A view of the EM algorithm that justifies incremental, sparse, and other variants

R. M. Neal, G. E. Hinton
presented by Silvia Zuffi

Introduction

- **Context**: y observed, z hidden $P(y, z|\theta)$
- **Goal**: ML parameter learning with hidden variables given observed data
- ML estimate: maximize log likelihood

$$\hat{\theta}_{ML} = \arg\max_{\theta} L(\theta)$$

$$L(\theta) = log \sum_{z} P(y, z|\theta)$$

Expectation-Maximization

- Initialization: assign parameter estimate
- **E**-STEP: computes the probability distribution of the hidden variables conditioned on observed data and actual parameters

$$\tilde{P}^{(t)}(z) = P(z|\mathbf{y}, \theta^{(t-1)})$$

- **M**-STEP maximizes

$$\theta^{(t)} = \arg\max_{\theta} E_{\tilde{P}^{(t)}}[logP(y, z|\theta)]$$

Expected complete log likelihood

- It is assumed that this maximization is easier

$$\hat{\theta}_{ML} = \arg\max_{\theta} \sum_{z} [(log \prod_{i} P(y_i, z|\theta^{(t)}) \, P(z|\mathbf{y}, \theta^{(t-1)})]$$

$$Q(\tilde{P}, \theta^{(t)}) = \sum_{z} [(log \prod_{i} P(y_i, z|\theta^{(t)}) \tilde{P}(z)]$$

$$\tilde{P}(z) = P(z|\mathbf{y}, \theta^{(t-1)})$$

Alternative solutions

- If M-step cannot be solved in closed form:
  - **partial M-step**: increase likelihood instead of maximize (GEM algorithms, Dempster 1977)
- Speed-up convergence:
  - **incremental** (partial) **E-step**

a different view of EM

- See the algorithm as an optimization problem over a function of the parameters and distribution of the hidden variables
- Both the steps optimize this function

$$F(\tilde{P}, \theta) = E_{\tilde{P}}[logP(y, z|\theta)] + H(\tilde{P})$$

$$H(\tilde{P}) = -E_{\tilde{P}}[\tilde{P}]$$

$$F(\tilde{P}, \theta) = Q(\tilde{P}, \theta) + H(\tilde{P})$$

$$\tilde{P}(z) = P(z|\mathbf{y}, \theta^{(t-1)})$$

Observation

- Jensen's inequality applied to the log function implies:

$$L(\theta) \geq Q(\tilde{P}, \theta) + H(\tilde{P})$$

$$L(\theta) = Q(\tilde{P}, \theta) + H(\tilde{P}) + KL(\tilde{P}, P_\theta) = F(\tilde{P}, \theta) + KL(\tilde{P}, P_\theta)$$

$$P_\theta(z) = P(z|\mathbf{y}, \theta)$$

Key point

Theorem 2 *If $F(\widetilde{P}, \theta)$ has a local maximum at $\widetilde{P}^*$ and θ^*, then $L(\theta)$ has a local maximum at θ^* as well. Similarly, if F has a global maximum at $\widetilde{P}^*$ and θ^*, then L has a global maximum at θ^*.*

EM Algorithm

- Initialize $\theta^{(0)}$
- E-step

$$\tilde{P}^{(t)} = \arg\max_{\tilde{P}} F(\tilde{P}, \theta^{(t-1)})$$

- M-step

$$\theta^{(t)} = \arg\max_{\theta} F(\tilde{P}^{(t)}, \theta)$$

$$\theta^{(t)} = \arg\max_{\theta} E_{\tilde{P}(t)}[logP(y, z|\theta)]$$

Different versions

This view of the EM algorithm justifies different algorithms that have been proposed:

- Incremental
- Sparse
- “Winner take all”

1. Incremental algorithms

- Independent data allow factorization

$$P(\mathbf{y},\mathbf{z}|\theta)=\prod_i P(y_i,z_i|\theta)$$

$$P(z_i|\mathbf{y},\theta)=\tilde{P}_i(z_i) \qquad \tilde{P}(\mathbf{z})=P(\mathbf{z}|\mathbf{y},\theta)=\prod_i \tilde{P}_i(z_i)$$

$$F(\tilde{P},\theta)=\sum_i F_i(\tilde{P},\theta)$$

$$Q(\tilde{P},\theta)=E_{\tilde{P}}[logP(\mathbf{y},\mathbf{z}|\theta)]=E_{\tilde{P}}[log\prod_i P(y_i,z_i|\theta)]=$$

$$E_{\tilde{P}}[\sum_i logP(y_i,z_i|\theta)]=\sum_i E_{\tilde{P}_i}[logP(y_i,z_i|\theta)]=\sum_i Q_i(\tilde{P}_i,\theta)$$

Incremental EM

- Initialize $(\theta^{(0)}, \tilde{P}^{(0)})$
- E-step: choose some data item i to update; set

$$\tilde{P}_i^{(t)} = \arg\max_{\tilde{P}_i} F_i(\tilde{P}_i, \theta^{(t-1)})$$

- M-step: set

$$\theta^{(t)} = \arg\max_{\theta} Q(\tilde{P}^{(t)}, \theta)$$

Incremental EM
Sufficient Statistics

- If the model probability distribution is in the exponential family, E- and M- steps can be implemented in terms of sufficient statistics
- E-step: compute expectations of sufficient statistics
- M-step: compute ML parameters from sufficient statistics

Incremental EM

Sufficient Statistics

$s(y, z) = \sum_i s_i(y_i, z_i)$ Vector of s.s. for N data items

- E-step: update s.s. for data item i

$$\tilde{s}_i^{(t)} = E_{\tilde{P}_i}[s_i(y_i, z_i)]$$

$$\tilde{s}^{(t)} = \tilde{s}^{(t-1)} - \tilde{s}_i^{(t-1)} + \tilde{s}_i^{(t)}$$

- M-step: compute ML parameters from sufficient statistics

Incremental EM

Approximate Sufficient Statistics

Nowlan, 1991

- GOAL: faster convergence
- E-step: update s.s. for data item i

$$\tilde{s}_i^{(t)} = E_{\tilde{P}_i}[s_i(y_i, z_i)]$$

$$\tilde{s}^{(t)} = \gamma \tilde{s}^{(t-1)} + \tilde{s}_i^{(t)} \qquad 0 < \gamma < 1$$

- M-step: compute ML parameters from sufficient statistics

Comparison

Between Incremental EM and Incremental EM with decay

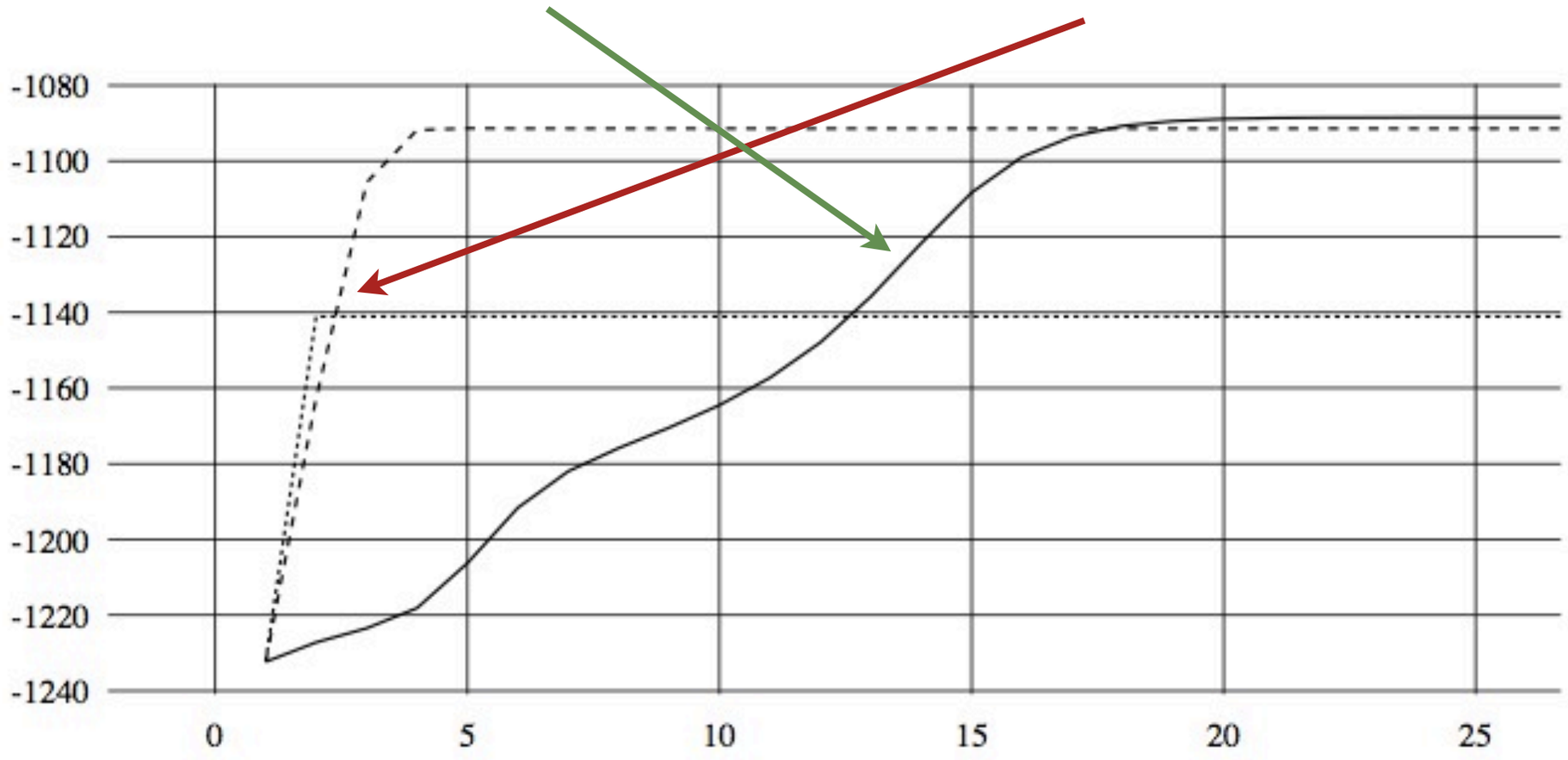


Figure 2. Convergence rates of the algorithm using exponentially decayed statistics with $\gamma = 0.99$ (dashed line) and $\gamma = 0.95$ (dotted line). For comparison, the performance of the incremental algorithm (solid line) is reproduced as well (as in Figure 1).

Comparison

Between EM and Incremental EM

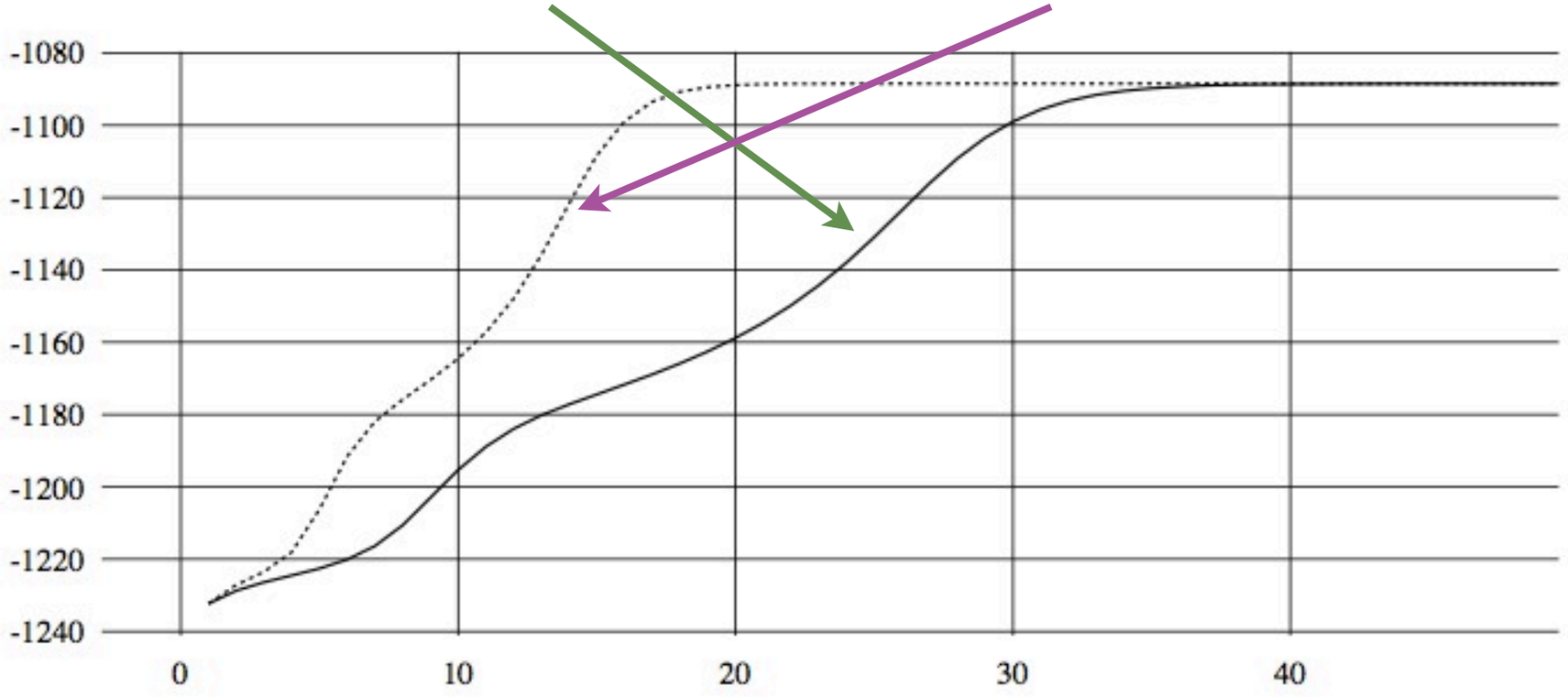


Figure 1. Comparison of convergence rates for the standard EM algorithm (solid line) and the incremental algorithm (dotted line). The log likelihood is shown on the vertical axis, the number of passes of the algorithm on the horizontal axis.

2. Sparse EM

- GOAL: faster convergence
- **Context**: small set of "plausible" values for the hidden variables (given observed data and parameter estimate)
- **Idea**: "freeze" the probabilities of the implausible values for many iterations
- Needs some heuristics

3. “Winner-take-all”

- **Idea**: at each E-step, all hidden variables but one take zero probability
- Does not find a maximum for the likelihood
- May be useful at the first stages (?)
- K-means (hard clustering)

Conclusion

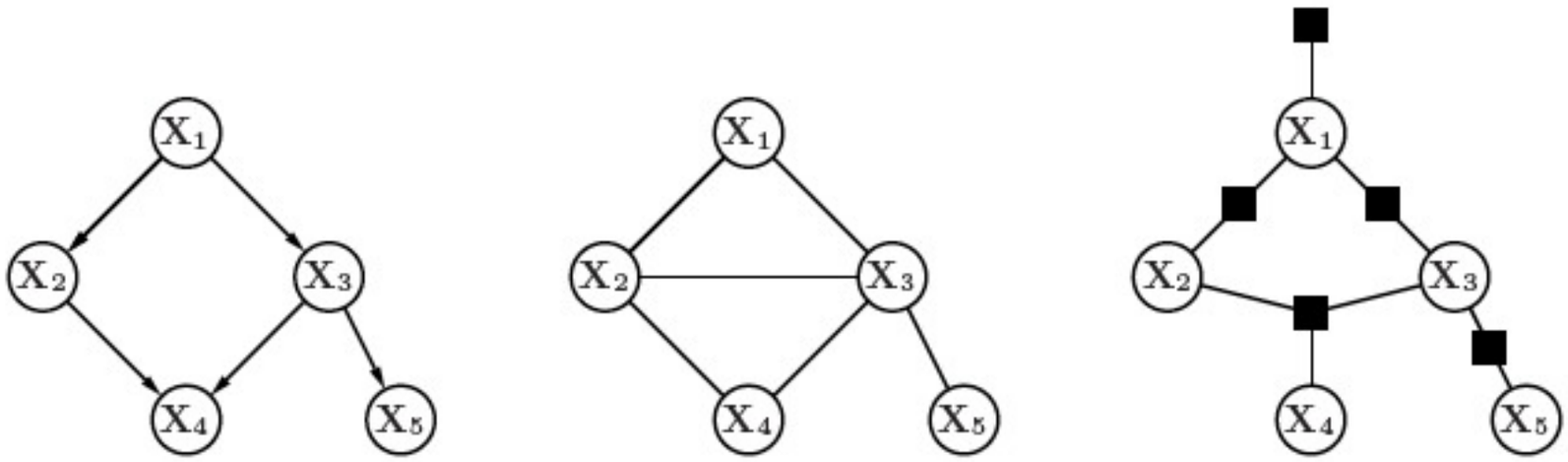
- Seeing EM as a maximization-maximization algorithm allows to perform E-step and M-step in different ways
- Convergence to the maximum of the likelihood is not guarantee

A tutorial on learning with Bayesian networks

part 3-6

D. Heckerman

Bayesian networks (DAG)



Bayesian Network **Markov Random Field** **Factor Graph**

©L. Sigal

Characterized by:

- Structure
- Local probabilities

$$p(\mathbf{x}) = \prod_i p(x_i|\mathbf{pa}_i)$$

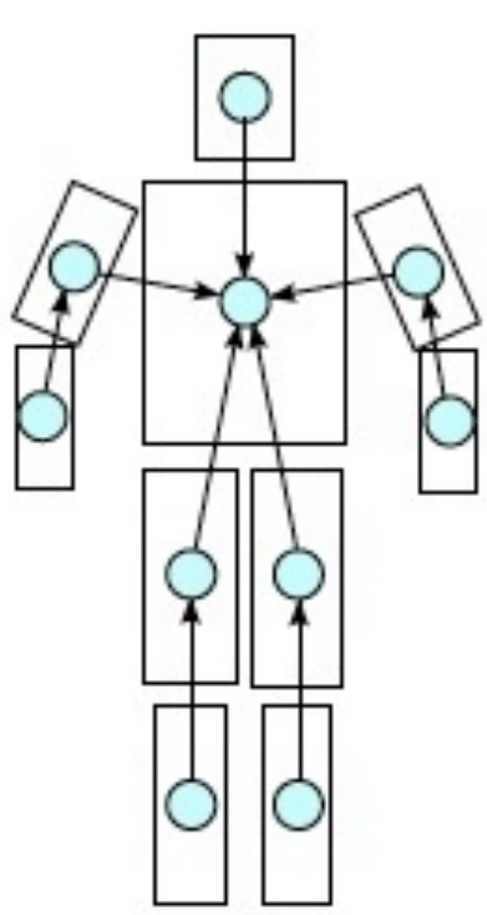
joint distribution
n variables

Bayesian networks (DAG)

- Learning model structure (model selection)
- Learning model local probabilities (parameter learning), given the structure and data
- Inference: find the value or distribution of hidden variables given the observations (param. learning can be seen as inference)

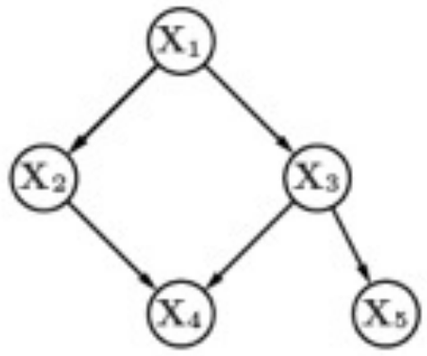
Structure

- Draw arcs from cause variables to their immediate effects



©S.Johnson

Learn Probabilities



- **Context**: $p(\mathbf{x}|\theta_S, S^h) = \prod_i p(x_i|\mathbf{pa}_i, \theta_i, S^h)$

$$\theta_S = (\theta_1, ..., \theta_n)$$

- **Goal**: given a random sample D, compute the posterior $p(\theta_S|D, S^h)$

- From complete data
- From incomplete data

Learn Probabilities from complete data

- Example: multinomials, discrete variables:

$$p(x_i^k | pa_i^j, \theta_i, S^h) = \theta_{ijk}$$

i index of variable
k = value of variable [$1....r_i$]
j=value of pa_i [$1....q_i$]

- Prior

$$\theta_{ij} \sim Dir(\theta_{ij} | \alpha_{ij1}, ...\alpha_{ijr_i})$$

$$q_i = \prod_{X_i \in Pa_i} r_i$$

- Posterior

$$p(\theta_{ij} | D, S^h) = Dir(\theta_{ij} | \alpha_{ij1} + N_{ij1}, ...\alpha_{ijr_i} + N_{ijr_i})$$

Learn Probabilities from incomplete data

- Complex posterior, intractable to compute, approximations are required
  - Gibbs sampling
  - Gaussian approximation
  - EM

Gibbs sampling

- GOAL: compute the expectation of a function given a joint probability

$$p(\mathbf{x}) \qquad E[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})$$

- **Idea**: sample each variable from the conditional

$$p(x_i|x_1, ...x_{i-1}, x_{i+1}, ...x_n)$$

- then compute the value for f() and take the mean

$$E_{p(\mathbf{x})}[f(\mathbf{x})] = \frac{1}{N}\sum_k f(\mathbf{x}_k)$$

Gibbs sampling

$$p(\theta_{ij}|D, S^h) = Dir(\theta_{ij}|\alpha_{ij1} + N_{ij1}, ...\alpha_{ijr_i} + N_{ijr_i})$$

- Example: use Gibbs sampling to compute the posterior for the multinomial example
- **Idea**: assign values to the unobserved variables according to some distribution
  →get a complete dataset
- Compute posterior
- Iterate and take the mean

Gaussian Approximation

- Large sample size
- Posterior approximate to a multivariate Gaussian with mean value = MAP

$$g(\theta_S) \equiv log(p(D|\theta_S)p(\theta_S)) = log(p(\theta_S|D) + k$$

$$\tilde{\theta}_S = \arg\max_{\theta_S} g(\theta_S) = \arg\max_{\theta_S} p(\theta_S|D)$$

$$\frac{\partial g(\theta_S)}{\partial \theta_S}|_{\tilde{\theta}_S} = 0$$

$$p(\theta_S|D, S^h) \approx p(D|\tilde{\theta}_S, S^h)p(\tilde{\theta}_S|S^h)exp(-\frac{1}{2}(\theta_S - \tilde{\theta}_S)A(\theta_S - \tilde{\theta}_S))$$

EM

- For large datasets, can use MAP or ML
- Use EM to find local MAP or ML estimates