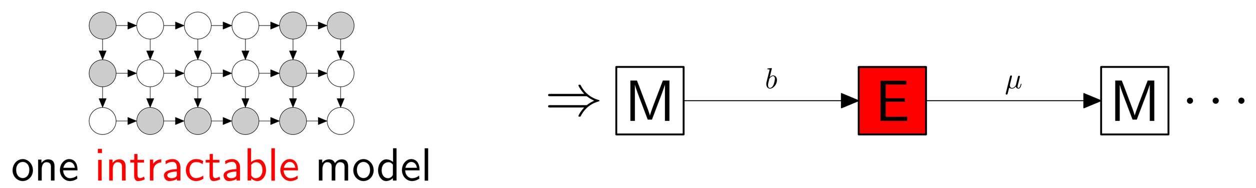
Agreement-Based Learning

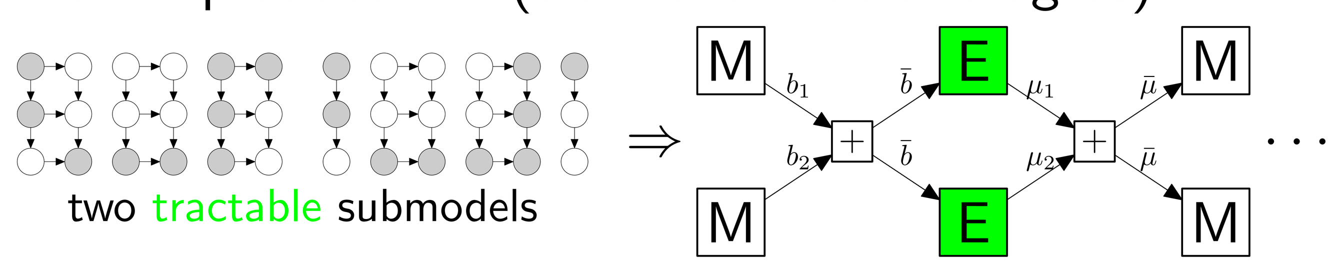
Percy Liang, Dan Klein, Michael I. Jordan
UC Berkeley • Computer Science Division

1 minute summary

Problem: learning complex hidden-variable models
Traditional solution: approximate EM



Our solution: product EM (train submodels to agree)

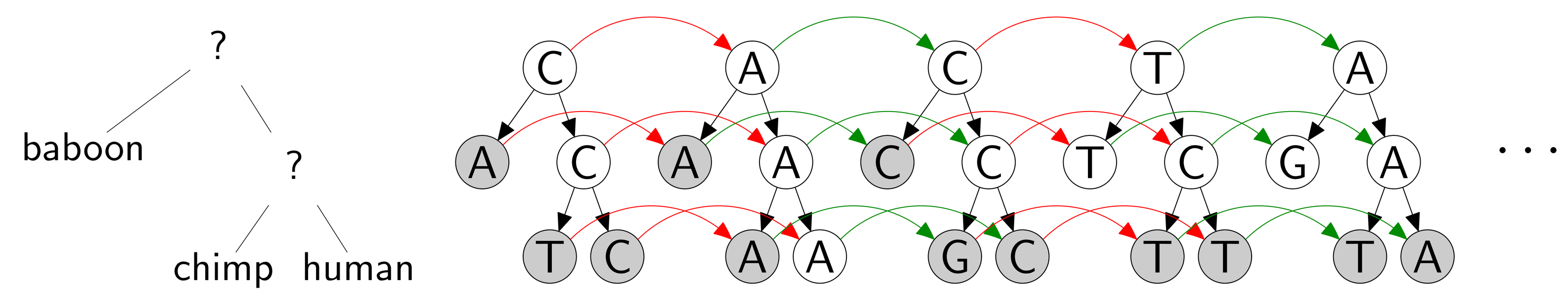


Applications: unsupervised NLP, phylogenetic HMMs

Motivating applications

Phylogenetic HMMs

Goal: model both nucleotide mutations across species and dependencies between adjacent sites

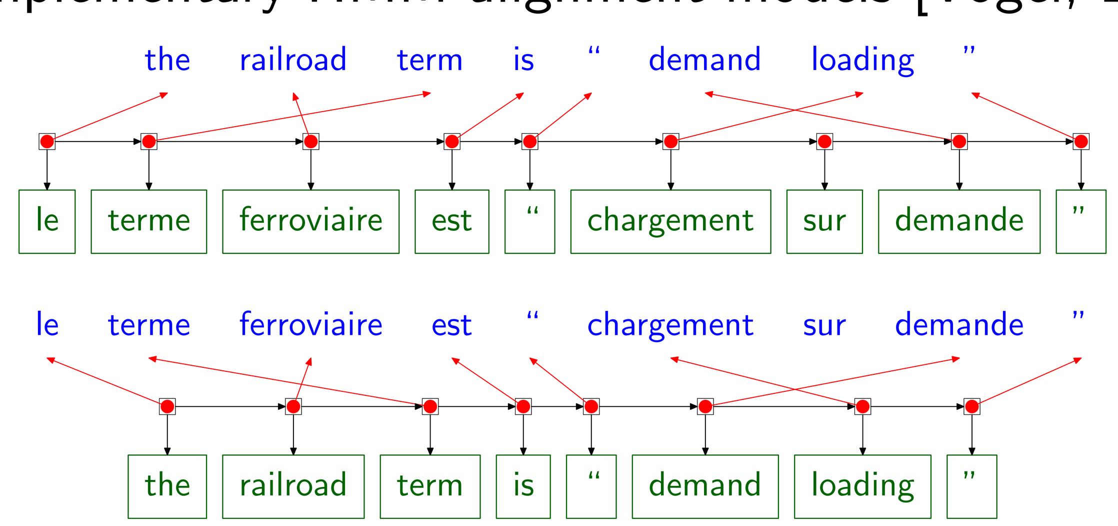


Computational challenge: doing inference in a loopy graph
Agreement-based solution:
Break up model into the red part and the green part

Unsupervised word alignment

Goal: learn to output a matching between two sequences by modeling the translation process of words between a pair of sentences

Computational challenge: enumerating all matchings
Agreement-based solution:
Two complementary HMM alignment models [Vogel, 1996]:



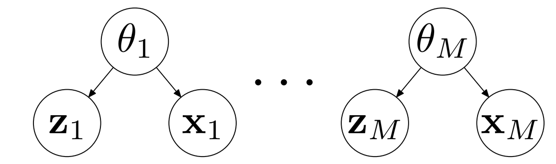
Product EM

Setup:
M submodels $\{p_m(\mathbf{x},\mathbf{z};\theta_m) : m = 1,\dots,M\}$

Objective function:

$$\mathcal{O}_{\text{agree}}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \log \sum_{\mathbf{z}} \prod_m p_m(\mathbf{x},\mathbf{z};\theta_m)$$

Interpretation:
Each submodel m independently generates $(\mathbf{x}_m,\mathbf{z}_m)$

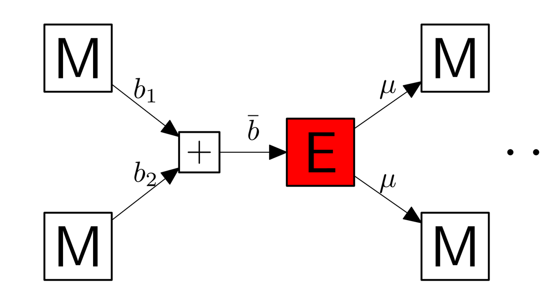


$$\mathcal{O}_{\text{agree}}(\boldsymbol{\theta}) = p(\mathbf{x}_1 = \cdots = \mathbf{x}_M = \mathbf{x}, \mathbf{z}_1 = \cdots = \mathbf{z}_M; \boldsymbol{\theta})$$

Algorithm:
Introduce auxiliary q, use Jensen's inequality:

$$\mathcal{O}_{\text{agree}} \ge \mathcal{L}(\boldsymbol{\theta}, q) \stackrel{\text{def}}{=} \sum_m \mathbb{E}_q \log p_m(\mathbf{x},\mathbf{z};\theta_m) + H(q)$$

E-step: $q(\mathbf{z}) \propto \prod_m p_m(\mathbf{x},\mathbf{z};\theta_m)$
M-step: $\theta_m = \text{argmax}_{\theta'_m} \mathbb{E}_q \log p(\mathbf{x},\mathbf{z};\theta'_m)$



Properties:
- E-step couples submodels: could be intractable
- M-step decomposes into M tractable steps

Exponential family formulation

Assume submodels are in exponential family:

$$p_m(\mathbf{x},\mathbf{z};\theta_m) = \exp\left\{\theta_m^T \left(\phi_m^{\mathcal{X}}(\mathbf{x})\phi^{\mathcal{Z}}(\mathbf{z})\right) - A_m(\theta_m)\right\}$$
for $\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Z}_m$ and 0 otherwise

Reformulation of Product EM:

Aggregate parameters: $b = \sum_m b_m, b_m = \phi_m^{\mathcal{X}}(\mathbf{x})^T\theta_m$
E-step: compute expected sufficient statistics

$$\mu = E(b, \cap_m \mathcal{Z}_m) \stackrel{\text{def}}{=} \mathbb{E}_{q(\mathbf{z};b)}\phi^{\mathcal{Z}}(\mathbf{z}) \text{ with support } \cap_m \mathcal{Z}_m$$

M-step: set θ_m to match moments $\phi_m^{\mathcal{X}}(\mathbf{x})\mu$

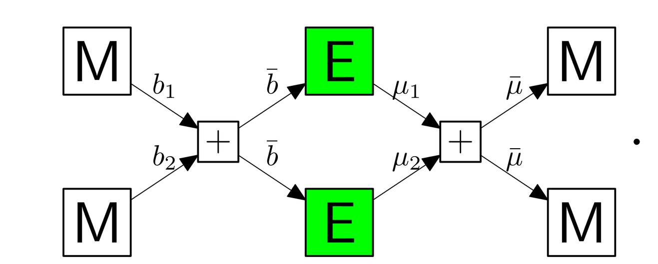
Approximate product EM

Two sources of intractability in the E-step:
- Domain $\mathcal{Z} = \cap_m \mathcal{Z}_m$ is unwieldy (e.g., matchings)
- Parameters b result in high tree-width graph

New objective function:
- A function of sufficient statistics μ_m and parameters θ_m for each submodel $m = 1,\dots,M$
- See paper for some preliminary bounds

Algorithm:

Aggregate parameters: $b = \sum_m b_m$
E-step: compute statistics $\mu_m = E(b', \mathcal{Z}')$
Aggregate statistics: $\bar{\mu} = \frac{1}{M}\sum_m \mu_m$
M-step: set θ_m to match moments $\phi_m^{\mathcal{X}}(\mathbf{x})\bar{\mu}$



Choices for E-steps:
- Domain-approximate product EM: $b' = b, \mathcal{Z}' = \mathcal{Z}_m$ (used for word alignment)
- Parameter-approximate product EM: $b = Mb_m, \mathcal{Z}' = \mathcal{Z}$ (used for phylogenetic HMMs)

Properties:
- E-step decomposes into M tractable steps now
- M-step decomposes into M tractable steps as in product EM

Experimental results

- Phylogenetic HMMs: agreement-based learning yields faster convergence
- Unsupervised word alignment: agreement-based learning yields best published results