HMM: Hidden Markov Models

Jean-Cédric Chappelier

Jean-Cedric.Chappelier@epfl.ch

Artificial Intelligence Laboratory

Objectives of this lecture

➥ Introduce fundamental concepts necessary to use HMMs for PoS tagging

Example: PoS tagging with HMM

Sentence to tag: Time flies like an arrow.

Example of HMM model:

- PoS tags: $\mathcal{T} = \{\texttt{Adj}, \texttt{Adv}, \texttt{Det}, \texttt{N}, \texttt{V}, \ldots\}$
- Transition probabilities:
  $P(\texttt{N}|\texttt{Adj}) = 0.1, P(\texttt{V}|\texttt{N}) = 0.3, P(\texttt{Adv}|\texttt{N}) = 0.01, P(\texttt{Adv}|\texttt{V}) = 0.005,$
  $P(\texttt{Det}|\texttt{Adv}) = 0.1, P(\texttt{Det}|\texttt{V}) = 0.3, P(\texttt{N}|\texttt{Det}) = 0.5$

  (plus all the others, such that stochastic constraints are fullfilled)
- Initial probabilities: $P_I(\texttt{Adj}) = 0.01, P_I(\texttt{Adv}) = 0.001, P_I(\texttt{Det}) = 0.1,$
  $P_I(\texttt{N}) = 0.2, P_I(\texttt{V}) = 0.003$ (+...)
- Words: $\mathcal{L} = \{$*an*, *arrow*, *flies*, *like*, *time*, $\ldots\}$
- Emission probabilities: $P(\textit{time}|\texttt{N}) = 0.1, P(\textit{time}|\texttt{Adj}) = 0.01, P(\textit{time}|\texttt{V}) = 0.05, P(\textit{flies}|\texttt{N}) = 0.1, P(\textit{flies}|\texttt{V}) = 0.01, P(\textit{like}|\texttt{Adv}) = 0.005, P(\textit{like}|\texttt{V}) = 0.1, P(\textit{an}|\texttt{Det}) = 0.3, P(\textit{arrow}|\texttt{N}) = 0.5$ (+...)

In this example, 12 = $3 \cdot 2 \cdot 2 \cdot 1 \cdot 1$ analyses are possible, for example:

$$P(\textit{time}/\texttt{N}\ \textit{flies}/\texttt{V}\ \textit{like}/\texttt{Adv}\ \textit{an}/\texttt{Det}\ \textit{arrow}/\texttt{N}) = 1.13 \cdot 10^{-11}$$
$$P(\textit{time}/\texttt{Adj}\ \textit{flies}/\texttt{N}\ \textit{like}/\texttt{V}\ \textit{an}/\texttt{Det}\ \textit{arrow}/\texttt{N}) = 6.75 \cdot 10^{-10}$$

$$\begin{aligned}
&P(\textit{time}/\texttt{N}\ \textit{flies}/\texttt{V}\ \textit{like}/\texttt{Adv}\ \textit{an}/\texttt{Det}\ \textit{arrow}/\texttt{N}) \\
&= P_I(\texttt{N}) \cdot P(\textit{time}|\texttt{N}) \cdot P(\texttt{V}|\texttt{N}) \cdot P(\textit{flies}|\texttt{V}) \cdot P(\textit{Adv}|\texttt{V}) \cdot P(\textit{like}|\texttt{Adv}) \\
&\quad \cdot P(\texttt{Det}|\texttt{Adv}) \cdot P(\textit{an}/\texttt{Det}) \cdot P(\texttt{N}|\texttt{Det}) \cdot P(\textit{arrow}|\texttt{N}) \\
&= \text{2e-1} \cdot \text{1e-1} \cdot \text{3e-1} \cdot \text{1e-2} \cdot \text{5e-3} \cdot \text{5e-3} \cdot \text{1e-1} \cdot \text{3e-1} \cdot \text{5e-1} \cdot \text{5e-1}
\end{aligned}$$

The aim is to choose the most probable tagging

Contents

- HMM models, three basic problems
- Forward-Backward algorithms
- Viterbi algorithm
- Baum-Welch algorithm

Markov Models

Markov model: a discrete-time stochastic process $\mathbf{T}$ on $\mathcal{T} = \left\{t^{(1)}, ..., t^{(m)}\right\}$ satisfying the *Markov property* (limited conditioning)

$$P(T_i|T_1, ..., T_{i-1}) = P(T_i|T_{t-k}, ..., T_{i-1})$$

k : *order* of the Markov model

In practice $k = 1$ (bigrams) or 2 (trigrams) rarely 3 or 4 ($\longrightarrow$ learning difficulties)

From a theoretical point of view: every Markov model of order k can be represented as another Markov model of order 1 (choose $Y_i = (T_{i-k+1}, ..., T_i)$)

Vocable:

$$P(T_1, ..., T_i) = P(T_1) \cdot P(T_2|T_1) \cdot ... \cdot P(T_i|T_{i-1})$$

initial probabilities transition probabilities

Hidden Markov Models (HMM)

What is hidden?

☞ The model itself (i.e. the state sequence)

What do we see then?

☞ An *observation* w related to the state (but not the state itself)

Formally:

- a set of states $\mathcal{T} = \{t^{(1)}, ..., t^{(m)}\}$ PoS tags
- a transition probabilities matrix $\mathbf{A}$ such that $A_{tt'} = P(T_{i+1} = t' | T_i = t)$, shorten $P(t'|t)$ (independant of i)
- an initial probabilities vector $\mathbf{I}$ such that $I_t = P(T_1 = t)$, shorten $P_I(t)$
- ☆ an alphabet $\mathcal{L}$ (not necessarily finite) words
- ☆ n probability densities on $\mathcal{L}$ (*emission probabilities*): $B_t(w) = P(W_i = w | T_i = t)$ (for $w \in \mathcal{L}$), shorten $P(w|t)$.

Simple example of HMM

Example: a cheater tossing from two hidden (unfair) coins

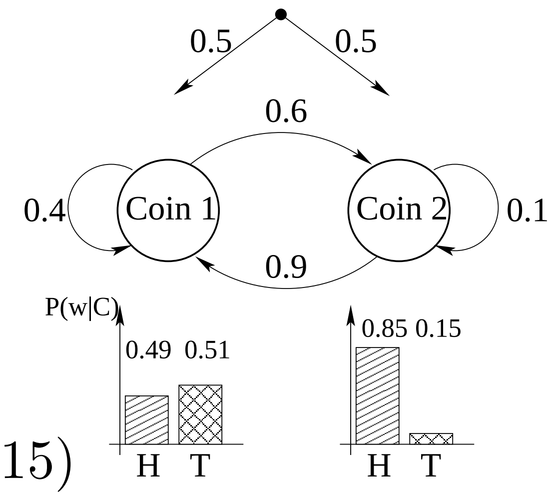
States: coin 1 and coin 2: $\mathcal{T} = \{1, 2\}$

transition matrix $\mathbf{A} = \begin{bmatrix} 0.4 & 0.6 \\ 0.9 & 0.1 \end{bmatrix}$

alphabet = $\{\mathtt{H}, \mathtt{T}\}$

emission probabilities $\mathbf{B_1} = (0.49, 0.51)$ et $\mathbf{B_2} = (0.85, 0.15)$

initial probabilities $\mathbf{I} = (0.5, 0.5)$



☞ 5 free parameters: I_1, A_{11}, A_{21}, $B_1(\mathtt{H})$, $B_2(\mathtt{H})$

Observation: HTTHTTHHTTHTTTHHTHHTTHTTTTHTHHTHTHHTTTH

[(Hidden) sequence of states: 2112112111211121121112111121211212111112]

The three basic problems for HMMs

Problems: Given an HMM and an observation sequence $\mathbf{w} = w_1 ... w_n$

- given the parameters $\boldsymbol{\theta}$ of the HMM, what is the probability of the observation sequence: $P(\mathbf{w}|\boldsymbol{\theta})$

  Application: Language Identification

- given the parameters $\boldsymbol{\theta}$ of the HMM, find the most likely state sequence $\mathbf{t}$ that produces $\mathbf{w}$: $\underset{\mathbf{t}}{\mathrm{Argmax}}\, P(\mathbf{t}|\mathbf{w}, \boldsymbol{\theta})$

  Application: PoS Tagging, Speech recognition

- find the parameters that maximize the probability of producing $\mathbf{w}$: $\underset{\boldsymbol{\theta}}{\mathrm{Argmax}}\, P(\boldsymbol{\theta}|\mathbf{w})$

  Application: Unsupervised learning

Remarks:

❶
$$
\begin{aligned}
\boldsymbol{\theta} &= (\mathbf{I}, \mathbf{A}, \mathbf{B}) \\
&= (I_1, ..., I_m, B_1(w_1), B_1(w_2), ..., B_1(w_L), B_2(w_1), ..., B_2(w_L), \\
&\quad ..., B_m(w_1), ..., B_m(w_L), A_{11}, ..., A_{1m}, ..., A_{m1}, ..., A_{mm})
\end{aligned}
$$

i.e. $(m-1) + m \cdot (L-1) + m \cdot (m-1) = m \cdot (m+L-1) - 1$ free parameters (because of sum-to-1 contraints), where $m = |\mathcal{T}|$ and $L = |\mathcal{L}|$ (in the finite case, otherwise L stands for the total number of parameters used to represent $\mathcal{L}$)

❷ Supervised learning (i.e $\underset{\boldsymbol{\theta}}{\mathrm{Argmax}}\, P(\boldsymbol{\theta}|\mathbf{w}, \mathbf{t})$) is easy

❸ WARNING! There is a difference between $P(\boldsymbol{\theta}|\mathbf{w})$ and $P(\mathcal{M}|\mathbf{w})$!
The model $\mathcal{M}$ is supposed to be known here, but its parameters $\boldsymbol{\theta}$: i.e. the HMM design is already defined (number of states, alphabet) only the parameters are missing.

Contents

- ➥ HMM models, three basic problems
- ☞ Forward-Backward algorithms
- ➥ Viterbi algorithm
- ➥ Baum-Welch algorithm

Computation of $P(\mathbf{W}|\boldsymbol{\theta})$

Computation of $P(\mathbf{W}|\boldsymbol{\theta})$ is mathematically trivial:

$$P(\mathbf{W}|\boldsymbol{\theta}) = \sum_{\mathbf{t}} P(\mathbf{W}, \mathbf{t}|\boldsymbol{\theta}) = \sum_{\mathbf{t}} P(\mathbf{W}|\mathbf{t}, \boldsymbol{\theta}) \cdot P(\mathbf{t}|\boldsymbol{\theta})$$

Practical limitation: complexity is $\mathcal{O}(n\, m^n)$ $\rightsquigarrow$ exponential!

Practical computation: forward/backward algorithms $\longrightarrow$ complexity is $\mathcal{O}(n\, m^2)$

Forward-Backward algorithms

"forward" variable : $\alpha_i(t) = P(w_1, ..., w_i, T_i = t|\boldsymbol{\theta}) \qquad t \in \mathcal{T}$

iterative computation: $\alpha_{i+1}(t') = B_{t'}(w_{i+1}) \cdot \sum_{t \in \mathcal{T}} \left(\alpha_i(t) \cdot A_{tt'}\right)$

$\alpha_1(t) = B_t(w_1) \cdot I_t$

"backward" variable : $\beta_i(t) = P(w_{i+1}, ..., w_n|T_i = t, \boldsymbol{\theta})$

$\beta_{i-1}(t') = \sum_{t \in \mathcal{T}} \left(\beta_i(t) \cdot A_{t't} \cdot B_t(w_i)\right)$

$\beta_n(t) = 1$ (by convention, practical considerations)

Forward-Backward algorithms (2)

"forward-backward" variable : $\gamma_i(t) = P(T_i = t|\mathbf{w}, \boldsymbol{\theta})$

$$\gamma_i(t) = \frac{P(\mathbf{w}, T_i = t|\boldsymbol{\theta})}{P(\mathbf{w}|\boldsymbol{\theta})} = \frac{\alpha_i(t) \cdot \beta_i(t)}{\sum_{t' \in \mathcal{T}} \alpha_i(t') \cdot \beta_i(t')}$$

Computation in $\mathcal{O}(n\,m^2)$ $\rightarrow$ efficient solutions to "first problem":

$$P(\mathbf{w}|\boldsymbol{\theta}) = \sum_{t \in \mathcal{T}} P(\mathbf{w}, T_n = t|\boldsymbol{\theta}) = \sum_{t \in \mathcal{T}} \alpha_n(t)$$

$$P(\mathbf{w}|\boldsymbol{\theta}) = \sum_{t \in \mathcal{T}} \alpha_i(t) \cdot \beta_i(t) \qquad \forall i : 1 \leq i \leq n$$

Contents

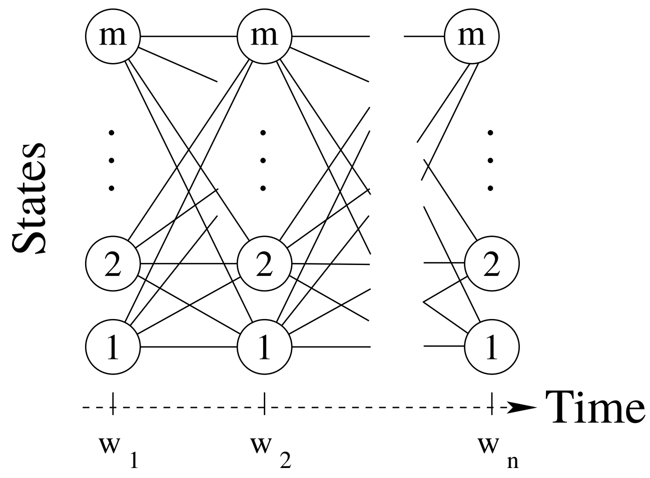
- ➥ HMM models, three basic problems
- ➥ Forward-Backward algorithms
- ☞ Viterbi algorithm
- ➥ Baum-Welch algorithm

Viterbi algorithm (1)

Efficient solution to the "second problem": find the most likely sequence of states $\mathbf{t}$ (knowing $\mathbf{w}$ and the parameters $\boldsymbol{\theta}$) : $\mathrm{Argmax}_{\mathbf{t}}\, P(\mathbf{t}|\mathbf{w},\boldsymbol{\theta})$

$\Rightarrow$ maximize (in $\mathbf{t}$) $P(\mathbf{t},\mathbf{w}|\boldsymbol{\theta})$.

"The" lattice ☞ temporal unfolding of all possible walks through the Markov chain



Viterbi algorithm (2)

Let $\rho_i(t) = \max_{t_1,...,t_{i-1}} P(t_1, ..., t_{i-1}, t_i = t, w_1, ..., w_i | \boldsymbol{\theta})$

We are looking for $\max_{t \in \mathcal{T}} \rho_n(t)$

It's easy (exercise) to show that $\rho_i(t) = \max_{t'} \left[P(t|t', \boldsymbol{\theta})\, P(w_i|t, \boldsymbol{\theta})\, \rho_{i-1}(t') \right]$

from which the following algorithm comes:

...

Viterbi algorithm (3)

for all $t \in \mathcal{T}$ **do**

$$\rho_1(t) = I_t \cdot B_t(w_1)$$

....

for i from 2 to n **do**

for all $t \in \mathcal{T}$ **do**

- $\rho_i(t) = B_t(w_i) \cdot \max_{t'}\big(A_{t't} \cdot \rho_{i-1}(t')\big)$
- mark one of the transitions from t' to t where the maximum is reached

....

reconstruct backwards (from t_n) the best path following the marked transitions

Contents

- ➥ HMM models, three basic problems
- ➥ Forward-Backward algorithms
- ➥ Viterbi algorithm
- ☞ Baum-Welch algorithm

Expectation-Maximization

Our goal: maximize $P(\boldsymbol{\theta}|\mathbf{w})$

☞ Maximum-likelihood estimation of $\boldsymbol{\theta}$ $\rightarrow$ maximization of $P(\mathbf{w}|\boldsymbol{\theta})$

To achieve it: **Expectation-Maximization (EM) algorithm**

General formulation of EM:

given

- observed data $\mathbf{w} = w_1 ... w_n$
- a parameterized probability distribution $P(\mathbf{T}, \mathbf{W}|\boldsymbol{\theta})$ where
  - $\mathbf{T} = T_1 ... T_n$ are unobserved data
  - $\boldsymbol{\theta}$ are the parameters of the model

determine $\boldsymbol{\theta}$ that maximizes $P(\mathbf{w}|\boldsymbol{\theta})$ by convergence of iterative computation of the series $\boldsymbol{\theta}^{(i)}$ that maximizes (in $\boldsymbol{\theta}$) $\mathbf{E}_{\mathbf{T}}\left[\log P(\mathbf{T}, \mathbf{W}|\boldsymbol{\theta})|\mathbf{w}, \boldsymbol{\theta}^{(i-1)}\right]$

Expectation-Maximization (2)

To do so, define the auxilary function

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbf{E_T}\left[\log P(\mathbf{T}, \mathbf{W}|\boldsymbol{\theta})|\mathbf{w}, \boldsymbol{\theta}'\right] = \sum_{\mathbf{t}} P(\mathbf{t}|\mathbf{w}, \boldsymbol{\theta}') \log P(\mathbf{t}, \mathbf{w}|\boldsymbol{\theta})$$

as it can be shown (with Jensen inequality) that

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}') > Q(\boldsymbol{\theta}', \boldsymbol{\theta}') \Rightarrow P(\mathbf{w}|\boldsymbol{\theta}) > P(\mathbf{w}|\boldsymbol{\theta}')$$

This is the fundamental principle of EM: **if** we already have an estimation $\boldsymbol{\theta}'$ of the parameters and we find another parameter configuration $\boldsymbol{\theta}$ for which the first inequality (on Q) holds, **then** $\mathbf{w}$ is most probable with model $\boldsymbol{\theta}$ rather than with model $\boldsymbol{\theta}'$.

Expectation-Maximization (3)

EM algorithm:

- Estimation Step: Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$
- Maximization Step: Compute $\boldsymbol{\theta}^{(i+1)} = \underset{\boldsymbol{\theta}}{\mathrm{Argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$

in other words:

1. Choose $\boldsymbol{\theta}^{(0)}$ (and set $i = 0$)
2. Find $\boldsymbol{\theta}^{(i+1)}$ which maximizes $\sum_{\mathbf{t}} P(\mathbf{t}|\mathbf{w}, \boldsymbol{\theta}^{(i)}) \log P(\mathbf{t}, \mathbf{w}|\boldsymbol{\theta}^{(i+1)})$
3. Set $i \leftarrow i + 1$ and go back to (2) unless some convergence test is fulfilled

Baum-Welch Algorithm

The Baum-Welch Algorithm is an EM algorithm for estimating HMM parameters. It's an answer to the "third problem".

The goal is therefore to find

$$\operatorname*{Argmax}_{\boldsymbol{\theta}} \sum_{\mathbf{t}} P(\mathbf{t}|\mathbf{w},\boldsymbol{\theta}') \log P(\mathbf{t},\mathbf{w}|\boldsymbol{\theta}) = \operatorname*{Argmax}_{\boldsymbol{\theta}} \underbrace{\sum_{\mathbf{t}} P(\mathbf{t},\mathbf{w}|\boldsymbol{\theta}') \log P(\mathbf{t},\mathbf{w}|\boldsymbol{\theta})}_{\stackrel{\text{def}}{=} \widehat{Q}(\boldsymbol{\theta},\boldsymbol{\theta}')}$$

since $P(\mathbf{w}|\boldsymbol{\theta}')$ does not depend on $\boldsymbol{\theta}$.

What is $\log P(\mathbf{t},\mathbf{w}|\boldsymbol{\theta})$?

$$\log P(\mathbf{t},\mathbf{w}|\boldsymbol{\theta}) = \log P_I(t_1) + \sum_{i=2}^{n} \log P(t_i|t_{i-1}) + \sum_{i=1}^{n} \log P(w_i|t_i)$$

$\widehat{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ consists therefore of 3 terms:

$$\widehat{Q}((\mathbf{I}, \mathbf{A}, \mathbf{B}), \boldsymbol{\theta}') = Q_I(\mathbf{I}, \boldsymbol{\theta}') + Q_A(\mathbf{A}, \boldsymbol{\theta}') + Q_B(\mathbf{B}, \boldsymbol{\theta}')$$

Let's compute one of these:

$$\begin{aligned}
Q_I(\mathbf{I}, \boldsymbol{\theta}') &= \sum_{\mathbf{t}} P(\mathbf{t}, \mathbf{w}|\boldsymbol{\theta}') \log P_I(t_1) \\
&= \sum_{t_1} \sum_{t_2,\ldots,t_n} P(t_1, \mathbf{w}|\boldsymbol{\theta}') \cdot P(t_2, \ldots t_n|t_1, \mathbf{w}, \boldsymbol{\theta}') \cdot \log P_I(t_1) \\
&= \sum_{t \in \mathcal{T}} P(t_1 = t, \mathbf{w}|\boldsymbol{\theta}') \cdot \log P_I(t) \underbrace{\sum_{t_2,\ldots,t_n} P(t_2, \ldots, t_n|t_1, \mathbf{w}, \boldsymbol{\theta}')}_{=1} \\
&= \sum_{t \in \mathcal{T}} P(t_1 = t, \mathbf{w}|\boldsymbol{\theta}') \cdot \log I_t
\end{aligned}$$

Similarly we have:

$$Q_A(\mathbf{A}, \boldsymbol{\theta}') = \sum_{i=2}^{n} \sum_{t \in \mathcal{T}} \sum_{t' \in \mathcal{T}} P(T_{i-1} = t, T_i = t', \mathbf{w} | \boldsymbol{\theta}') \log A_{tt'}$$

$$Q_B(\mathbf{B}, \boldsymbol{\theta}') = \sum_{i=2}^{n} \sum_{t \in \mathcal{T}} P(T_i = t, \mathbf{w} | \boldsymbol{\theta}') \log B_t(w_i)$$

Therefore $\widehat{Q}$ is a sum of three **independent** terms (e.g. Q_I does not depend on $\mathbf{A}$ nor on $\mathbf{B}$) and therefore the maximisation over $\boldsymbol{\theta}$ is achieved by the three terms separately, i.e. maximizing $Q_I(\mathbf{I}, \boldsymbol{\theta}')$ over $\mathbf{I}$, $Q_A(\mathbf{A}, \boldsymbol{\theta}')$ over $\mathbf{A}$ and $Q_B(\mathbf{B}, \boldsymbol{\theta}')$ over $\mathbf{B}$ separately.

Notice that all these three functions are sums (over i) of functions of the form:

$$f(\mathbf{x}) = \sum_{j=1}^{m} y_j \log x_j$$

and all the above three functions have to be maximized under the constraint $\sum_{j=1}^{m} x_j = 1$.[a]

This maximization under constraints is achieved using Lagrange multipliers, i.e. looking at

$$g(\mathbf{x}) = f(\mathbf{x}) - \lambda \cdot \sum_{j=1}^{m} x_j = \sum_{j=1}^{m} (y_j \log x_j - \lambda \cdot x_j)$$

Solving this by $\frac{\partial}{\partial x} g(x) = 0$, we find that $\lambda = \frac{y_j}{x_j}$.

Putting this back in the constraint we find:

$$x_j = \frac{y_j}{\sum_{j=1}^{m} y_j}$$

[a] To be accurate: for $\mathbf{B_t}$ the constraint is $\sum_{w \in \mathcal{L}} B_t(w) = 1$. This changes the formulas a bit, but not the essence of the computation.

Summarizing the obtained results, we have the following reestimation formulas (where the max. is reached):

$$\widehat{I_t} = \frac{P(T_1 = t, \mathbf{w}|\boldsymbol{\theta}')}{\sum\limits_{t' \in \mathcal{T}} P(T_1 = t', \mathbf{w}|\boldsymbol{\theta}')} = \frac{P(T_1 = t, \mathbf{w}|\boldsymbol{\theta}')}{P(\mathbf{w}|\boldsymbol{\theta}')}$$

$$\widehat{A_{tt'}} = \frac{\sum\limits_{i=2}^{n} P(T_{i-1} = t, T_i = t', \mathbf{w}|\boldsymbol{\theta}')}{\sum\limits_{i=2}^{n} \sum\limits_{\tau \in \mathcal{T}} P(T_{i-1} = t, T_i = \tau, \mathbf{w}|\boldsymbol{\theta}')}$$

$$= \frac{\sum\limits_{i=2}^{n} P(T_{i-1} = t, T_i = t', \mathbf{w}|\boldsymbol{\theta}')}{\sum\limits_{i=2}^{n} P(T_{i-1} = t, \mathbf{w}|\boldsymbol{\theta}')}$$

and:

$$\widehat{B_t(w)} \quad = \quad \frac{\displaystyle\sum_{\substack{i=1 s.t.\\ w_i=w}}^{n} P(T_i=t,\mathbf{w}|\boldsymbol{\theta}')}{\displaystyle\sum_{i=2}^{n} P(T_i=t,\mathbf{w}|\boldsymbol{\theta}')} = \frac{\displaystyle\sum_{i=2}^{n} P(T_i=t,\mathbf{w}|\boldsymbol{\theta}')\,\delta_{w_i,w}}{\displaystyle\sum_{i=2}^{n} P(T_i=t,\mathbf{w}|\boldsymbol{\theta}')}$$

with $\delta_{w,w'} = 1$ if $w = w'$ and 0 otherwise.

Baum-Welch Algorithm: effective computation

How do we compute these reestimation formulas?

Let $\chi_i(t,t') = P(T_i = t, T_{i+1} = t' | \mathbf{w})$

χ_i is easy to compute with "forward" and "backward" variables:

$$\chi_i(t,t') = \frac{\alpha_i(t) \cdot A_{tt'} \cdot B_{t'}(w_{i+1}) \cdot \beta_{i+1}(t')}{\sum_{\tau \in \mathcal{T}} \sum_{\tau' \in \mathcal{T}} \alpha_i(\tau) \cdot A_{\tau\tau'} \cdot B_{\tau'}(w_{i+1}) \cdot \beta_{i+1}(\tau')}$$

Notice: $\gamma_i(t) = \sum_{t' \in \mathcal{T}} \chi_i(t,t')$ for all $1 \leq i < n$

Effective Reestimation formulas

$$\widehat{I_t} = \gamma_1(t)$$

$$\widehat{A_{tt'}} = \frac{\displaystyle\sum_{i=1}^{n-1} \chi_i(t,t')}{\displaystyle\sum_{i=1}^{n-1} \gamma_i(t)}$$

$$\widehat{B_t(w)} = \frac{\displaystyle\sum_{\substack{i=1 s.t.\\ w_i=w}}^{n} \gamma_i(t)}{\displaystyle\sum_{i=1}^{n} \gamma_i(t)} = \frac{\displaystyle\sum_{i=1}^{n} \gamma_i(t)\,\delta_{w_i,w}}{\displaystyle\sum_{i=1}^{n} \gamma_i(t)}$$

with $\delta_{w,w'} = 1$ if $w = w'$ and 0 otherwise.

Baum-Welch Algorithm

1. Let $\boldsymbol{\theta}^{(0)}$ be an initial parameter set
2. Compute iteratively α, β and then γ and χ
3. Compute $\boldsymbol{\theta}^{(t+1)}$ with reestimation formulas
4. If $\boldsymbol{\theta}^{(t+1)} \neq \boldsymbol{\theta}^{(t)}$, go to (2) [or another weaker stop test]

WARNING!

The algorithm converges but only towards a **local** maximum of $\mathbf{E}\left[\log P(\mathbf{T}, \mathbf{W}|\boldsymbol{\theta})\right]$

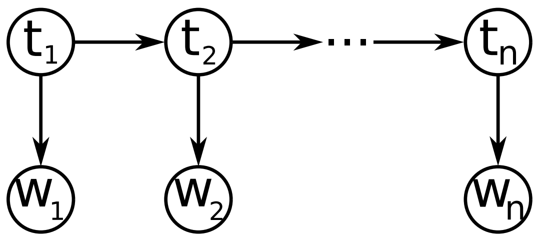
CRF versus HMM

(linear) **Conditional Random Fields** (CRF) are a discriminative generalization of the HMMs where “features” no longer needs to be state-conditionnal probabilities (less constraint).

For instance (order 1):

HMM

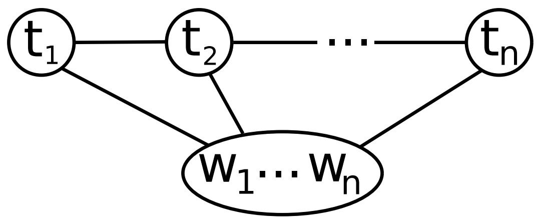
$$P(\mathbf{t},\mathbf{w}) = P(t_1)\,P(w_1|t_1)\cdot \prod_{i=2}^{n} P(w_i|t_i)\,P(t_i|t_{i-1})$$



CRF

$$P(\mathbf{t}|\mathbf{w}) = \prod_{i=2}^{n} P(t_{i-1},t_i|\mathbf{w})$$

(with $P(t_{i-1},t_i|\mathbf{w}) \propto \exp\left(\sum_j \lambda_j f_j(t_{i-1},t_i,\mathbf{w},i)\right)$)



Keypoints

- HMMs definitions, their applications
- Three basic problems for HMMs
- Algorithms needed to solve these problems: Forward-Backward, Viterbi, Baum-Welch (be aware of their existence, but not the implementation details)

References

[1] L. R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, vol. 77, No. 2, 1989.

[2] C. D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT, 1999.

[3] A. P. Dempster, N. M. Laird, D. B. Rubin, *Maximum-likelihood from incomplete data via the EM algorithm*, Journal of Royal Statistical Society B, 1977.

[4] H. Bourlard et al., *Traitement de la parole*, 2000; pp. 179-200, 202-214, 232-260.

APPENDUM

Justification of the maximization of the auxilary function Q for finding $\boldsymbol{\theta}$ maximizing $P(\mathbf{w}|\boldsymbol{\theta})$:

$$
\begin{aligned}
\log P(\mathbf{w}|\boldsymbol{\theta}) - \log P(\mathbf{w}|\boldsymbol{\theta}') = \log \frac{P(\mathbf{w}|\boldsymbol{\theta})}{P(\mathbf{w}|\boldsymbol{\theta}')} &= \log \sum_{\mathbf{t}} \frac{P(\mathbf{w},\mathbf{t}|\boldsymbol{\theta})}{P(\mathbf{w}|\boldsymbol{\theta}')} \\
&= \log \sum_{\mathbf{t}} P(\mathbf{t}|\mathbf{w},\boldsymbol{\theta}') \frac{P(\mathbf{w},\mathbf{t}|\boldsymbol{\theta})}{P(\mathbf{w},\mathbf{t}|\boldsymbol{\theta}')} \\
&\overset{\text{Jensen}}{\geq} \sum_{\mathbf{t}} P(\mathbf{t}|\mathbf{w},\boldsymbol{\theta}') \log \frac{P(\mathbf{w},\mathbf{t}|\boldsymbol{\theta})}{P(\mathbf{w},\mathbf{t}|\boldsymbol{\theta}')} \\
&\geq \mathbf{E}_{\mathbf{T}}\left[\log P(\mathbf{T},\mathbf{W}|\boldsymbol{\theta})|\mathbf{w},\boldsymbol{\theta}'\right] - \mathbf{E}_{\mathbf{T}}\left[\log P(\mathbf{T},\mathbf{W}|\boldsymbol{\theta}')|\mathbf{w},\boldsymbol{\theta}'\right] \\
&\geq Q(\boldsymbol{\theta},\boldsymbol{\theta}') - Q(\boldsymbol{\theta}',\boldsymbol{\theta}')
\end{aligned}
$$

Therefore:

$$
Q(\boldsymbol{\theta},\boldsymbol{\theta}') > Q(\boldsymbol{\theta}',\boldsymbol{\theta}') \Rightarrow \log P(\mathbf{w}|\boldsymbol{\theta}) > \log P(\mathbf{w}|\boldsymbol{\theta}') \Rightarrow P(\mathbf{w}|\boldsymbol{\theta}) > P(\mathbf{w}|\boldsymbol{\theta}')
$$