EM for HMMs a.k.a. The Baum-Welch Algorithm

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

April 11th, 2007

The general learning problem with missing data

- Marginal likelihood – **x** is observed, **z** is missing:

$$\begin{aligned}
\ell(\theta : \mathcal{D}) &= \log \prod_{j=1}^{m} \underbrace{P(\mathbf{x}_j \mid \theta)}_{\text{observed parts}} \\
&= \sum_{j=1}^{m} \log P(\mathbf{x}_j \mid \theta) \\
&= \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)
\end{aligned}$$

sum over (marginalize out) hidden vars

EM is coordinate ascent

$$\ell(\theta : \mathcal{D}) \geq F(\theta, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

- **M-step**: Fix Q, maximize F over θ (a lower bound on $\ell(\theta : \mathcal{D})$):

$$\ell(\theta : \mathcal{D}) \geq F(\theta, Q^{(t)}) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) + m.H(Q^{(t)})$$

- **E-step**: Fix θ, maximize F over Q:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) = \ell(\theta^{(t)} : \mathcal{D}) - m \sum_{j=1}^{m} KL\left(Q(\mathbf{z} \mid \mathbf{x}_j) || P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\right)$$

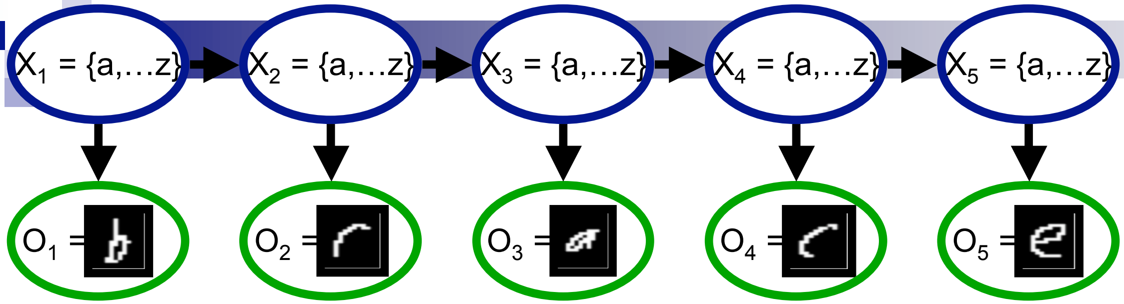
  - "Realigns" F with likelihood:

$$F(\theta^{(t)}, Q^{(t+1)}) = \ell(\theta^{(t)} : \mathcal{D})$$

What you should know about EM

- K-means for clustering:
  - algorithm
  - converges because it's coordinate ascent
- EM for mixture of Gaussians:
  - How to "learn" maximum likelihood parameters (locally max. like.) in the case of unlabeled data
- Be happy with this kind of probabilistic analysis
- Remember, E.M. can get stuck in local minima, and empirically it <u>DOES</u>
- EM is coordinate ascent
- General case for EM

Learning HMMs from fully observable data is easy



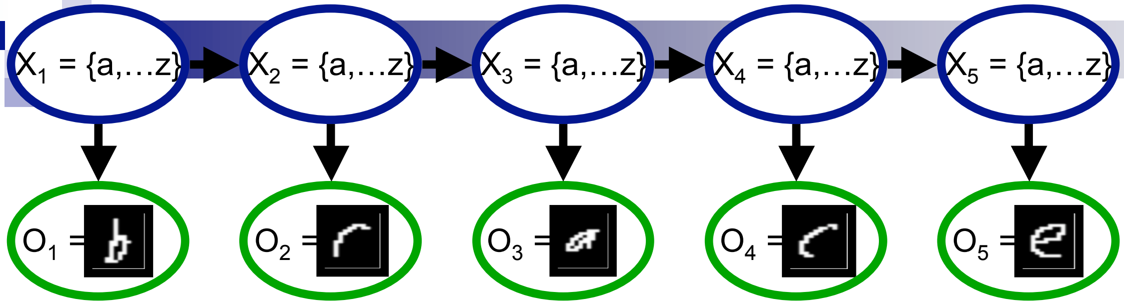
Learn 3 distributions:

$$P(X_1)$$

$$P(O_i \mid X_i)$$

$$P(X_i \mid X_{i-1})$$

Learning HMMs from fully observable data is easy



Learn 3 distributions:

$$P(X_1)$$

=a

= Count (# first letter was a)

N := dataset size

select training data where letter was a

$$P(O_i \mid X_i)$$

=pixel 17 is white

=a

= Count(pixel 17 was white, X_i=a)

any…in

$$P(X_i \mid X_{i-1})$$

=a

=b

What if **O** is observed, but **X** is hidden

Log likelihood for HMMs when **X** is hidden

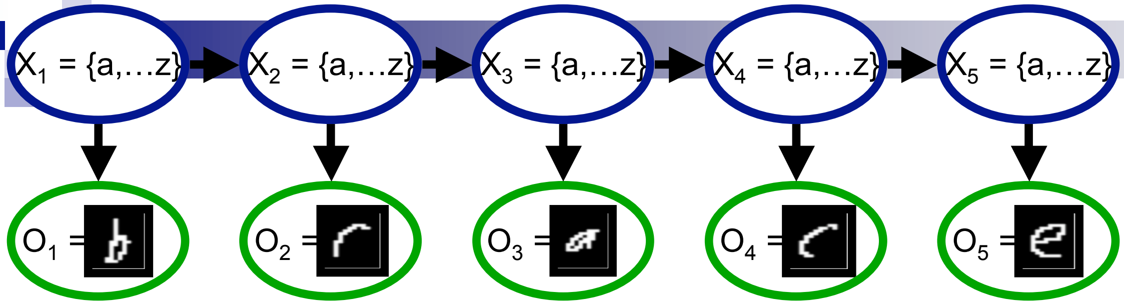
- Marginal likelihood – **O** is observed, **X** is missing
  - For simplicity of notation, training data consists of only one sequence:

$$\begin{aligned} \ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\ &= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta) \end{aligned}$$

  - If there were m sequences:

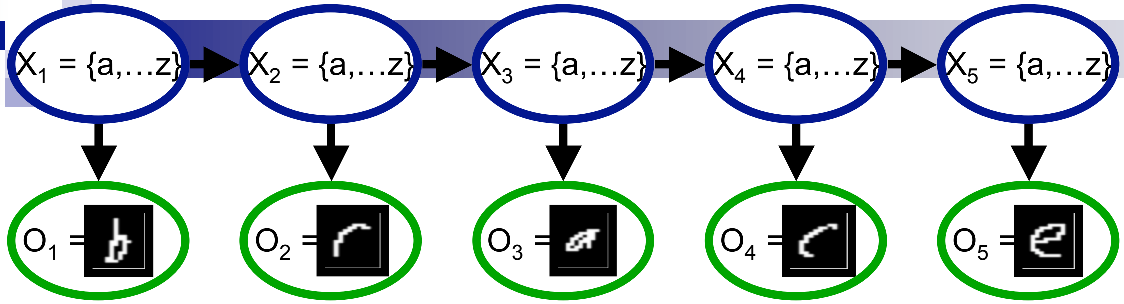
$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o}^{(j)} \mid \theta)$$

Computing Log likelihood for HMMs when **X** is hidden



$$\begin{aligned} \ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\ &= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta) \end{aligned}$$

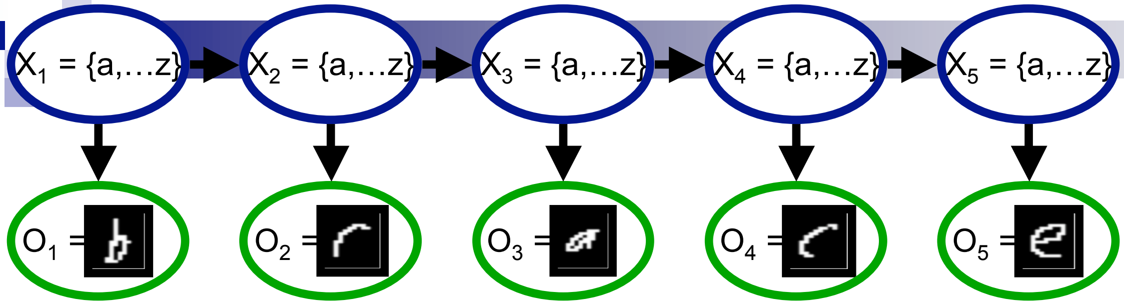
Computing Log likelihood for HMMs when **X** is hidden – variable elimination



- Can compute efficiently with variable elimination:

$$
\begin{aligned}
\ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\
&= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta)
\end{aligned}
$$

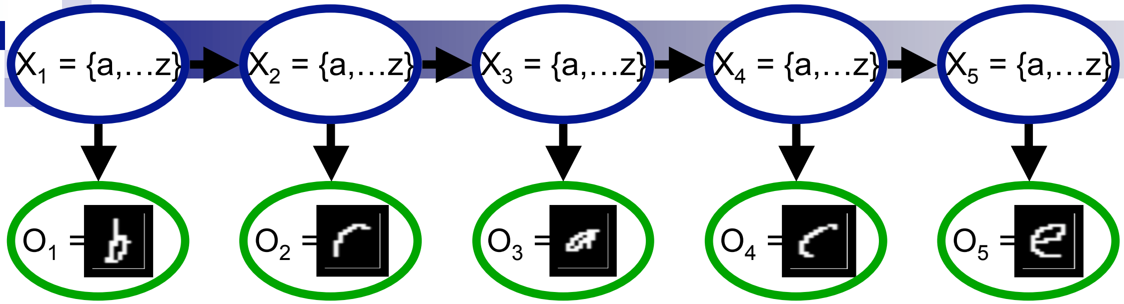
EM for HMMs when **X** is hidden



- E-step: Use inference (forwards-backwards algorithm)

- M-step: Recompute parameters with weighted data

E-step

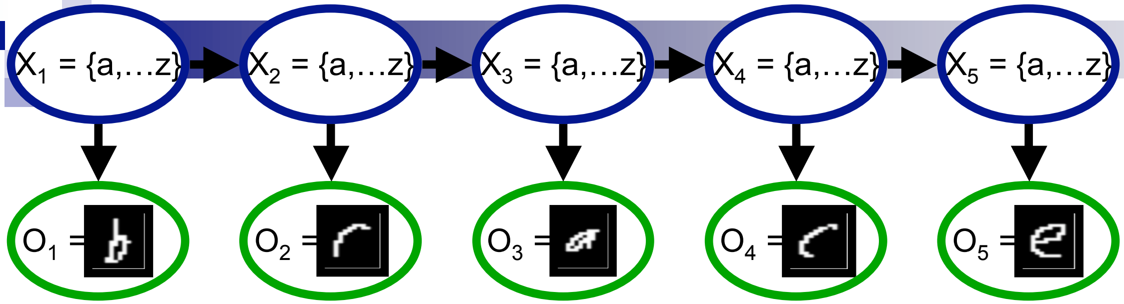


- E-step computes probability of hidden vars **x** given **o**

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \;=\; P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$

- Will correspond to inference
  - use forward-backward algorithm!

The M-step



- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{\mathbf{x}} Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \log P(\mathbf{x}, \mathbf{o} \mid \theta)$$

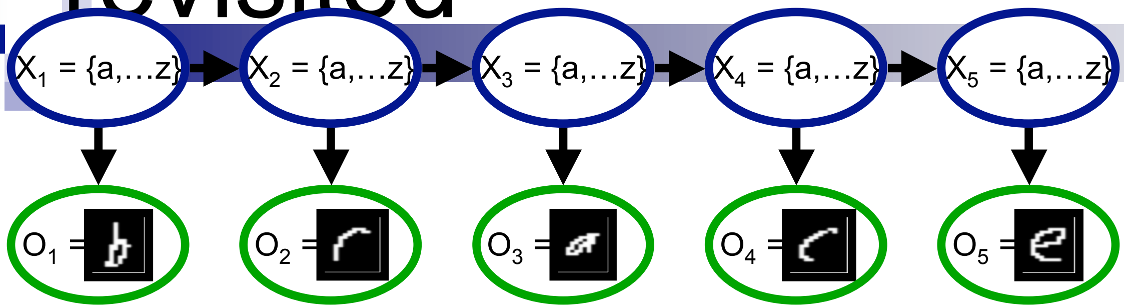
- Use expected counts instead of counts:
  - If learning requires Count(**x**,**o**)
  - Use $E_{Q(t+1)}$[Count(**x**,**o**)]

Decomposition of likelihood revisited

$$P(X_1)$$

$$P(O_i \mid X_i)$$

$$P(X_i \mid X_{i-1})$$



- Likelihood optimization decomposes:

$$\max_\theta \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log P(\mathbf{x}, \mathbf{o} \mid \theta) =$$

$$\max_\theta \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log P(x_1 \mid \theta_{X_1}) P(o_1 \mid x_1, \theta_{O|X}) \prod_{t=2}^{n} P(x_t \mid x_{t-1}, \theta_{X_t|X_{t-1}}) P(o_t \mid x_t, \theta_{O|X})$$

Starting state probability $P(X_1)$

- Using expected counts
  - $P(X_1=a) = \theta_{X1=a}$

$$\max_{\theta_{X_1}} \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log P(x_1 \mid \theta_{X_1})$$

$$\boxed{\theta_{X_1=a} = \frac{\sum_{j=1}^{m} Q(X_1 = a \mid \mathbf{o}^{(j)})}{m}}$$

Transition probability $P(X_t|X_{t-1})$

- Using expected counts
  - $P(X_t=a|X_{t-1}=b) = \theta_{Xt=a|Xt-1=b}$

$$\max_{\theta_{X_t|X_{t-1}}} \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log \prod_{t=2}^{n} P(x_t \mid x_{t-1}, \theta_{X_t|X_{t-1}})$$

$$\boxed{\theta_{X_t=a|X_{t-1}=b} = \frac{\sum_{j=1}^{m} \sum_{t=2}^{n} Q(X_t = a, X_{t-1} = b \mid \mathbf{o}^{(j)})}{\sum_{j=1}^{m} \sum_{t=2}^{n} \sum_{i=1}^{k} Q(X_t = i, X_{t-1} = b \mid \mathbf{o}^{(j)})}}$$

Observation probability $P(O_t|X_t)$

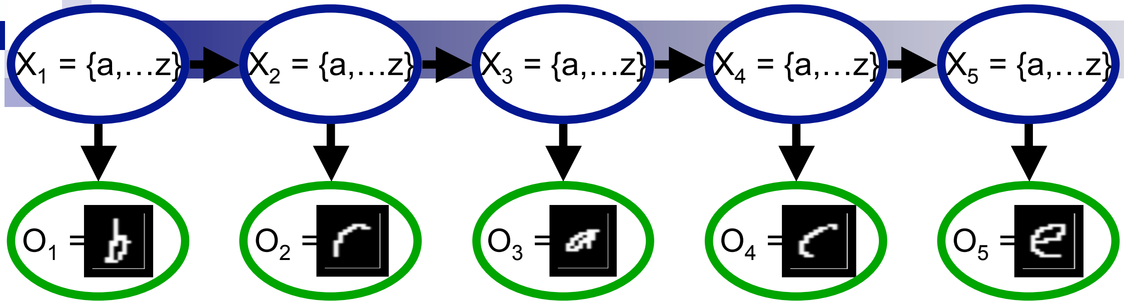
- Using expected counts
  - $P(O_t=a|X_t=b) = \theta_{Ot=a|Xt=b}$

$$\max_{\theta_{O|X}} \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log \prod_{t=1}^{n} P(o_t \mid x_t, \theta_{O|X})$$

$$\boxed{\theta_{O_t=a|X_t=b} = \frac{\sum_{j=1}^{m} \sum_{t=1}^{n} \delta(\mathbf{o}_t^{(j)} = a) Q(X_t = b \mid \mathbf{o}^{(j)})}{\sum_{j=1}^{m} \sum_{t=1}^{n} Q(X_t = b \mid \mathbf{o}^{(j)})}}$$

E-step revisited

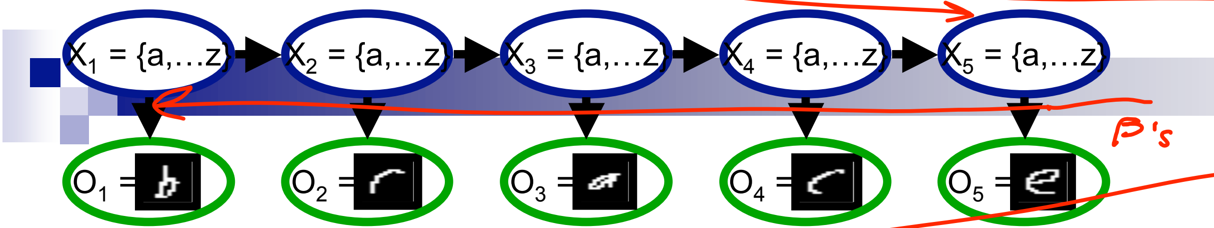
$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) = P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$



- E-step computes probability of hidden vars **x** given **o**
- Must compute:
  - $Q(x_t=a|\mathbf{o})$ – marginal probability of each position
  - $Q(x_{t+1}=a,x_t=b|\mathbf{o})$ – joint distribution between pairs of positions

The forwards-backwards algorithm

α's



$$P(X_i \mid o_{1..n})$$

drop O_1 in notation

- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$
- For i = 2 to n
  - Generate a forwards factor by eliminating X_{i-1}

sum out previous var   prob obs   transition prob

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

$\alpha_5(a)$
$\alpha_5(b)$
$\vdots$
$\alpha_5(z)$

$\alpha_n(X_n)$ normalized $= P(X_n \mid O_{1:n})$

$\beta_1(X_1)\alpha_1(X_1)$ normalized $= P(X_1 \mid O_{1:n})$

- Initialization: $\beta_n(X_n) = 1$
- For i = n-1 to 1
  - Generate a backwards factor by eliminating X_{i+1}

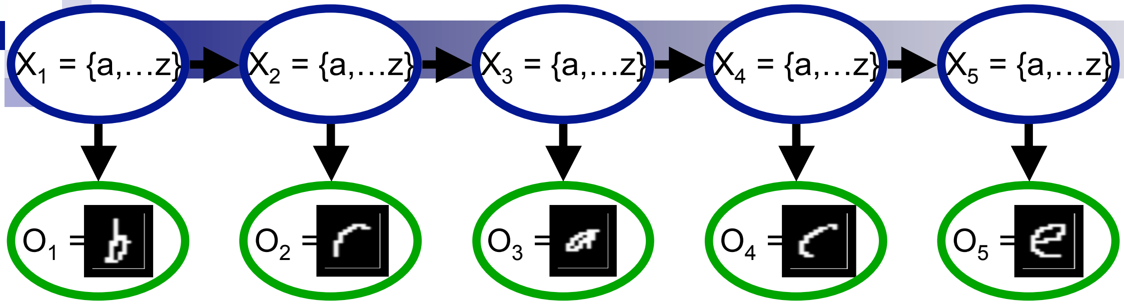
$\forall x_i$

$$\beta_i(\underbrace{X_i}_{=x_i}) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid \underbrace{X_i}_{=x_i})\beta_{i+1}(x_{i+1})$$

- 8 i, probability is: $\boxed{P(X_i \mid o_{1..n}) \cancel{=}\propto \alpha_i(X_i)\beta_i(X_i)}$

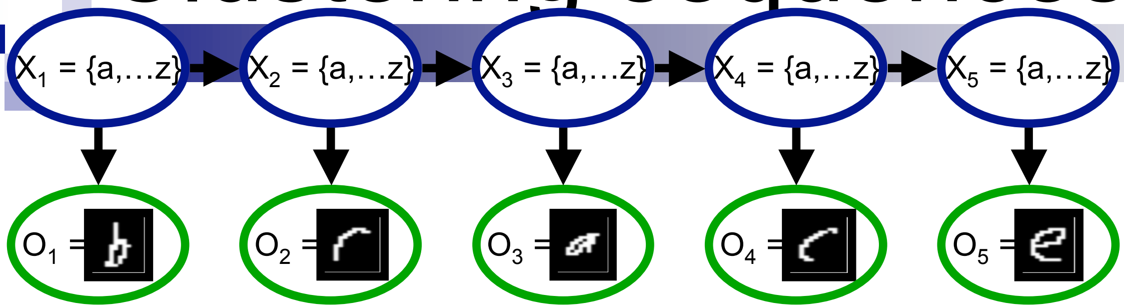
E-step revisited

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) = P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$



- E-step computes probability of hidden vars **x** given **o**
- Must compute:
  - $Q(x_t=a|\mathbf{o})$ – marginal probability of each position
    - Just forwards-backwards!
  - $Q(x_{t+1}=a,x_t=b|\mathbf{o})$ – joint distribution between pairs of positions

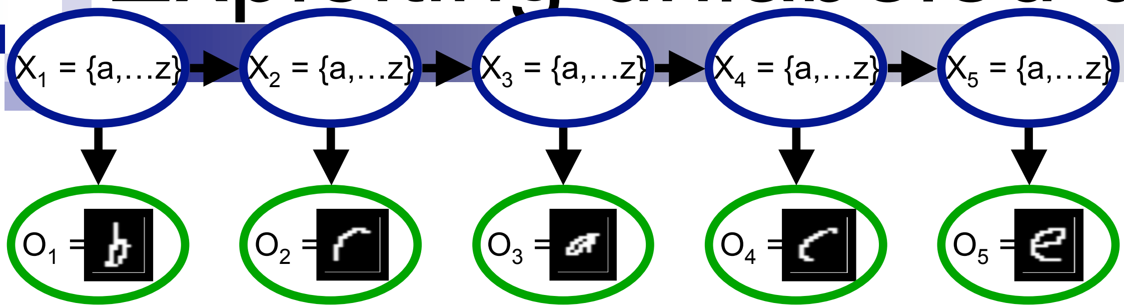
What can you do with EM for HMMs? 1 – Clustering sequences



Independent clustering:

Sequence clustering:

What can you do with EM for HMMs? 2 – Exploiting unlabeled data



- Labeling data is hard work ! save (graduate student) time by using both labeled and unlabeled data
  - Labeled data:
    - <X="brace",O= >
  - Unlabeled data:
    - <X=?????,O= >

Exploiting unlabeled data in clustering

- A few data points are labeled
  - <x,o>
- Most points are unlabeled
  - <?,o>
- In the E-step of EM:
  - If i'th point is unlabeled:
    - compute $Q(X|o_i)$ as usual
  - If i'th point is labeled:
    - set $Q(X=x|o_i)=1$ and $Q(X\neq x|o_i)=0$
- M-step as usual

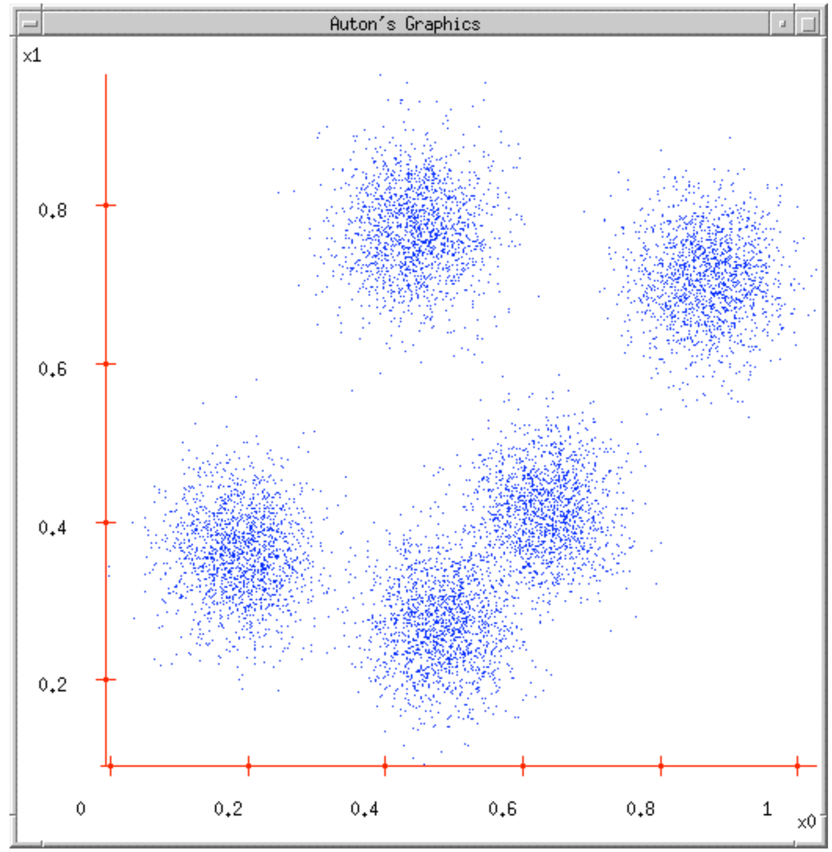
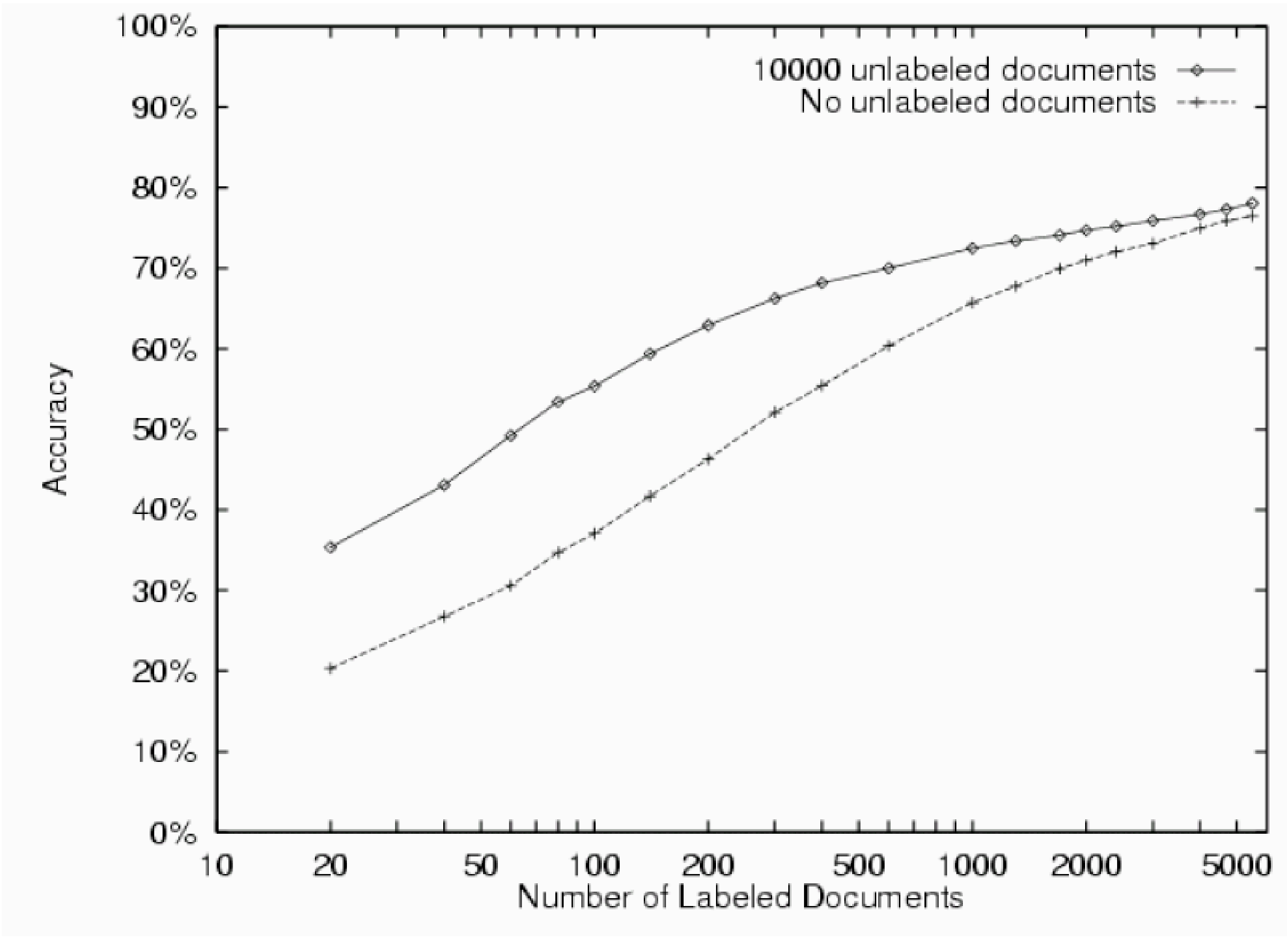


Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

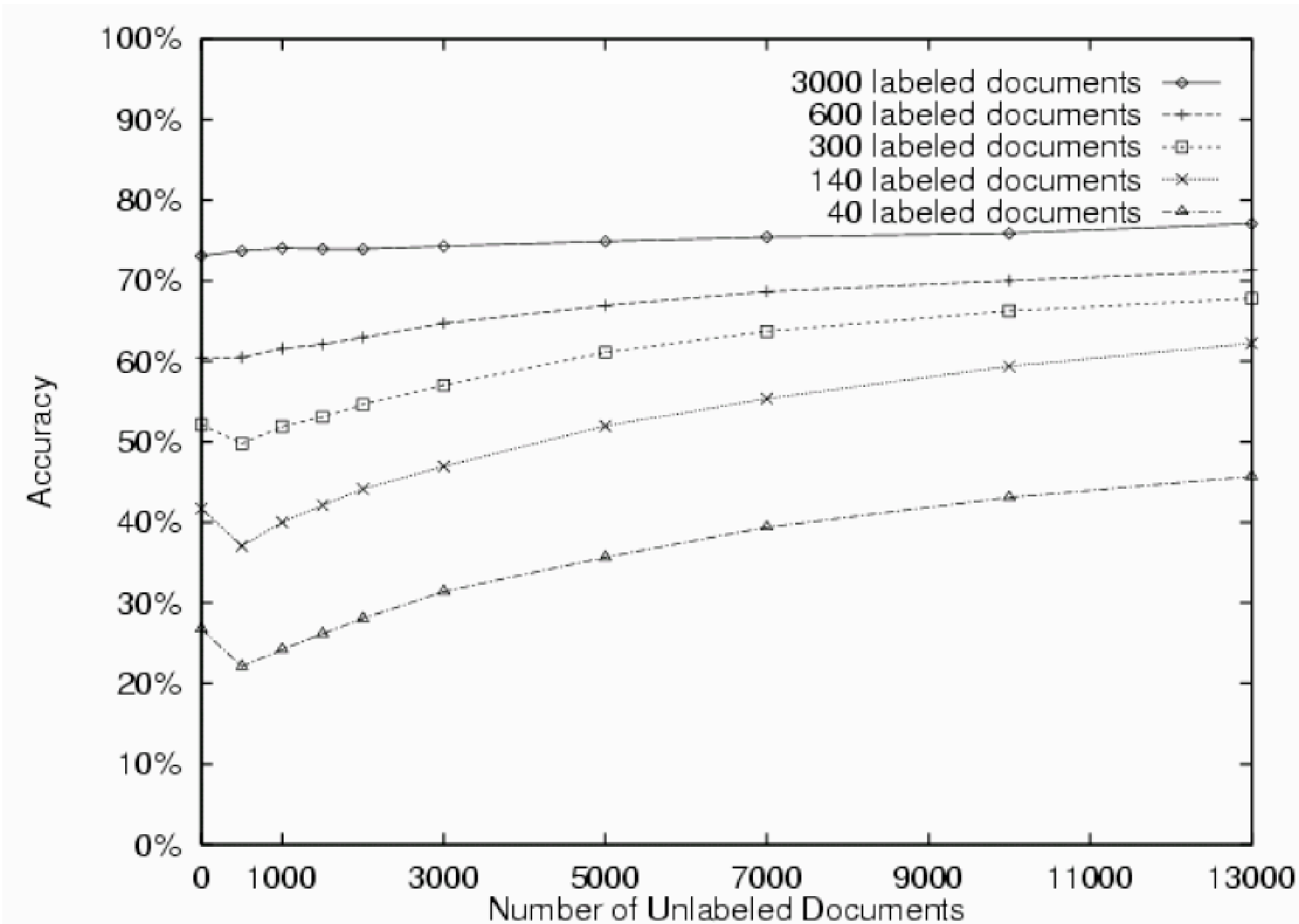
Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^{\star}$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^{\star}$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	DDam
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

Using one labeled example per class

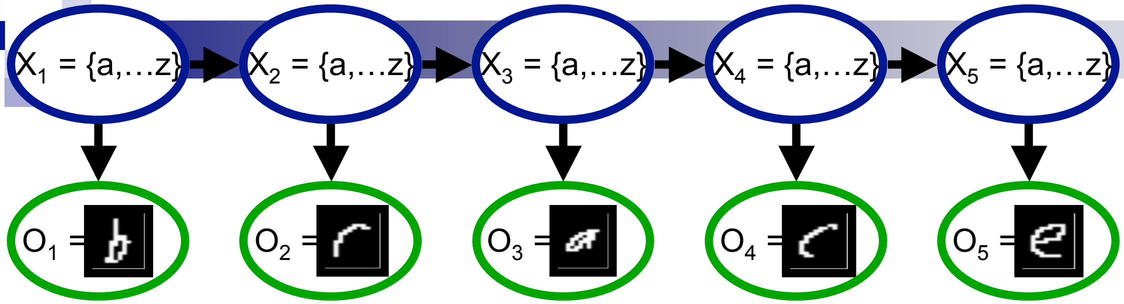
20 Newsgroups data – advantage of adding unlabeled data



20 Newsgroups data – Effect of additional unlabeled data



Exploiting unlabeled data in HMMs



- A few data points are labeled
  - <x,o>
- Most points are unlabeled
  - <?,o>
- In the E-step of EM:
  - If i'th point is unlabeled:
    - compute $Q(X|o_i)$ as usual
  - If i'th point is labeled:
    - set $Q(X=x|o_i)=1$ and $Q(X\neq x|o_i)=0$
- M-step as usual
  - Speed up by remembering counts for labeled data

What you need to know

- Baum-Welch = EM for HMMs
- E-step:
  - Inference using forwards-backwards
- M-step:
  - Use weighted counts
- Exploiting unlabeled data:
  - Some unlabeled data can help classification
  - Small change to EM algorithm
    - In E-step, only use inference for unlabeled data

Acknowledgements

- Experiments combining labeled and unlabeled data provided by Tom Mitchell