DSC291: Advanced Statistical Natural Language Processing

Unsupervised Learning

Zhiting Hu

Lecture 7, April 19, 2022

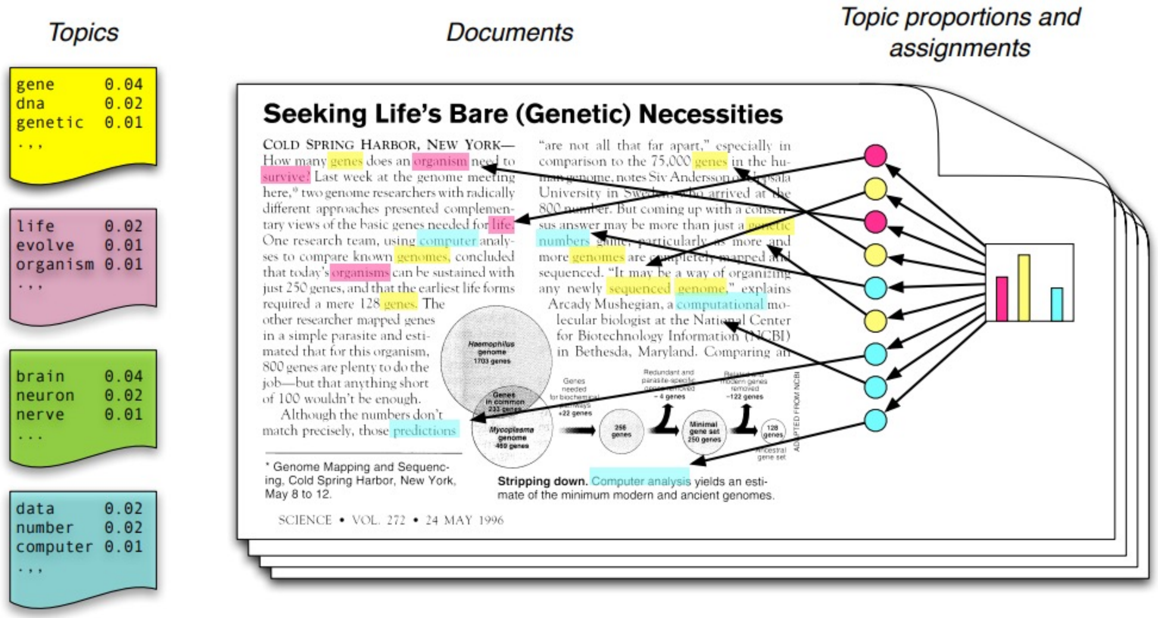
UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

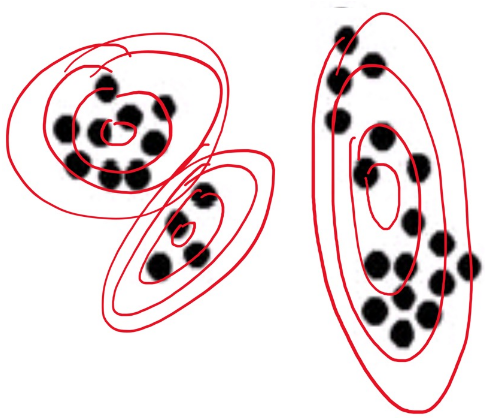
Outline

- Unsupervised Learning: Expectation Maximization

Recap: Unsupervised Learning for Representations

- For text $\boldsymbol{x}$, derive a latent representation $\boldsymbol{z}$
  - with no annotation
- Example 1: Topic models (e.g., LDA)
  - $\boldsymbol{z}$: a distribution over topics (or assignment to topics)



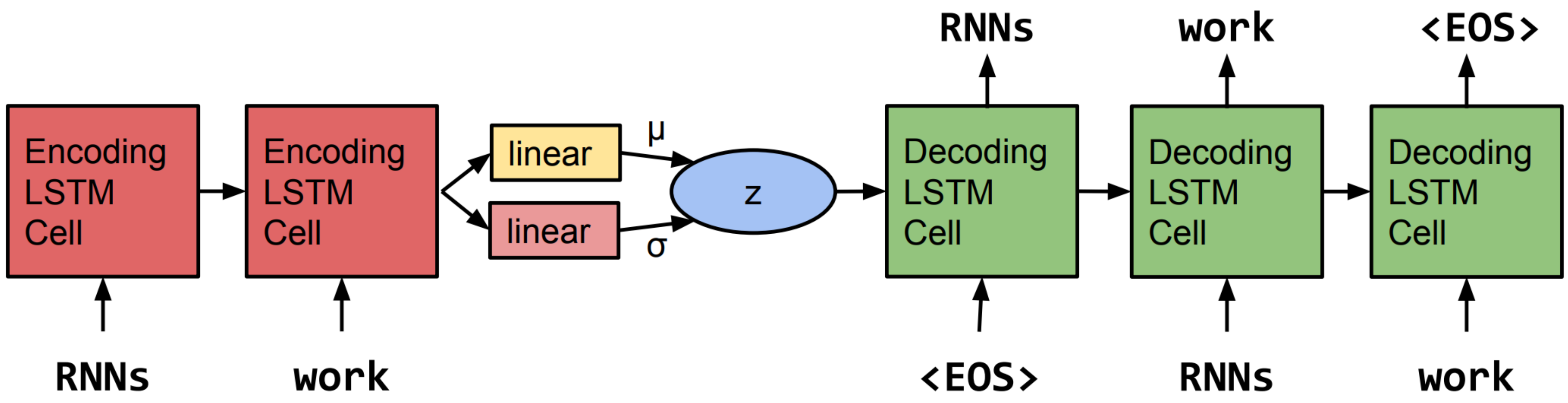


Clustering

[Blei et al., 2003, Latent Dirichlet Allocation]

Recap: Unsupervised Learning for Representations

- For text $\boldsymbol{x}$, derive a latent representation $\boldsymbol{z}$
  - with no annotation
- Example 1: Topic models (e.g., LDA)
  - $\boldsymbol{z}$: a distribution over topics (or assignment to topics)
- Example 2: Variational Autoencoders (VAEs)
  - $\boldsymbol{z}$: a dense feature vector



[Blei et al., 2003, Latent Dirichlet Allocation]

Recap: Unsupervised Learning

- Each instance has two parts:
  - observed variables $\boldsymbol{x}$
  - latent (unobserved) variables $\boldsymbol{z}$
  - A.k.a., “incomplete” data
- Want to learn a model $p_\theta(\boldsymbol{x}, \boldsymbol{z})$

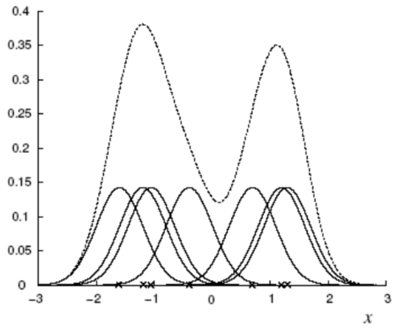
[Content adapted from CMU 10-708]

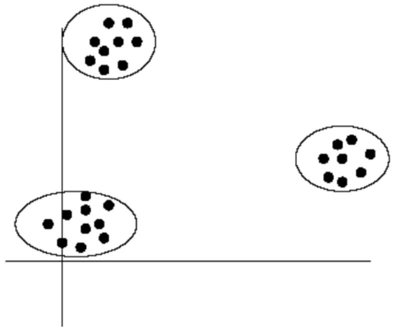
Recap: Running Example: Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

$$p(x_n|\mu,\Sigma) = \sum_k \pi_k N(x,|\mu_k,\Sigma_k)$$

mixture proportion   mixture component





- This model can be used for unsupervised clustering

Recap: Running Example: Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:
  - Z is a latent class indicator vector:

$$p(z_n) = \text{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_n^k}$$

  - X is a conditional Gaussian variable with a class-specific mean/covariance

$$p(x_n \mid z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)\right\}$$

  - The likelihood of a sample:

Parameters $\boldsymbol{\theta}$ to be learned:

mixture component

mixture proportion

$$\begin{aligned} p(x_n | \mu, \Sigma) &= \sum_k p(z^k = 1 \mid \pi) p(x, \mid z^k = 1, \mu, \Sigma) \\ &= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k} \right) = \sum_k \pi_k N(x, \mid \mu_k, \Sigma_k) \end{aligned}$$

Recap: Running Example: Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components: $p(x_n|\mu,\Sigma) = \sum_k \pi_k N(x,| \mu_k, \Sigma_k)$
- Recall MLE for completely observed data
  - Data log-likelihood: $$\ell(\boldsymbol{\theta}; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n \mid \pi) p(x_n \mid z_n, \mu, \sigma)$$
$$= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_n^k}$$
$$= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k z_n^k \frac{1}{2\sigma^2}(x_n - \mu_k)^2 + C$$
  - MLE:

$$\hat{\pi}_{k,MLE} = \arg\max_\pi \ell(\boldsymbol{\theta}; D),$$
$$\hat{\mu}_{k,MLE} = \arg\max_\mu \ell(\boldsymbol{\theta}; D) \quad \Rightarrow \quad \hat{\mu}_{k,MLE} = \frac{\sum_n z_n^k x_n}{\sum_n z_n^k}$$
$$\hat{\sigma}_{k,MLE} = \arg\max_\sigma \ell(\boldsymbol{\theta}; D)$$

- What if we do not know z_n?

Why is Learning Harder?

- Complete log likelihood: if both $\boldsymbol{x}$ and $\boldsymbol{z}$ can be observed, then

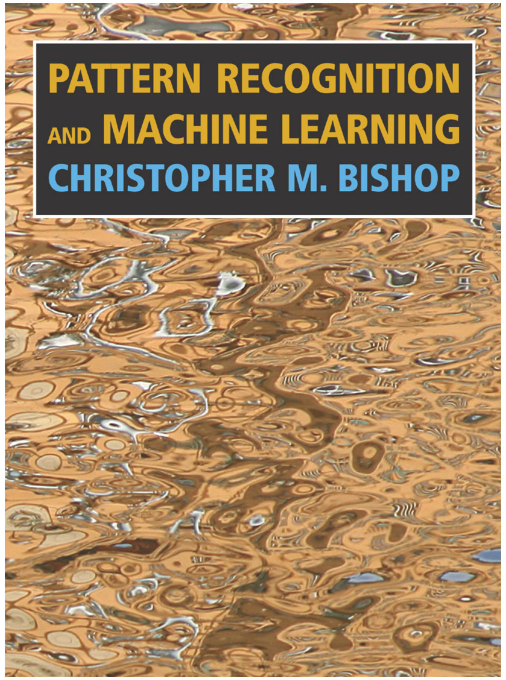
$$\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z}) = \log p(\boldsymbol{x}, \boldsymbol{z}|\theta) = \log p(\boldsymbol{z}|\theta_z) + \log p(\boldsymbol{x}|\boldsymbol{z}, \theta_x)$$

  - Decomposes into a sum of factors, the parameter for each factor can be estimated separately
- But given that $\boldsymbol{z}$ is not observed, $\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})$ is a random quantity, cannot be maximized directly
- Incomplete (or marginal) log likelihood: with $\boldsymbol{z}$ unobserved, our objective becomes the log of a marginal probability:

$$\ell(\theta; \boldsymbol{x}) = \log p(\boldsymbol{x}|\theta) = \log \sum_z p(\boldsymbol{x}, \boldsymbol{z}|\theta)$$

  - All parameters become coupled together
  - In other models when $\boldsymbol{z}$ is complex (continuous) variables (as we'll see later), marginalization over $\boldsymbol{z}$ is intractable.

Expectation Maximization (EM)



Expectation Maximization (EM)

- For any distribution $q(\boldsymbol{z}|\boldsymbol{x})$, define expected complete log likelihood:

$$\mathbb{E}_q[\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})] = \sum_z q(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$$

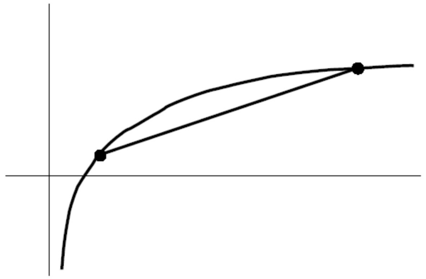
  - A deterministic function of θ
  - Inherit the factorizability of $\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})$
- Use this as the surrogate objective
- Does maximizing this surrogate $\mathbb{E}_q[\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})]$ yield a maximizer of the likelihood $\ell(\theta; \boldsymbol{x})$?

Expectation Maximization (EM)

- For any distribution $q(\boldsymbol{z}|\boldsymbol{x})$, define expected complete log likelihood:

$$\mathbb{E}_q[\ell_c(\theta;\boldsymbol{x},\boldsymbol{z})] = \sum_z q(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x},\boldsymbol{z}|\theta)$$

- Jensen's inequality



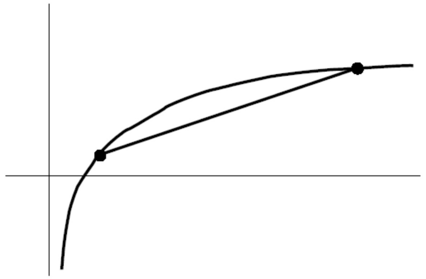
$$\begin{aligned}
\ell(\theta;x) &= \log p(x\,|\,\theta) \\
&= \log \sum_z p(x,z\,|\,\theta) \\
&= \log \sum_z q(z\,|\,x) \frac{p(x,z\,|\,\theta)}{q(z\,|\,x)} \\
&\geq \sum_z q(z\,|\,x) \log \frac{p(x,z\,|\,\theta)}{q(z\,|\,x)} \quad \text{Evidence Lower Bound (ELBO)} \\
&= \sum_z q(z\,|\,x) \log p(x,z\,|\,\theta) - \sum_z q(z\,|\,x) \log q(z\,|\,x) \\
&= \mathbb{E}_q[\ell_c(\theta;\boldsymbol{x},\boldsymbol{z})] + H(q)
\end{aligned}$$

Expectation Maximization (EM)

- For any distribution $q(\boldsymbol{z}|\boldsymbol{x})$, define expected complete log likelihood:

$$\mathbb{E}_q[\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})] = \sum_z q(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$$

- Jensen's inequality



$$\begin{aligned} \ell(\theta; x) &= \log p(x \mid \theta) \\ &= \log \sum_z p(x, z \mid \theta) \\ &= \log \sum_z q(z \mid x) \frac{p(x, z \mid \theta)}{q(z \mid x)} \\ &\geq \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \end{aligned}$$

- Indeed we have

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

Lower Bound and Free Energy

- For fixed data $\boldsymbol{x}$, define a functional called the (variational) free energy:

$$F(q,\theta) = -\mathbb{E}_q[\ell_c(\theta;\boldsymbol{x},\boldsymbol{z})] - H(q) \geq -\ell(\theta;\boldsymbol{x})$$

- The EM algorithm is coordinate-decent on F
  - At each step t:
    - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right)$
    - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right)$

E-step: minimization of $F(q, \theta)$ w.r.t q

- Claim:

$$q^{t+1} = \text{argmin}_q\, F(q, \theta^t) = p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$$

  - This is the posterior distribution over the latent variables given the data and the current parameters.

- Proof (easy): recall

$$\ell(\theta^t; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta^t)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)\big)$$

Independent of q $-F(q, \theta^t)$ ≥ 0

  - $F(q, \theta^t)$ is minimized when $\text{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)\big) = 0$, which is achieved only when $q(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$

M-step: minimization of $F(q, \theta)$ w.r.t $\boldsymbol{\theta}$

- Note that the free energy breaks into two terms:

$$F(q, \theta) = -\mathbb{E}_q[\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})] - H(q) \geq \ell(\theta; \boldsymbol{x})$$

  - The first term is the expected complete log likelihood and the second term, which does not depend on q, is the entropy.
- Thus, in the M-step, maximizing with respect to θ for fixed q we only need to consider the first term:

$$\theta^{t+1} = \text{argmax}_\theta \, \mathbb{E}_q[\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})] = \text{argmax}_\theta \sum_z q^{t+1}(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\theta)$$

  - Under optimal q^{t+1}, this is equivalent to solving a standard MLE of fully observed model $p(\boldsymbol{x}, \boldsymbol{z}|\theta)$, with z replaced by its expectation w.r.t $p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$

Running Example: Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:
  - Z is a latent class indicator vector:

$$p(z_n) = \mathrm{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_n^k}$$

  - X is a conditional Gaussian variable with a class-specific mean/covariance

$$p(x_n \mid z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left\{ -\tfrac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) \right\}$$

  - The likelihood of a sample:

mixture component

mixture proportion

$$p(x_n | \mu, \Sigma) = \sum_k p(z^k = 1 \mid \pi) p(x, \mid z^k = 1, \mu, \Sigma)$$

$$= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k} \right) = \sum_k \pi_k N(x, \mid \mu_k, \Sigma_k)$$

Running Example: Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components
- The expected complete log likelihood

$$\mathbb{E}_q\left[\ell_c(\boldsymbol{\theta}; x, z)\right] = \sum_n \mathbb{E}_q\left[\log p\left(z_n \mid \pi\right)\right] + \sum_n \mathbb{E}_q\left[\log p\left(x_n \mid z_n, \mu, \Sigma\right)\right]$$

$$= \sum_n \sum_k \mathbb{E}_q\left[z_n^k\right] \log \pi_k - \frac{1}{2} \sum_n \sum_k \mathbb{E}_q\left[z_n^k\right] \left(\left(x_n - \mu_k\right)^T \Sigma_k^{-1} \left(x_n - \mu_k\right) + \log\left|\Sigma_k\right| + C\right)$$

- E-step: computing the posterior of z_n given the current estimate of the parameters (i.e., π , μ, Σ)

$$p(z_n^k = 1 \mid x, \mu^{(t)}, \Sigma^{(t)}) = \frac{\pi_k^{(t)} N(x_n, \mid \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_i \pi_i^{(t)} N(x_n, \mid \mu_i^{(t)}, \Sigma_i^{(t)})}$$

Numerator → $p(z_n^k = 1, x \mid \mu^{(t)}, \Sigma^{(t)})$

Denominator → $p(x \mid \mu^{(t)}, \Sigma^{(t)})$

Example: Gaussian Mixture Models (GMMs)

$q(z_n)$

- M-step: computing the parameters given the current estimate of z_n

$$\pi_k^* = \arg\max \langle l_c(\boldsymbol{\theta}) \rangle, \quad \Rightarrow \frac{\partial}{\partial \pi_k} \langle l_c(\boldsymbol{\theta}) \rangle = 0, \forall k, \quad \text{s.t.} \sum_k \pi_k = 1$$

$$\Rightarrow \pi_k^* = \sum_n \langle z_n^k \rangle_{q^{(t)}} \Big/ N = \sum_n \tau_n^{k(t)} \Big/ N = \langle n_k \rangle \Big/ N$$

$$\mu_k^* = \arg\max \langle l(\boldsymbol{\theta}) \rangle, \quad \Rightarrow \mu_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} x_n}{\sum_n \tau_n^{k(t)}}$$

$$\Sigma_k^* = \arg\max \langle l(\boldsymbol{\theta}) \rangle, \quad \Rightarrow \Sigma_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^T}{\sum_n \tau_n^{k(t)}}$$

Fact:

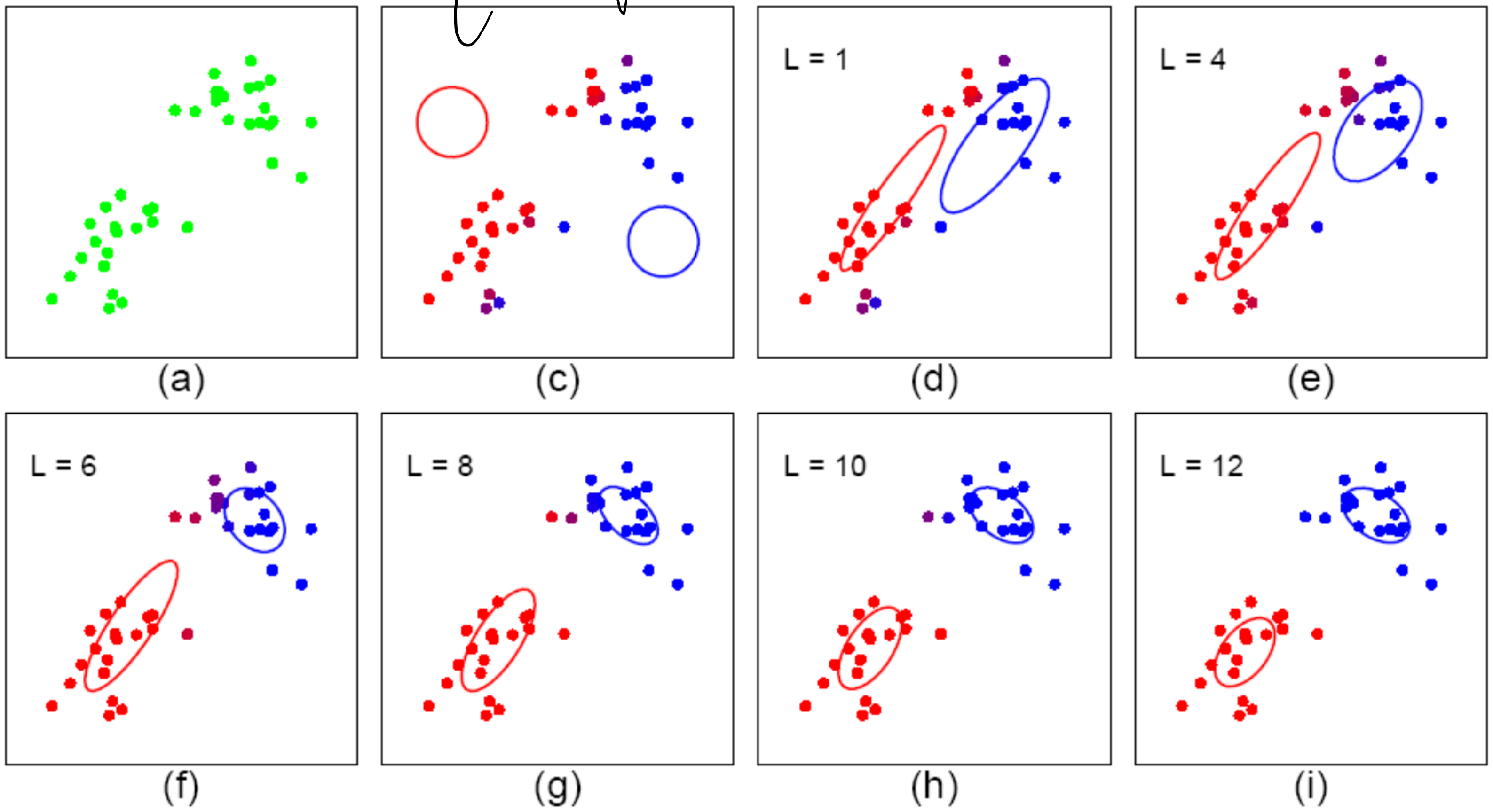
$$\frac{\partial \log |A^{-1}|}{\partial A^{-1}} = A^T$$

$$\frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial A} = \mathbf{x}\mathbf{x}^T$$

Example: Gaussian Mixture Models (GMMs)

- Start: "guess" the centroid μ_k and covariance Σ_k of each of the K clusters
- Loop:

$q(z|x_i;\theta)$



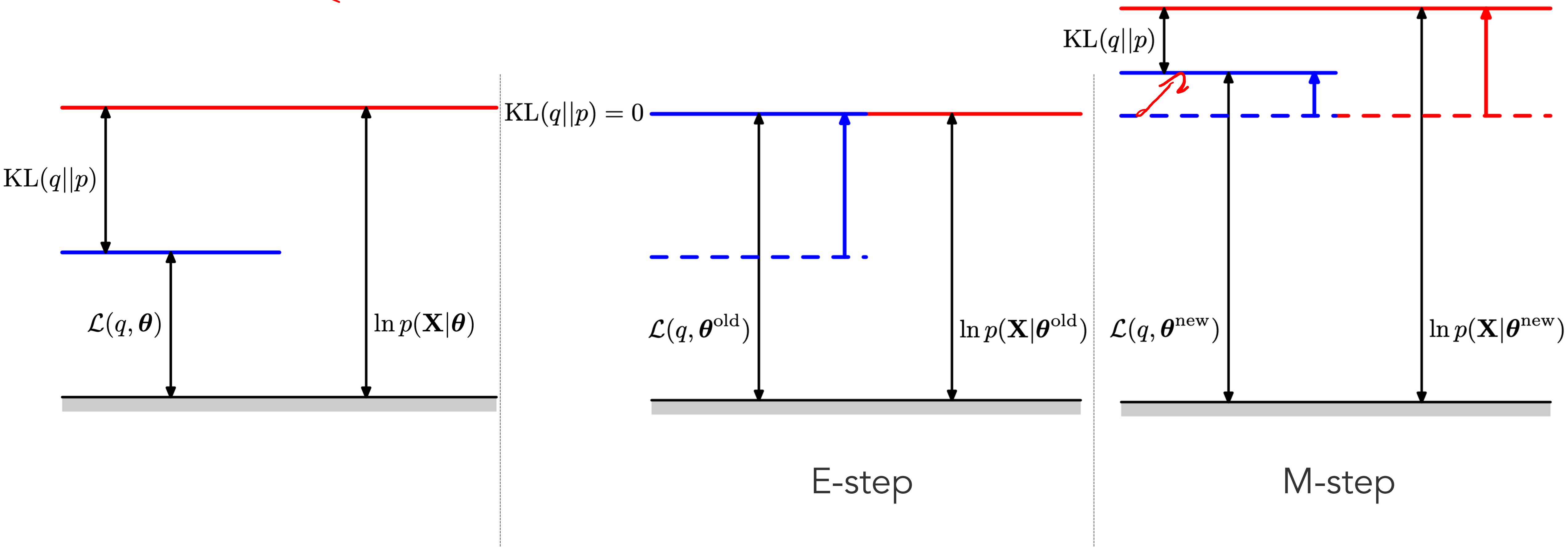
Summary: EM Algorithm

- A way of maximizing likelihood function for latent variable models. Finds MLE of parameters when the original (hard) problem can be broken up into two (easy) pieces
  - Estimate some "missing" or "unobserved" data from observed data and current parameters.
  - Using this "complete" data, find the maximum likelihood parameter estimates.

- Alternate between filling in the latent variables using the best guess (posterior) and updating the parameters based on this guess:
  - E-step: $q^{t+1} = \arg\min_{q} F\left(q, \theta^{t}\right)$
  - M-step: $\theta^{t+1} = \arg\min_{\theta} F\left(q^{t+1}, \theta^{t}\right)$

Each EM iteration guarantees to improve the likelihood

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$



[PRML, Chap 9.4]

EM Application: machine translation

- Lexical translation

[Slides courtesy: Yulia Tsvetkov - CMU; Philipp Koehn - JHU; Chris Dyer - DeepMind]

EM Application: machine translation

- Lexical translation
  - How do we translate a word? Look it up in the dictionary

Haus — house, building, home, household, shell

Look at a parallel corpus (German text along with English translation)

Translation of *Haus*	Count
house	8,000
building	1,600
home	200
household	150
shell	50

EM Application: machine translation

- Lexical translation
  - How do we translate a word? Look it up in the dictionary

Haus — house, building, home, household, shell

Maximum likelihood estimation

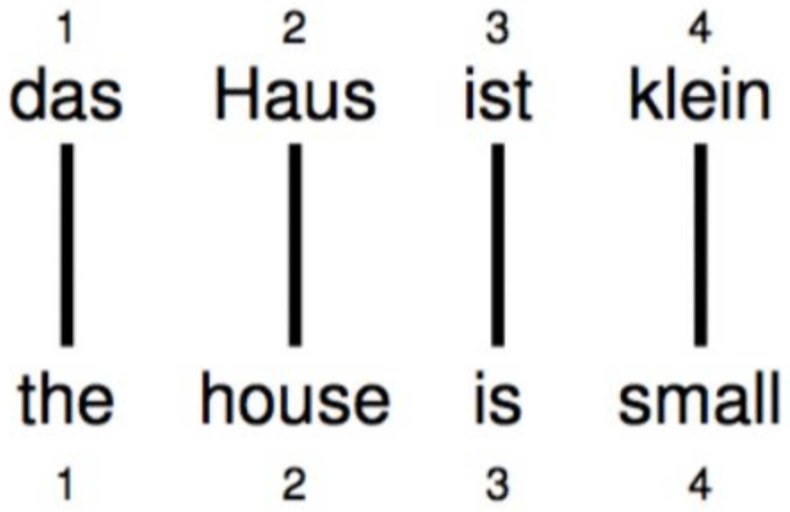
$$\hat{p}_{\text{MLE}}(e \mid \texttt{Haus}) = \begin{cases} 0.8 & \text{if } e = \text{house}, \\ 0.16 & \text{if } e = \text{building}, \\ 0.02 & \text{if } e = \text{home}, \\ 0.015 & \text{if } e = \text{household}, \\ 0.005 & \text{if } e = \text{shell}. \end{cases}$$

Challenge: alignment

- In a parallel text (or when we translate), we align words in one language with the words in the other
- Alignments are represented as vectors of positions:

$$\mathbf{f} = \langle f_1, f_2, \ldots, f_n \rangle$$

$$\mathbf{e} = \langle e_1, e_2, \ldots, e_m \rangle$$



$$\mathbf{a} = (1, 2, 3, 4)$$

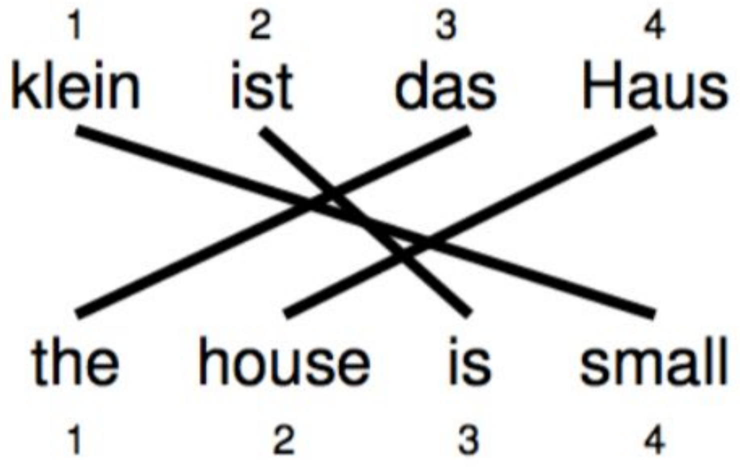
Challenge: alignment

- Formalizing alignment with an alignment function
  - Mapping an English target word at position *i* to a German source word at position *j* with a function $a : i \rightarrow j$
  - Example

$$\mathbf{a} = (1, 2, 3, 4)$$

Challenge: alignment

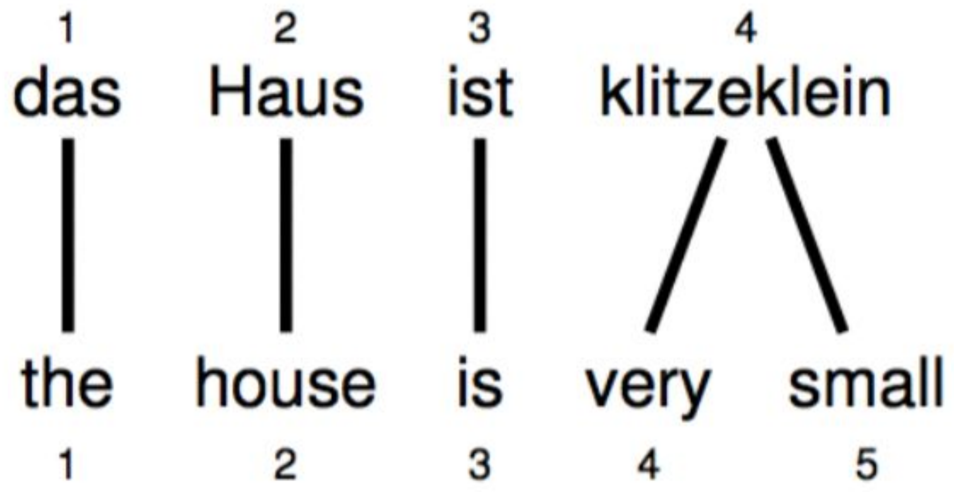
- Reordering
  - Words may be reordered during translation.



$$\mathbf{a} = (3, 4, 2, 1)$$

Challenge: alignment

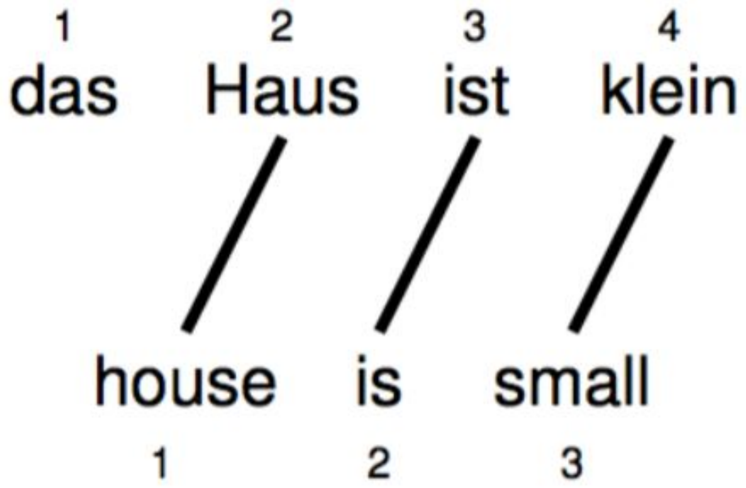
- One-to-many Translation
  - A source word may translate into more than one target word
  -



$$\mathbf{a} = (1, 2, 3, 4, 4)$$

Challenge: alignment

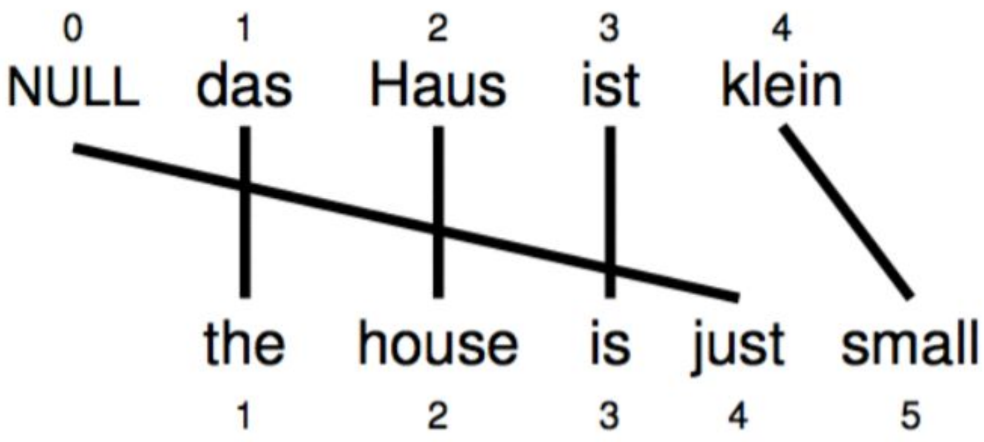
- Word Dropping
  - A source word may not be translated at all



$$\mathbf{a} = (2, 3, 4)$$

Challenge: alignment

- Word Insertion
  - Words may be inserted during translation
    - English *just* does not have an equivalent
    - But it must be explained - we typically assume every source sentence contains a NULL token



$$\mathbf{a} = (1, 2, 3, 0, 4)$$

IBM Model 1

- Simplest lexical translation model

- Translation probability
  - for a foreign sentence **f** = (f_1, ..., f_{l_f}) of length l_f
  - to an English sentence **e** = (e_1, ..., e_{l_e}) of length l_e
  - with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \rightarrow i$

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$

- parameter ϵ is a normalization constant

Example

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
’s	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

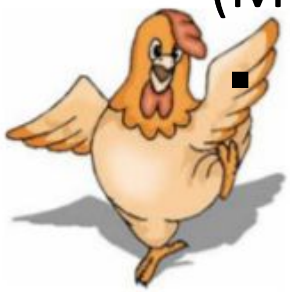
$$
\begin{aligned}
p(e,a|f) &= \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein}) \\
&= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\
&= 0.0028\epsilon
\end{aligned}
$$

Learning Lexical Translation Models

We would like to estimate the lexical translation probabilities *t(e|f)* from a parallel corpus

- ... but we do not have the alignments

- Chicken and egg problem
  - if we had the alignments,
    → we could estimate the parameters of our generative model (MLE)



  - if we had the parameters,
    → we could estimate the alignments

EM algorithm

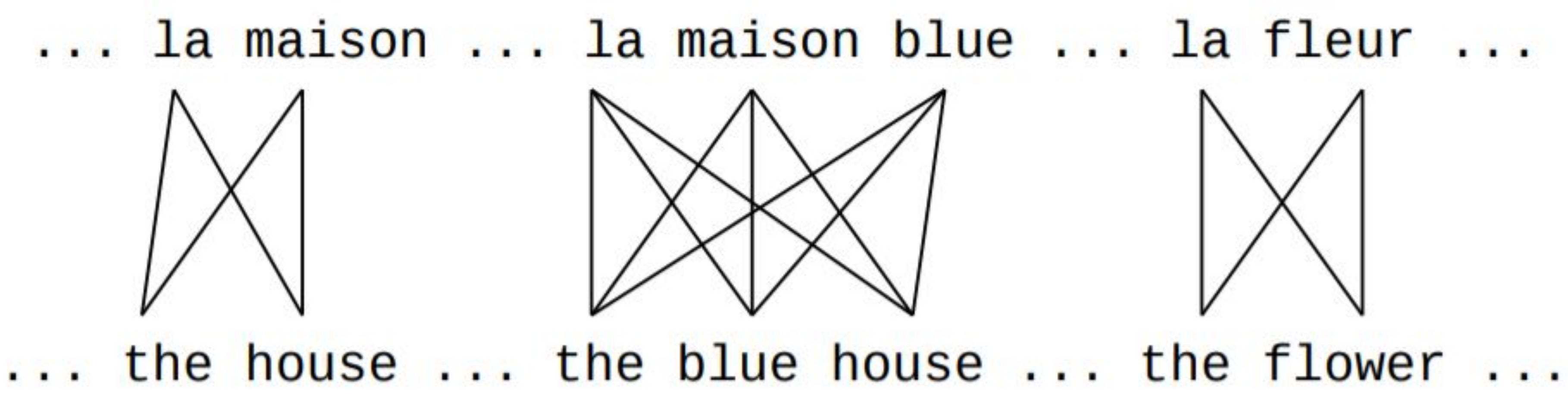
- Observed data: parallel pairs (e, f)
- Missing (latent) data: alignment a

Iterates:

- E-step: use the model to assign probabilities to the missing data
- M-step: estimate model parameters from completed data

EM algorithm

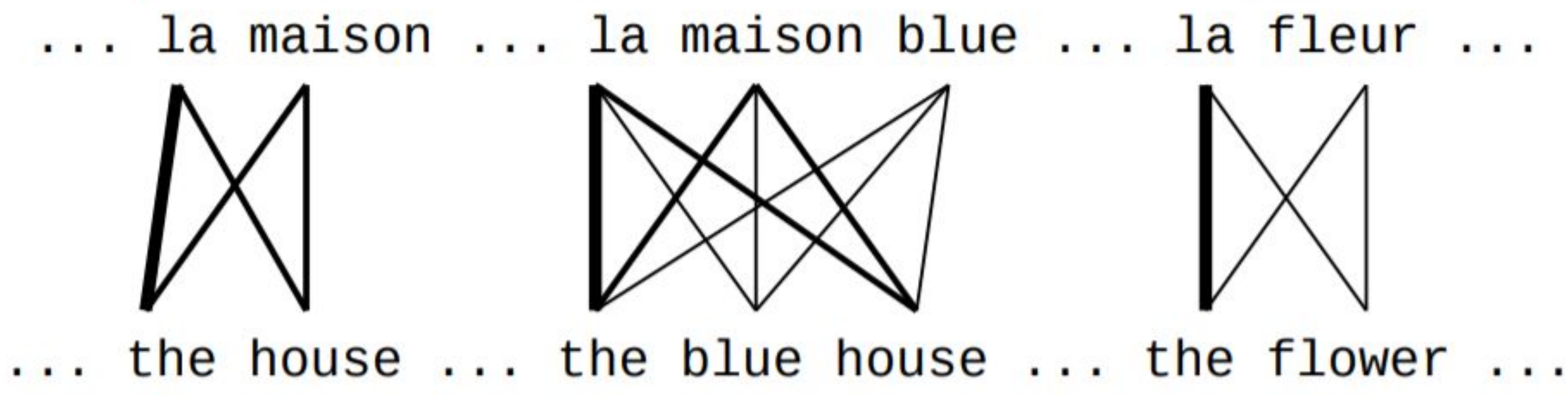
- Observed data: parallel pairs (e, f)
- Missing (latent) data: alignment a



- Initial step: all alignments equally likely
- Model learns that, e.g., *la* is often aligned with *the*

EM algorithm

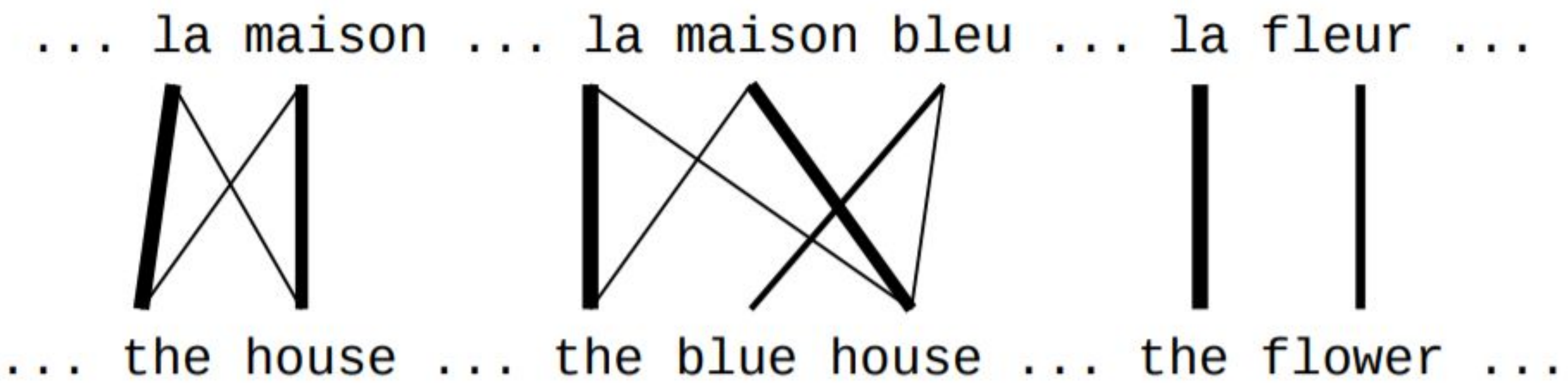
- Observed data: parallel pairs (e, f)
- Missing (latent) data: alignment a



- After one iteration
- Alignments, e.g., between *la* and *the* are more likely

EM algorithm

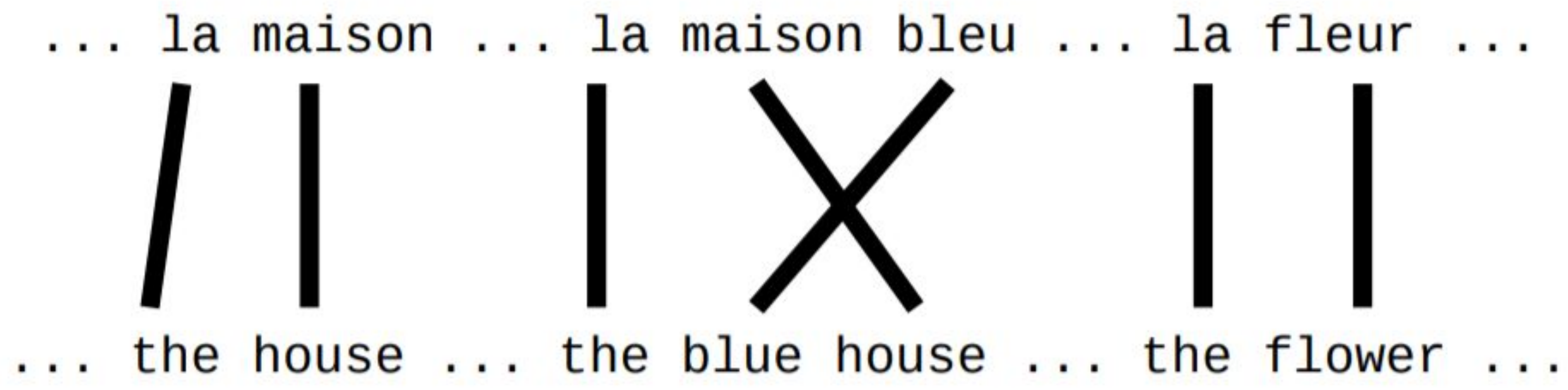
- Observed data: parallel pairs (e, f)
- Missing (latent) data: alignment a



- After another iteration
- It becomes apparent that alignments, e.g., between *fleur* and *flower* are more likely (pigeon hole principle)

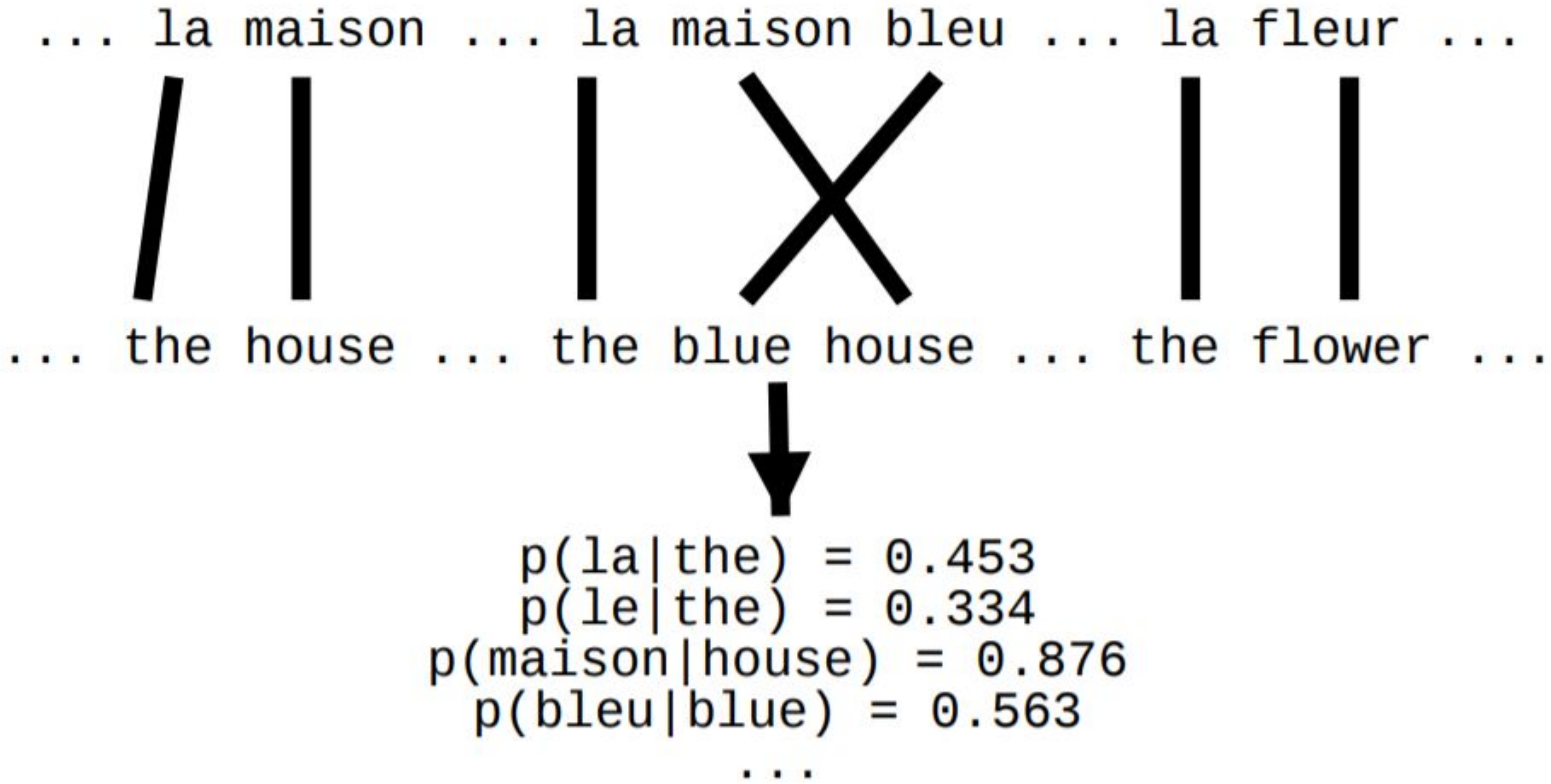
EM algorithm

- Observed data: parallel pairs (e, f)
- Missing (latent) data: alignment a



Convergence
Inherent hidden structure revealed by EM

EM algorithm



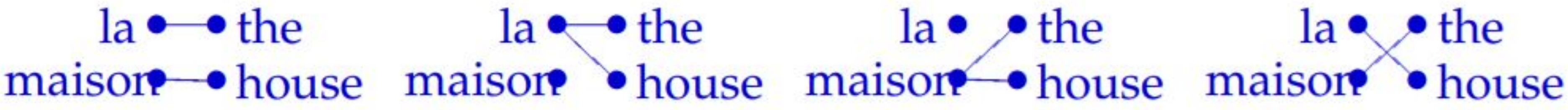
- Parameter estimation from the aligned corpus

IBM Model 1 and EM

t-table **Probabilities**

$$p(\text{the}|\text{la}) = 0.7 \qquad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \qquad p(\text{house}|\text{maison}) = 0.8$$

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

Applying the chain rule:

$$p(a|\mathbf{e}, \mathbf{f}) = \frac{p(\mathbf{e}, a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})}$$

$$p(e, a) = p(e)p(a|e)$$

IBM Model 1 and EM: E-step

We need to compute $p(\mathbf{e}|\mathbf{f})$

$$
\begin{aligned}
p(\mathbf{e}|\mathbf{f}) &= \sum_a p(\mathbf{e}, a|\mathbf{f}) \\
&= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} p(\mathbf{e}, a|\mathbf{f}) \\
&= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})
\end{aligned}
$$

IBM Model 1 and EM: E-step

$$p(\mathbf{e}|\mathbf{f}) = \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$

$$= \frac{\epsilon}{(l_f+1)^{l_e}} \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$

$$= \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)$$

- Note the trick in the last line
  - removes the need for an exponential number of products

→ this makes IBM Model 1 estimation tractable

The trick

(case $l_e = l_f = 2$)

$$\sum_{a(1)=0}^{2} \sum_{a(2)=0}^{2} = \frac{\epsilon}{3^2} \prod_{j=1}^{2} t(e_j|f_{a(j)}) =$$

$$= t(e_1|f_0)\ t(e_2|f_0) + t(e_1|f_0)\ t(e_2|f_1) + t(e_1|f_0)\ t(e_2|f_2) +$$

$$+ t(e_1|f_1)\ t(e_2|f_0) + t(e_1|f_1)\ t(e_2|f_1) + t(e_1|f_1)\ t(e_2|f_2) +$$

$$+ t(e_1|f_2)\ t(e_2|f_0) + t(e_1|f_2)\ t(e_2|f_1) + t(e_1|f_2)\ t(e_2|f_2) =$$

$$= t(e_1|f_0)\,(t(e_2|f_0) + t(e_2|f_1) + t(e_2|f_2)) +$$

$$+ t(e_1|f_1)\,(t(e_2|f_1) + t(e_2|f_1) + t(e_2|f_2)) +$$

$$+ t(e_1|f_2)\,(t(e_2|f_2) + t(e_2|f_1) + t(e_2|f_2)) =$$

$$= (t(e_1|f_0) + t(e_1|f_1) + t(e_1|f_2))\,(t(e_2|f_2) + t(e_2|f_1) + t(e_2|f_2))$$

IBM Model 1 and EM: E-step

Combine what we have:

E-step

$$p(\mathbf{a}|\mathbf{e},\mathbf{f}) = p(\mathbf{e},\mathbf{a}|\mathbf{f})/p(\mathbf{e}|\mathbf{f})$$

$$= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)}$$

$$= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}$$

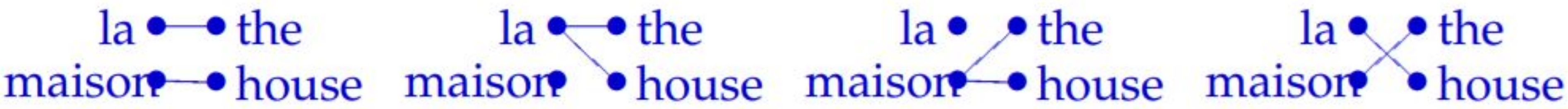
IBM Model 1 and EM: E-step

t-table **Probabilities**

$p(\text{the}|\text{la}) = 0.7$ $\quad$ $p(\text{house}|\text{la}) = 0.05$

$p(\text{the}|\text{maison}) = 0.1$ $\quad$ $p(\text{house}|\text{maison}) = 0.8$

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $\quad$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

E-step $\quad$ $p(a|\mathbf{e}, \mathbf{f}) = 0.824$ $\quad$ $p(a|\mathbf{e}, \mathbf{f}) = 0.052$ $\quad$ $p(a|\mathbf{e}, \mathbf{f}) = 0.118$ $\quad$ $p(a|\mathbf{e}, \mathbf{f}) = 0.007$

$$p(a|\mathbf{e}, \mathbf{f}) = \frac{p(\mathbf{e}, a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})}$$

IBM Model 1 and EM: M-step

Now we have to collect counts

Evidence from a sentence pair **e,f** that word e is a translation of word f:

$$c(e|f;\mathbf{e},\mathbf{f}) = \sum_a p(a|\mathbf{e},\mathbf{f}) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_{a(j)})$$

After collecting these counts over a corpus, we can estimate the model:

$$t(e|f;\mathbf{e},\mathbf{f}) = \frac{\sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f}))}{\sum_e \sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f}))}$$

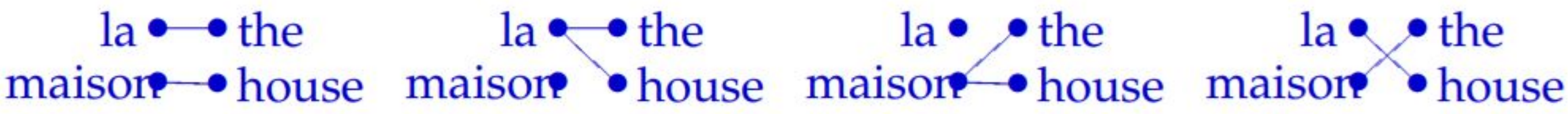
IBM Model 1 and EM: M-step

t-table **Probabilities**

$p(\text{the}|\text{la}) = 0.7$ $\quad p(\text{house}|\text{la}) = 0.05$

$p(\text{the}|\text{maison}) = 0.1$ $\quad p(\text{house}|\text{maison}) = 0.8$

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $\quad p(\mathbf{e}, a|\mathbf{f}) = 0.005$

E-step $p(a|\mathbf{e}, \mathbf{f}) = 0.824$ $\quad p(a|\mathbf{e}, \mathbf{f}) = 0.052$ $\quad p(a|\mathbf{e}, \mathbf{f}) = 0.118$ $\quad p(a|\mathbf{e}, \mathbf{f}) = 0.007$

M-step **Counts**

$c(\text{the}|\text{la}) = 0.824 + 0.052$ $\quad c(\text{house}|\text{la}) = 0.052 + 0.007$

$c(\text{the}|\text{maison}) = 0.118 + 0.007$ $\quad c(\text{house}|\text{maison}) = 0.824 + 0.118$

IBM Model 1 and EM: M-step

t-table **Probabilities**

$$p(\text{the}|\text{la}) = 0.7 \quad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \quad p(\text{house}|\text{maison}) = 0.8$$

E-step **Alignments**

$$p(a|\mathbf{e},\mathbf{f}) = 0.824 \quad p(a|\mathbf{e},\mathbf{f}) = 0.052 \quad p(a|\mathbf{e},\mathbf{f}) = 0.118 \quad p(a|\mathbf{e},\mathbf{f}) = 0.007$$

M-step **Counts**

$$c(\text{the}|\text{la}) = 0.824 + 0.052 \quad c(\text{house}|\text{la}) = 0.052 + 0.007$$
$$c(\text{the}|\text{maison}) = 0.118 + 0.007 \quad c(\text{house}|\text{maison}) = 0.824 + 0.118$$

Update t-table:

$$p(\text{the}|\text{la}) = c(\text{the}|\text{la})/c(\text{la})$$

Higher IBM Models

IBM Model 1	lexical translation
IBM Model 2	adds absolute reordering model
IBM Model 3	adds fertility model
IBM Model 4	relative reordering model
IBM Model 5	fixes deficiency

Key Takeaways

- Unsupervised learning
  - Maximum likelihood estimation (MLE) with latent variables
  - EM algorithm for MLE
    - Expected complete log likelihood
    - Evidence lower bound (ELBO)
    - Coordinate ascent: E-step, M-step

- Use case: EM for MT alignment

EM Variants

- Sparse EM
  - Do not re-compute exactly the posterior probability on each data point under all models, because it is almost zero.
  - Instead keep an "active list" which you update every once in a while.

- Generalized (Incomplete) EM:
  - It might be hard to find the ML parameters in the M-step, even given the completed data. We can still make progress by doing an M-step that improves the likelihood a bit (e.g. gradient step).

Questions?