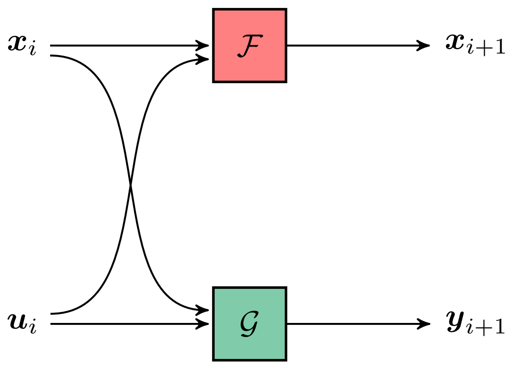
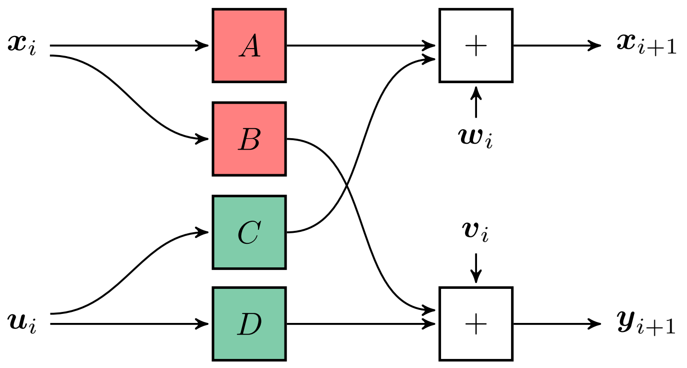
MTAT.03.227 Machine Learning

Expectation-Maximisation algorithm for sequential models

Sven Laur
University of Tartu

Discrete time systems



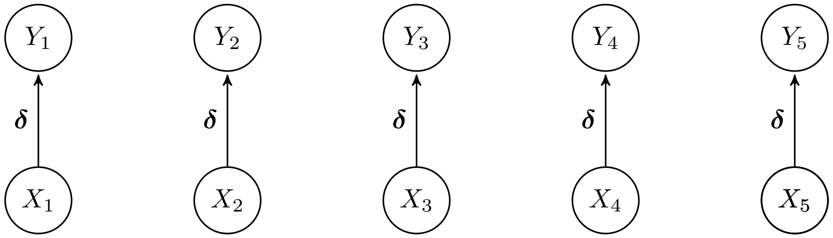


Sequential models describe evolution of discrete time systems.

- System has an hidden state $\boldsymbol{x}_i$ evolving over state space $\mathcal{X}$.
- We can make observations of the system $\boldsymbol{y}_i$ by measuring it.
- We can influence the system by changing the control signal $\boldsymbol{u}_i$.
- For linear system, uncontrollable noise $\boldsymbol{w}_i$ perturbs the state $\boldsymbol{x}_{i+1}$.
- For linear system, uncontrollable noise $\boldsymbol{v}_i$ perturbs the observation $\boldsymbol{y}_i$.

Enforcing temporal consistency

Multinomial mixture model



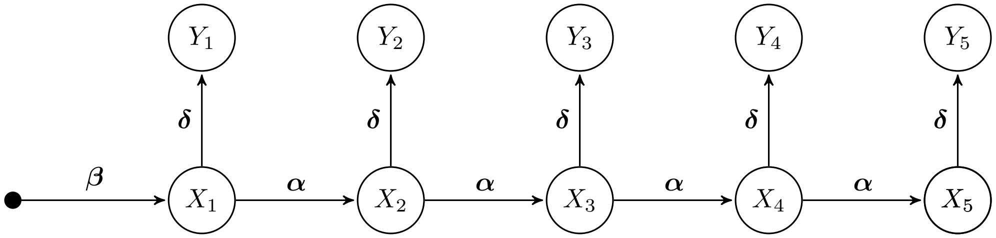
Multinomial mixture model is a discrete time-system.

- The state space $\mathcal{X}$ is finite.
- All states $x_1, \ldots, x_n$ are independently and identically distributed.
- Mixture proportions $(\lambda_x)_{x\in\mathcal{X}}$ quantify the corresponding probabilities.
- Emission matrix $(\delta_{xy})_{x\in\mathcal{X},y\in\mathcal{Y}}$ the conditional probability of outcomes.

$$\lambda_x = \Pr\left[x_i = x\right]$$

$$\delta_{xy} = \Pr\left[y_i = y | x_i = x\right]$$

Discrete Hidden Markov Model



Discrete HMM is a refinement of multinomial mixture model.

▷ State transition probabilities $(\alpha_{x,x'})_{x,x'\in\mathcal{X}}$ are non-trivial.

▷ Initial state probabilities $(\beta_x)_{x\in\mathcal{X}}$ become important now.

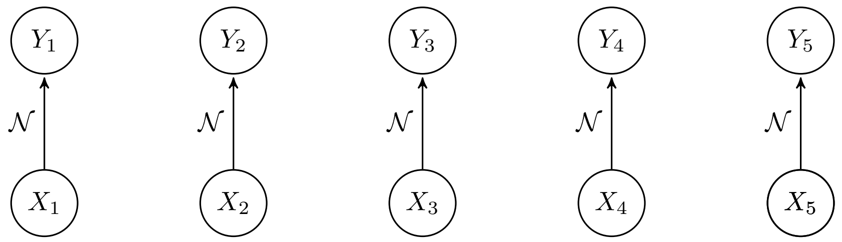
▷ Marginal state probabilities $(\lambda_{xi})_{x\in\mathcal{X},i\in\mathbb{N}}$ change over time.

$$\beta_x = \Pr\left[x_1 = x\right]$$

$$\alpha_{xx'} = \Pr\left[x_{i+1} = x' | x_i = x\right]$$

$$\delta_{xy} = \Pr\left[y_i = y | x_i = x\right]$$

Gaussian mixture model



Gaussian mixture model is a discrete time-system.

▷ The state space $\mathcal{X}$ is finite.

▷ All states $x_1, \ldots, x_n$ are independently and identically distributed.

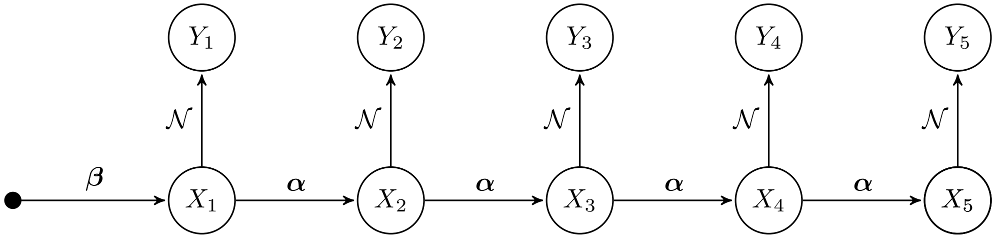
▷ Mixture proportions $(\lambda_x)_{x \in \mathcal{X}}$ quantify the corresponding probabilities.

▷ Observations $\boldsymbol{y}_i$ are determined by multivariate normal distributions.

$$\lambda_x = \Pr\left[x_i = x\right]$$

$$\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{\mu}_{x_i}, \boldsymbol{\Sigma}_{x_i})$$

Continuous Hidden Markov Model



Continuous HMM is a refinement of Gaussian mixture model.

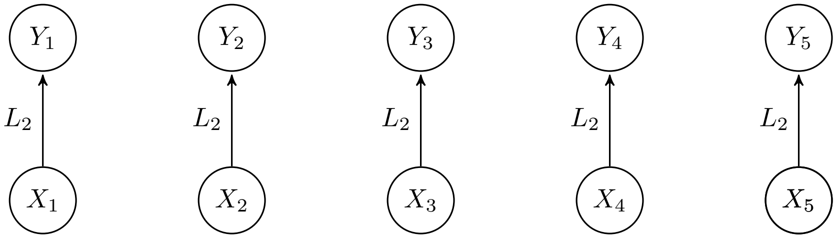
- ▷ State transition probabilities $(\alpha_{x,x'})_{x,x'\in\mathcal{X}}$ are non-trivial.
- ▷ Initial state probabilities $(\beta_x)_{x\in\mathcal{X}}$ become important now.
- ▷ Marginal state probabilities $(\lambda_{xi})_{x\in\mathcal{X},i\in\mathbb{N}}$ change over time.

$$\beta_x = \Pr\left[x_1 = x\right]$$

$$\alpha_{xx'} = \Pr\left[x_{i+1} = x' | x_i = x\right]$$

$$\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{\mu}_{x_i}, \boldsymbol{\Sigma}_{x_i})$$

Multivariate linear transformations



Multivariate linear transformation is a discrete time-system.

▷ The merged input and state space $\mathbb{R}^d \times \mathbb{R}^k$ is infinite.

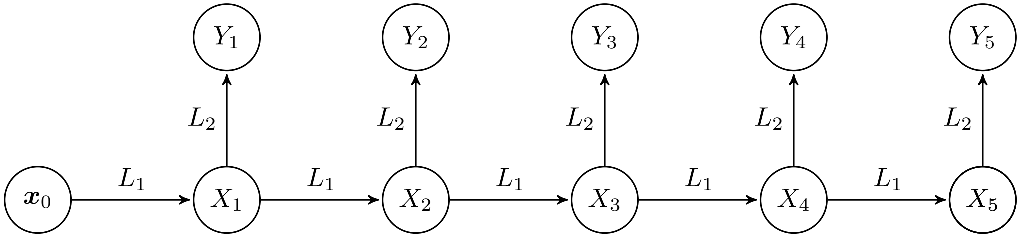
▷ All states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are independently and identically distributed.

▷ States and observations disturbed by white gaussian noise.

▷ Observations $\boldsymbol{y}_i$ are linear in inputs $\boldsymbol{u}_i$ and states $\boldsymbol{x}_i$.

$$\begin{aligned} \boldsymbol{x}_{i+1} &= \mathbf{0}\boldsymbol{x}_i + \mathbf{0}\boldsymbol{u}_i + \boldsymbol{w}_i, & \boldsymbol{w}_i &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \\ \boldsymbol{y}_i &= C\boldsymbol{x}_i + D\boldsymbol{u}_i + \boldsymbol{v}_i, & \boldsymbol{v}_i &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \end{aligned}$$

Kalman filter



Kalman filter is a refinement of multivariate linear transformation.

▷ Linear state evolution equation is non-trivial.

▷ The initial state $\boldsymbol{x}_0$ is assumed to be fixed value.

▷ States and observations disturbed by colored gaussian noise.

$$\begin{aligned}\boldsymbol{x}_{i+1} &= A\boldsymbol{x}_i + B\boldsymbol{u}_i + \boldsymbol{w}_i, &\qquad \boldsymbol{w}_i &\sim \mathcal{N}(\mathbf{0}, \Sigma_1)\\ \boldsymbol{y}_i &= C\boldsymbol{x}_i + D\boldsymbol{u}_i + \boldsymbol{v}_i, &\qquad \boldsymbol{v}_i &\sim \mathcal{N}(\mathbf{0}, \Sigma_2)\end{aligned}$$

EM-algorithm for HMM

Lower bound function

The lower bound function used in the EM algorithm is current notation

$$F(\boldsymbol{q},\boldsymbol{\Theta})=-\sum_{\boldsymbol{x}} q(\boldsymbol{x})\cdot\log q(\boldsymbol{x})+\sum_{\boldsymbol{x}} q(\boldsymbol{x})\cdot\log\left(p\left[\boldsymbol{\Theta},\boldsymbol{x}|\boldsymbol{y}\right]\right)$$

If we assign non-informative prior to the model parameters then in M-step it is sufficient to maximise the function

$$F_*=\sum_{\boldsymbol{x}} q(\boldsymbol{x})\cdot\log\left(p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}\right]\right)$$

Probability assignment in E-step

According to the theory the optimal probability assignment is

$$q(\boldsymbol{x}) = \Pr\left[\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\Theta}_*\right] = \frac{p\left[\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Theta}_*\right]}{p\left[\boldsymbol{y}|\boldsymbol{\Theta}_*\right]}$$

and thus we get

$$F_* = \frac{1}{p\left[\boldsymbol{y}|\boldsymbol{\Theta}_*\right]} \cdot \sum_{\boldsymbol{x}} p\left[\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Theta}_*\right] \cdot \log\left(p\left[\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Theta}\right]\right)$$

and thus it is sufficient to maximise

$$Q = \sum_{\boldsymbol{x}} p\left[\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Theta}_*\right] \cdot \log\left(p\left[\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\Theta}\right]\right)$$

Further decomposition

As the log-likelihood decomposes into three independent parameter groups

$$\log\left(p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}\right]\right)=\log\beta_{x_1}+\sum_{i=2}^{n}\log(\alpha_{x_{i-1}x_i})+\sum_{i=1}^{n}\log(p\left[y_i|x_i\right])$$

we can solve three independent maximisation tasks in the M-step:

$$Q_1=\sum_{\boldsymbol{x}}p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}_*\right]\cdot\log\beta_{x_1}$$

$$Q_2=\sum_{\boldsymbol{x}}p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}_*\right]\cdot\sum_{i=2}^{n}\log(\alpha_{x_{i-1}x_i})$$

$$Q_3=\sum_{\boldsymbol{x}}p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}_*\right]\cdot\sum_{i=1}^{n}\log(p\left[y_i|x_i\right])$$

Simplification of the first term

As

$$
\begin{aligned}
Q_1 &= \sum_{\boldsymbol{x}} p\left[y_1, x_1 | \boldsymbol{\Theta}_*\right] p\left[y_2 \ldots, y_n, x_2, \ldots, x_2 | x_1, \boldsymbol{\Theta}_*\right] \cdot \log \beta_{x_1} \\
&= \sum_{x_1} p\left[y_1, x_1 | \boldsymbol{\Theta}_*\right] \cdot p\left[y_2, \ldots, y_n | x_1, \Theta\right] \cdot \log \beta_{x_1} \\
&= \sum_{x_1} p\left[\boldsymbol{y}, x_1 | \boldsymbol{\Theta}_*\right] \cdot \log \beta_{x_1}
\end{aligned}
$$

we can establish

$$
\beta_x = \frac{p\left[\boldsymbol{y}, x_1 = x | \boldsymbol{\Theta}_*\right]}{p\left[\boldsymbol{y} | \boldsymbol{\Theta}_*\right]} = p\left[x_1 = x | \boldsymbol{\Theta}_*, \boldsymbol{y}\right]
$$

Simplification of the second term

For the term

$$Q_2 = \sum_{i=2}^{n} \sum_{\boldsymbol{x}} p\left[y_1, x_1 | \boldsymbol{\Theta_*}\right] \cdot \prod_{j=2}^{n} p\left[y_j, x_j | x_{j-1}, \boldsymbol{\Theta}_*\right] \cdot \log(\alpha_{x_{i-1}x_i})$$

we can use general equality $\sum_{\boldsymbol{x}} \prod_{\ell=1}^{n} a_{\ell x_\ell} = \prod_{\ell=1}^{n} \sum_{j=1}^{k} a_{\ell j}$ for getting

$$Q_2 = \sum_{i=2}^{n} \sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \log \alpha_{x,x'} \cdot p\left[\boldsymbol{y}, x_{i-1} = x, x_i = x' | \boldsymbol{\Theta}_*\right]$$

The latter allows to establish

$$\alpha_{xx'} = \frac{\sum_{i=2}^{n} \Pr\left[\boldsymbol{y}, x_{i-1} = x, x_i = x' | \boldsymbol{\Theta}_*\right]}{\sum_{i=2}^{n} \Pr\left[\boldsymbol{y}, x_{i-1} = x | \boldsymbol{\Theta}_*\right]}$$

Simplification of the third term

For the third term

$$Q_3 = \sum_{i=1}^{n} \sum_{\boldsymbol{x}} p\left[y_1, x_1 | \boldsymbol{\Theta_*}\right] \cdot \prod_{j=2}^{n} p\left[y_j, x_j | x_{j-1}, \boldsymbol{\Theta}_*\right] \cdot \log(p\left[y_i | x_i\right])$$

we can still use the general equality for getting

$$Q_3 = \sum_{i=1}^{n} \sum_{x \in \mathcal{X}} \log(p\left[y_i | x_i = x\right]) \cdot p\left[\boldsymbol{y}, x_i = x | \boldsymbol{\Theta}_*\right]$$

This term is identical to the term we maximise in the clustering algorithm.

Full recipe for discrete HMM

E-step. Compute the following marginal probabilities

$$\gamma_x(i) = \Pr\left[x_i = x | \boldsymbol{y}, \boldsymbol{\Theta}\right]$$
$$\xi_{xx'}(i) = \Pr\left[x_i = x, x_{i+1} = x' | \boldsymbol{y}, \boldsymbol{\Theta}\right]$$

M-step. Compute the following parameters

$$\beta_x = \gamma_x(1)$$
$$\alpha_{xx'} = \frac{\sum_{j=1}^{n-1} \xi_{xx'}(j)}{\sum_{j=1}^{n-1} \gamma_x(j)}$$
$$\delta_{xy} = \frac{\sum_{j=1}^{n-1} \gamma_x(j) \cdot [y_j = y]}{\sum_{j=1}^{n} \gamma_x(j)}$$

Full recipe for contiuous HMM

E-step. Compute the following marginal probabilities

$$\gamma_x(i) = \Pr\left[x_i = x | \boldsymbol{y}, \boldsymbol{\Theta}\right]$$
$$\xi_{xx'}(i) = \Pr\left[x_i = x, x_{i+1} = x' | \boldsymbol{y}, \boldsymbol{\Theta}\right]$$

M-step. Compute the following parameters

$$\beta_x = \gamma_x(1)$$
$$\alpha_{xx'} = \frac{\sum_{j=1}^{n-1} \xi_{xx'}(j)}{\sum_{j=1}^{n-1} \gamma_x(j)}$$

and find parameters $\boldsymbol{\mu}_j, \Sigma_j$ for the normal distribution by doing maximum likelihood fit for the datapoints with weights $w_{ix} = \Pr\left[x_i = x | \boldsymbol{y}, \boldsymbol{\Theta}_*\right]$.

EM-algorithm for Kalman filter

Lower bound function

The lower bound function used in the EM algorithm is current notation

$$F(\boldsymbol{q},\boldsymbol{\Theta}) = -\int_{\boldsymbol{x}} q(\boldsymbol{x})\cdot\log q(\boldsymbol{x})d\boldsymbol{x} + \int_{\boldsymbol{x}} q(\boldsymbol{x})\cdot\log\left(p\left[\boldsymbol{\Theta},\boldsymbol{x}|\boldsymbol{y},\boldsymbol{u}\right]\right)d\boldsymbol{x}$$

If we assign non-informative prior to the model parameters then in M-step it is sufficient to maximise the function

$$F_* = \int_{\boldsymbol{x}} q(\boldsymbol{x})\cdot\log\left(p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta},\boldsymbol{u}\right]\right)d\boldsymbol{x}$$

Probability assignment in E-step

According to the theory the optimal probability assignment is

$$q(\boldsymbol{x}) = \Pr\left[\boldsymbol{x}|\boldsymbol{y},\boldsymbol{u},\boldsymbol{\Theta}_*\right] = \frac{p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}_*,\boldsymbol{u}\right]}{p\left[\boldsymbol{y}|\boldsymbol{\Theta}_*,\boldsymbol{u}\right]}$$

and thus we get

$$F_* = \frac{1}{p\left[\boldsymbol{y}|\boldsymbol{\Theta}_*,\boldsymbol{u}\right]} \cdot \int_{\boldsymbol{x}} p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}_*,\boldsymbol{u}\right] \cdot \log\left(p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta},\boldsymbol{u}\right]\right) d\boldsymbol{x}$$

and thus it is sufficient to maximise

$$Q = \int_{\boldsymbol{x}} p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta}_*,\boldsymbol{u}\right] \cdot \log\left(p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta},\boldsymbol{u}\right]\right) d\boldsymbol{x}$$

Further decomposition

As the log-likelihood decomposes into two independent parameter groups

$$\log\left(p\left[\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\Theta},\boldsymbol{u}\right]\right)=\sum_{i=1}^{n}\log(p\left[\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1},\boldsymbol{\Theta}\right])+\sum_{i=1}^{n}\log(p\left[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{u}_i,\boldsymbol{\Theta}\right])$$

where

$$p\left[\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1},\boldsymbol{\Theta}\right]) = p_{\mathcal{N}}[\boldsymbol{x}_i - A\boldsymbol{x}_{i-1} - B\boldsymbol{u}_{i-1}|\Sigma_1]$$
$$p\left[\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{u}_i,\boldsymbol{\Theta}\right]) = p_{\mathcal{N}}[\boldsymbol{y}_i - C\boldsymbol{x}_i - D\boldsymbol{u}_i|\Sigma_2]$$

we can solve two independent maximisation tasks in the M-step. Again, finding Q-function seems to be a daunting task but the minimisation task can be reduced to finding marginal distributions as for the HMM.