|X,\Theta)]$$

  w.r.t. Θ to obtain Θ^{t+1}

Further Reading

- Ganchev et al. (2010): Posterior Regularization for Structured Latent Variable Models
- Hu et al. (2016): Harnessing Deep Neural Networks with Logic Rules



Results on Stanford Sentiment Treebank

(Socher et al., 2013)

Rule: sentence S with an "A-but-B" structure, then expect the sentiment of the whole sentence to be consistent with the sentiment of clause B

	Data size	5%	10%	30%	100%
1	CNN	79.9	81.6	83.6	87.2
2	-Rule-p	81.5	83.2	84.5	88.8
3	-Rule-q	82.5	83.9	85.6	**89.3**
4	-semi-PR	81.5	83.1	84.6	–
5	-semi-Rule-p	81.7	83.3	84.7	–
6	-semi-Rule-q	**82.7**	**84.2**	**85.7**	–

Table 3: Accuracy (%) on SST2 with varying sizes of labeled data and semi-supervised learning. The header row is the percentage of labeled examples for training. Rows 1-3 use only the supervised data. Rows 4-6 use semi-supervised learning where the remaining training data are used as unlabeled examples. For "-semi-PR" we only report its projected solution (in analogous to q) which performs better

References

Basu, S., Bilenko, M., and Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *KDD*, pages 59–68.

Ganchev, K., Gracca, J., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049.

Hu, Z., Ma, X., Liu, Z., Hovy, E. H., and Xing, E. P. (2016). Harnessing deep neural networks with logic rules. In *ACL*.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Palmer, M., Kingsbury, P., and Gildea, D. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Roth, D. and Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. In *CoNLL*, pages 1–8.