Learning for Hidden Markov Models & Course Recap

Michael Gutmann

Probabilistic Modelling and Reasoning (INFR11134)
School of Informatics, University of Edinburgh

Spring semester 2018

Recap

- We can decompose the log marginal of any joint distribution into a sum of two terms:
  - the free energy and
  - the KL divergence between the variational and the conditional distribution
- Variational principle: Maximising the free energy with respect to the variational distribution allows us to (approximately) compute the (log) marginal and the conditional from the joint.
- We applied the variational principle to inference and learning problems.
- For parameter estimation in presence of unobserved variables: Coordinate ascent on the free energy leads to the (variational) EM algorithm.

Program

1. EM algorithm to learn the parameters of HMMs

2. Course recap

Program

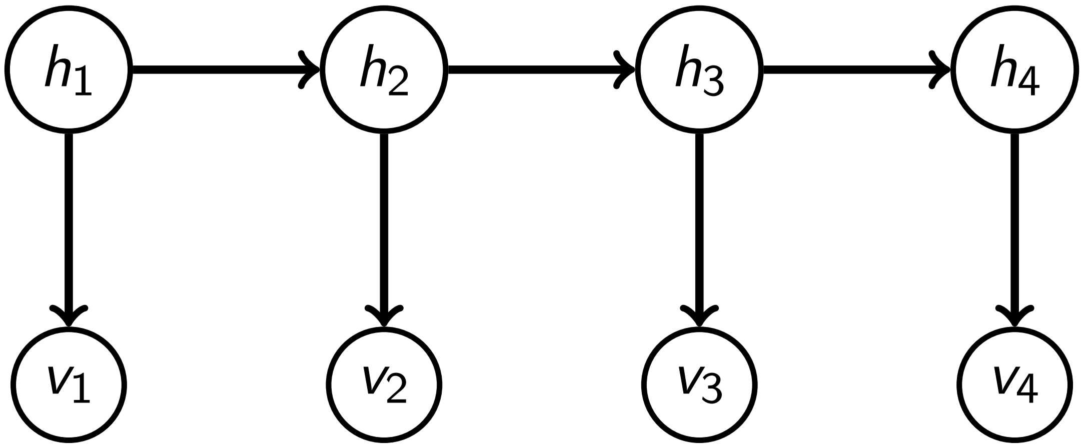
1. EM algorithm to learn the parameters of HMMs
   - Problem statement
   - Learning by gradient ascent on the log-likelihood or by EM
   - EM update equations

2. Course recap

Hidden Markov model

Specified by

- DAG (representing the independence assumptions)



- Transition distribution $p(h_i|h_{i-1})$
- Emission distribution $p(v_i|h_i)$
- Initial state distribution $p(h_1)$

The classical inference problems

- Classical inference problems:
  - Filtering: $p(h_t|v_{1:t})$
  - Smoothing: $p(h_t|v_{1:u})$ where $t < u$
  - Prediction: $p(h_t|v_{1:u})$ and/or $p(v_t|v_{1:u})$ where $t > u$
  - Most likely hidden path (Viterbi alignment): $\operatorname{argmax}_{h_{1:t}} p(h_{1:t}|v_{1:t})$
- Inference problems can be solved by message passing.
- Requires that the transition, emission, and initial state distributions are known.

Learning problem

- Data: $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, where each $\mathcal{D}_j$ is a sequence of visibles of length d, i.e.

$$\mathcal{D}_j = (v_1^{(j)}, \ldots, v_d^{(j)})$$

- Assumptions:
  - All variables are discrete: $h_i \in \{1, \ldots K\}$, $v_i \in \{1, \ldots, M\}$.
  - Stationarity
- Parametrisation:
  - Transition distribution is parametrised by the matrix $\mathbf{A}$

$$p(h_i = k | h_{i-1} = k'; \mathbf{A}) = A_{k,k'}$$

  - Emission distribution is parametrised by the matrix $\mathbf{B}$

$$p(v_i = m | h_i = k; \mathbf{B}) = B_{m,k}$$

  - Initial state distribution is parametrised by the vector $\mathbf{a}$

$$p(h_1 = k; \mathbf{a}) = a_k$$

- Task: Use the data $\mathcal{D}$ to learn $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{a}$

Learning problem

- Since **A**, **B**, and **a** represent (conditional) distributions, the parameters are constrained to be non-negative and to satisfy

$$\sum_{k=1}^{K} p(h_i = k | h_{i-1} = k') = \sum_{k=1}^{K} A_{k,k'} = 1$$

$$\sum_{m=1}^{M} p(v_i = m | h_i = k) = \sum_{m=1}^{M} B_{m,k} = 1$$

$$\sum_{k=1}^{k} p(h_1 = k) = \sum_{k=1}^{K} a_k = 1$$

- Note: Much of what follows holds more generally for HMMs and does not use the stationarity assumption or that the h_i and v_i are discrete random variables.
- The parameters together will be denoted by $\boldsymbol{\theta}$.

Options for learning the parameters

- The model $p(\mathbf{h}, \mathbf{v}; \boldsymbol{\theta})$ is normalised but we have unobserved variables.
- Option 1: Simple gradient ascent on the log-likelihood

$$\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{old}} + \epsilon \sum_{j=1}^{n} \mathbb{E}_{p(\mathbf{h}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\nabla_{\boldsymbol{\theta}} \log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}_{\text{old}}} \right]$$

  see slides *Intractable Likelihood Functions*
- Option 2: EM algorithm

$$\boldsymbol{\theta}_{\text{new}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{j=1}^{n} \mathbb{E}_{p(\mathbf{h}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} [\log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta})]$$

  see slides *Variational Inference and Learning*
- For HMMs, both are possible thanks to sum-product message passing.

Options for learning the parameters

Option 1: $\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{old}} + \epsilon \sum_{j=1}^{n} \mathbb{E}_{p(\mathbf{h}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\nabla_{\boldsymbol{\theta}} \log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}_{\text{old}}} \right]$

Option 2: $\boldsymbol{\theta}_{\text{new}} = \text{argmax}_{\boldsymbol{\theta}} \sum_{j=1}^{n} \mathbb{E}_{p(\mathbf{h}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta}) \right]$

- Similarities:
  - Both require computation of the posterior expectation.
  - Assume the "M" step is performed by gradient ascent,

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \epsilon \sum_{j=1}^{n} \mathbb{E}_{p(\mathbf{h}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\nabla_{\boldsymbol{\theta}} \log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}} \right]$$

    where $\boldsymbol{\theta}$ is initialised with $\boldsymbol{\theta}_{\text{old}}$, and the final $\boldsymbol{\theta}'$ gives $\boldsymbol{\theta}_{\text{new}}$.
    If only one gradient step is taken, option 2 becomes option 1.
- Differences:
  - Unlike option 2, option 1 requires re-computation of the posterior after each ϵ update of $\boldsymbol{\theta}$, which may be costly.
  - In some cases (including HMMs), the "M"/argmax step can be performed analytically in closed form.

Expected complete data log-likelihood

- Denote the objective in the EM algorithm by $J(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}})$,

$$J(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) = \sum_{j=1}^{n} \mathbb{E}_{p(\mathbf{h}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta})\right]$$

- We show on the next slide that in general for the HMM model, the full posteriors $p(\mathbf{h}|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})$ are not needed but just

$$p(h_i|h_{i-1}, \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \qquad p(h_i|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}).$$

  They can be obtained by the alpha-beta recursion (sum-product algorithm).

- Posteriors need to be computed for each observed sequence $\mathcal{D}_j$, and need to be re-computed after updating $\boldsymbol{\theta}$.

Expected complete data log-likelihood

- The HMM model factorises as

$$p(\mathbf{h}, \mathbf{v}; \boldsymbol{\theta}) = p(h_1; \mathbf{a}) p(v_1 | h_1; \mathbf{B}) \prod_{i=2}^{d} p(h_i | h_{i-1}; \mathbf{A}) p(v_i | h_i; \mathbf{B})$$

- For sequence $\mathcal{D}_j$, we have

$$\log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta}) = \log p(h_1; \mathbf{a}) + \log p(v_1^{(j)} | h_1; \mathbf{B}) + \sum_{i=2}^{d} \log p(h_i | h_{i-1}; \mathbf{A}) + \log p(v_i^{(j)} | h_i; \mathbf{B})$$

- Since

$$\mathbb{E}_{p(\mathbf{h} | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} [\log p(h_1; \mathbf{a})] = \mathbb{E}_{p(h_1 | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} [\log p(h_1; \mathbf{a})]$$

$$\mathbb{E}_{p(\mathbf{h} | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} [\log p(h_i | h_{i-1}; \mathbf{A})] = \mathbb{E}_{p(h_i, h_{i-1} | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} [\log p(h_i | h_{i-1}; \mathbf{A})]$$

$$\mathbb{E}_{p(\mathbf{h} | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} \left[\log p(v_i^{(j)} | h_i; \mathbf{B})\right] = \mathbb{E}_{p(h_i | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} \left[\log p(v_i^{(j)} | h_i; \mathbf{B})\right]$$

we do not need the full posterior but only the marginal posteriors and the joint of the neighbouring variables.

Expected complete data log-likelihood

With the factorisation (independencies) in the HMM model, the objective function thus becomes

$$
\begin{aligned}
J(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) &= \sum_{j=1}^{n} \mathbb{E}_{p(\mathbf{h}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\log p(\mathbf{h}, \mathcal{D}_j; \boldsymbol{\theta})\right] \\
&= \sum_{j=1}^{n} \mathbb{E}_{p(h_1|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\log p(h_1; \mathbf{a})\right] + \\
& \quad \sum_{j=1}^{n} \sum_{i=2}^{d} \mathbb{E}_{p(h_i, h_{i-1}|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\log p(h_i | h_{i-1}; \mathbf{A})\right] + \\
& \quad \sum_{j=1}^{n} \sum_{i=1}^{d} \mathbb{E}_{p(h_i|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})} \left[\log p(v_i^{(j)} | h_i; \mathbf{B})\right]
\end{aligned}
$$

In the derivation so far we have not yet used the assumed parametrisation of the model. We insert these assumptions next.

The term for the initial state distribution

- We have assumed that

$$p(h_1 = k; \mathbf{a}) = a_k \qquad k = 1, \ldots, K$$

which we can write as

$$p(h_1; \mathbf{a}) = \prod_k a_k^{\mathbb{1}(h_1 = k)}$$

(like for the Bernoulli model, see slides *Basics of Model-Based Learning* and Tutorial 7)

- The log pmf is thus

$$\log p(h_1; \mathbf{a}) = \sum_k \mathbb{1}(h_1 = k) \log a_k$$

- Hence

$$\begin{aligned}\mathbb{E}_{p(h_1|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})}\left[\log p(h_1; \mathbf{a})\right] &= \sum_k \mathbb{E}_{p(h_1|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})}\left[\mathbb{1}(h_1 = k)\right] \log a_k \\ &= \sum_k p(h_1 = k|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \log a_k\end{aligned}$$

The term for the transition distribution

- We have assumed that

$$p(h_i = k | h_{i-1} = k'; \mathbf{A}) = A_{k,k'} \qquad k, k' = 1, \dots K$$

which we can write as

$$p(h_i | h_{i-1}; \mathbf{A}) = \prod_{k,k'} A_{k,k'}^{\mathbb{1}(h_i = k, h_{i-1} = k')}$$

(see slides *Basics of Model-Based Learning* and Tutorial 7)

- Further:

$$\log p(h_i | h_{i-1}; \mathbf{A}) = \sum_{k,k'} \mathbb{1}(h_i = k, h_{i-1} = k') \log A_{k,k'}$$

- Hence $\mathbb{E}_{p(h_i, h_{i-1} | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} [\log p(h_i | h_{i-1}; \mathbf{A})]$ equals

$$\sum_{k,k'} \mathbb{E}_{p(h_i, h_{i-1} | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})} \left[\mathbb{1}(h_i = k, h_{i-1} = k') \right] \log A_{k,k'}$$

$$= \sum_{k,k'} p(h_i = k, h_{i-1} = k' | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \log A_{k,k'}$$

The term for the emission distribution

We can do the same for the emission distribution.

With

$$p(v_i|h_i;\mathbf{B}) = \prod_{m,k} B_{m,k}^{\mathbb{1}(v_i=m,h_i=k)} = \prod_{m,k} B_{m,k}^{\mathbb{1}(v_i=m)\mathbb{1}(h_i=k)}$$

we have

$$\mathbb{E}_{p(h_i|\mathcal{D}_j;\boldsymbol{\theta}_{\text{old}})}\left[\log p(v_i^{(j)}|h_i;\mathbf{B})\right] = \sum_{m,k} \mathbb{1}(v_i^{(j)} = m)p(h_i = k|\mathcal{D}_j, \boldsymbol{\theta}_{\text{old}}) \log B_{m,k}$$

E-step for discrete-valued HMM

- Putting all together, we obtain the complete data log likelihood for the HMM with discrete visibles and hiddens.

$$
\begin{aligned}
J(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) = & \sum_{j=1}^{n} \sum_{k} p(h_1 = k | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \log a_k + \\
& \sum_{j=1}^{n} \sum_{i=2}^{d} \sum_{k,k'} p(h_i = k, h_{i-1} = k' | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \log A_{k,k'} + \\
& \sum_{j=1}^{n} \sum_{i=1}^{d} \sum_{m,k} \mathbb{1}(v_i^{(j)} = m) p(h_i = k | \mathcal{D}_j, \boldsymbol{\theta}_{\text{old}}) \log B_{m,k}
\end{aligned}
$$

- The objectives for **a**, and the columns of **A** and **B** decouple.
- Does not completely decouple because of the constraint that the elements of **a** have to sum to one, and that the columns of **A** and **B** have to sum to one.

M-step

- We discuss the details for the maximisation with respect to **a**. The other cases are done equivalently.
- Optimisation problem:

$$\max_{\mathbf{a}} \sum_{j=1}^{n} \sum_{k} p(h_1 = k | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \log a_k$$

$$\text{subject to } a_k \geq 0 \quad \sum_{k} a_k = 1$$

- The non-negativity constraint could be handled by re-parametrisation, but the constraint is here not active (the objective is not defined for $a_k \leq 0$) and can be dropped.
- The normalisation constraint can be handled by using the methods of Lagrange multipliers (see e.g. Barber Appendix A.6).

M-step

- Lagrangian: $\sum_{j=1}^{n} \sum_k p(h_1 = k | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \log a_k - \lambda (\sum_k a_k - 1)$
- The derivative with respect to a specific a_i is

$$\sum_{j=1}^{n} p(h_1 = i | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \frac{1}{a_i} - \lambda$$

- Gives the necessary condition for optimality

$$a_i = \frac{1}{\lambda} \sum_{j=1}^{n} p(h_1 = i | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})$$

- The derivative with respect to λ gives back the constraint

$$\sum_i a_i = 1$$

- Set $\lambda = \sum_i \sum_{j=1}^{n} p(h_1 = i | \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})$ to satisfy the constraint.
- The Hessian of the Lagrangian is negative definite, which shows that we have found a maximum.

M-step

- Since $\sum_i p(h_1 = i|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) = 1$, we obtain $\lambda = n$ so that

$$a_k = \frac{1}{n} \sum_{j=1}^{n} p(h_1 = k|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})$$

Average of all posteriors of h_1 obtained by message passing.

- Equivalent calculations give

$$A_{k,k'} = \frac{\sum_{j=1}^{n} \sum_{i=2}^{d} p(h_i = k, h_{i-1} = k'|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}{\sum_k \sum_{j=1}^{n} \sum_{i=2}^{d} p(h_i = k, h_{i-1} = k'|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}$$

and

$$B_{m,k} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{d} \mathbb{1}(v_i^{(j)} = m) p(h_i = k|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}{\sum_m \sum_{j=1}^{n} \sum_{i=1}^{d} \mathbb{1}(v_i^{(j)} = m) p(h_i = k|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}$$

Inferred posteriors obtained by message passing are averaged over different sequences $\mathcal{D}_j$ and across each sequence (stationarity).

EM for discrete-valued HMM (Baum-Welch algorithm)

Given parameters $\boldsymbol{\theta}_{\text{old}}$

1. For each sequence $\mathcal{D}_j$ compute the posteriors

$$p(h_i|h_{i-1}, \mathcal{D}_j; \boldsymbol{\theta}_{\text{old}}) \qquad p(h_i|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})$$

   using the alpha-beta recursion (sum-product algorithm)

2. Update the parameters

$$a_k = \frac{1}{n}\sum_{j=1}^{n} p(h_1 = k|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})$$

$$A_{k,k'} = \frac{\sum_{j=1}^{n}\sum_{i=2}^{d} p(h_i = k, h_{i-1} = k'|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}{\sum_k \sum_{j=1}^{n}\sum_{i=2}^{d} p(h_i = k, h_{i-1} = k'|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}$$

$$B_{m,k} = \frac{\sum_{j=1}^{n}\sum_{i=1}^{d} \mathbb{1}(v_i^{(j)} = m)p(h_i = k|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}{\sum_m \sum_{j=1}^{n}\sum_{i=1}^{d} \mathbb{1}(v_i^{(j)} = m)p(h_i = k|\mathcal{D}_j; \boldsymbol{\theta}_{\text{old}})}$$

Repeat step 1 and 2 using the new parameters for $\boldsymbol{\theta}_{\text{old}}$. Stop e.g. if change in parameters is less than a threshold.

Program

1. EM algorithm to learn the parameters of HMMs
    - Problem statement
    - Learning by gradient ascent on the log-likelihood or by EM
    - EM update equations

2. Course recap

Program

1. EM algorithm to learn the parameters of HMMs

2. Course recap

Course recap

- We started the course with the basic observation that variability is part of nature.
- Variability leads to uncertainty when analysing or drawing conclusions from data.
- This motivates taking a probabilistic approach to modelling and reasoning.

Course recap

- Probabilistic modelling:
  - Identify the quantities that relate to the aspects of reality that you wish to capture with your model.
  - Consider them to be random variables, e.g. $\mathbf{x}, \mathbf{y}, \mathbf{z}$, with a joint pdf (pmf) $p(\mathbf{x}, \mathbf{y}, \mathbf{z})$.
- Probabilistic reasoning:
  - Assume you know that $\mathbf{y} \in \mathcal{E}$ (measurement, evidence)
  - Probabilistic reasoning about $\mathbf{x}$ then consists in computing

    $$p(\mathbf{x}|\mathbf{y} \in \mathcal{E})$$

    or related quantities like its maximiser or posterior expectations.

Course recap

- Principled framework but naive implementation quickly runs into computational issues.
- For example,

$$p(\mathbf{x}|\mathbf{y}_o) = \frac{\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{y}_o, \mathbf{z})}{\sum_{\mathbf{x},\mathbf{z}} p(\mathbf{x}, \mathbf{y}_o, \mathbf{z})}$$

  cannot be computed if $\mathbf{x}, \mathbf{y}, \mathbf{z}$ each are $d = 500$ dimensional, and if each element of the vectors can take $K = 10$ values.
- The course had four main topics.

  Topic 1: Representation We discussed reasonable weak assumptions to efficiently represent $p(\mathbf{x}, \mathbf{y}, \mathbf{z})$.
  - Two classes of assumptions: independence and parametric assumptions.
  - Directed and undirected graphical models
  - Expressive power of the graphical models
  - Factor graphs

Course recap

Topic 2: Exact inference We have seen that the independence assumptions allow us, under certain conditions, to efficiently compute the posterior probability or derived quantities.

- Variable elimination for general factor graphs
- Inference when the model can be represented as a factor tree (message passing algorithms)
- Application to Hidden Markov models

Topic 3: Learning We discussed methods to learn probabilistic models from data by introducing parameters and learning them from data.

- Learning by Bayesian inference
- Learning by parameter estimation
- Likelihood function
- Factor analysis and independent component analysis

Course recap

Topic 4: Approximate inference and learning We discussed that intractable integrals may hinder inference and likelihood-based learning.

- Intractable integrals may be due to unobserved variables or intractable partition functions.
- Alternative criteria for learning when the partition function is intractable (score matching)
- Monte Carlo integration and sampling
- Variational approaches to learning and inference
- EM algorithm and its application to hidden Markov models