Learning Partially Observable Markov Models from First Passage Times

Jérôme Callut and Pierre Dupont

European Conference on Machine Learning (ECML)
18 September 2007

Computing science & engineering dept. (INGI)
UCL Machine Learning Group

Outline

1. FPT in models and sequences
2. Partially Observable Markov Models (POMMs)
3. FPT dynamics in POMMs
4. POMM induction: POMMStruct
5. Experimental results

HMM induction

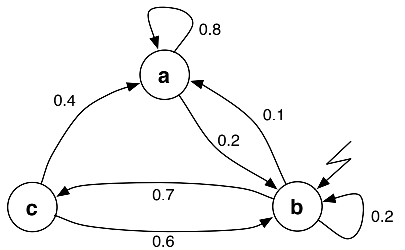
Problem: Estimate the **model structure** and its **probabilistic parameters** from observed sequences

To do what?
- Predict the future outcomes of the process
- **Predict when future events will occur**

Special focus: **First Passage Times** (FPT) between events of interest

Contribution: A novel **induction algorithm** to induce models from FPT

FPT in models and sequences



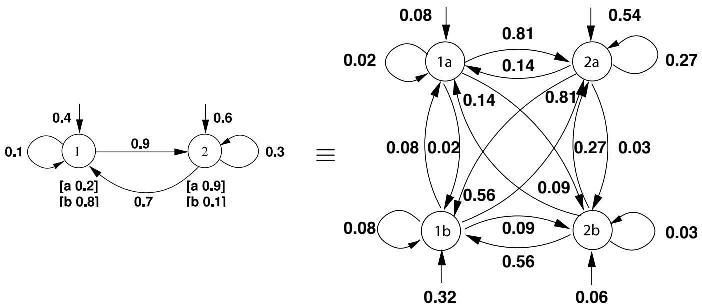
s = b c b b c a a a b c b c b a a a a a a b c a a a a a

FPT(c, a) = {4, 1, 4, 2, 1}

- ***FPT statistics*** can be computed from ***models*** or from ***sequences***
- The ***FPT dynamics*** of a process denotes its ***FPT distributions***
- FPT features are time ***unbounded features*** unlike N-grams
  ⇒ characterize long-term dependencies and temporal dynamics

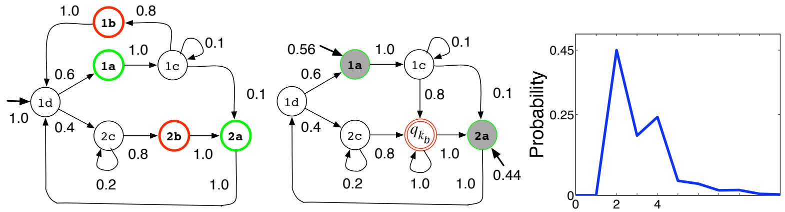
Partially Observable Markov Models (POMMs)

A ***POMM*** is a HMM such that any state emits a ***single letter*** with probability 1. The *same letter* can be emitted by *several states*.



We have shown that for any HMM there is an ***equivalent*** POMM.
⇒ We use POMMs as ***target formalism*** (convenient FPT computations).

FPT dynamics in POMMs

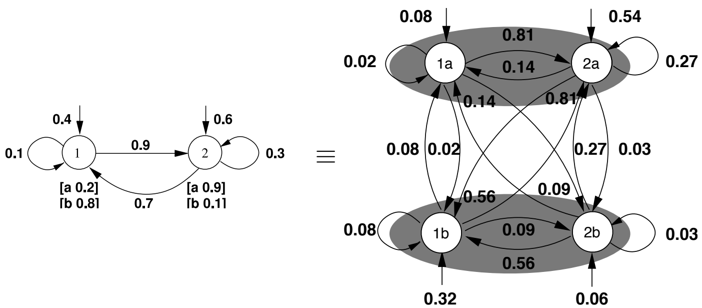


FPT(a,b) — *Time to absorption* — Time to absorption

- The distributions of FPT in POMMs are of ***phase-type***
  - Merge states `1b` and `2b` into an absorbing state
  - Start in `1a` or `2a` according to the relative proportion of time spent in these states
- The ***multimodal*** distribution reveals the presence of dominant path lengths
- ***Long-term dependencies*** are also reflected in the FPT dynamics

Partially Observable Markov Models (POMMs)

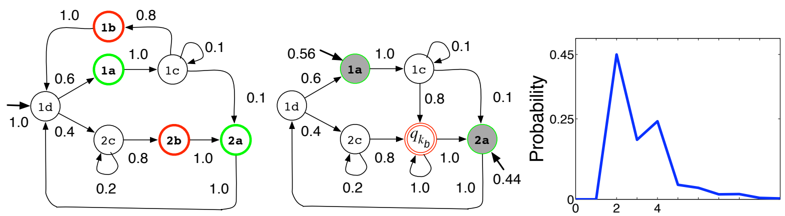
A ***POMM*** is a HMM such that any state emits a ***single letter*** with probability 1. The *same letter* can be emitted by *several states*.



We have shown that for any HMM there is an ***equivalent*** POMM.
⇒ We use POMMs as ***target formalism*** (convenient FPT computations).

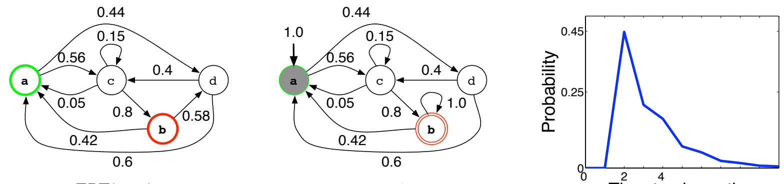
- ***States*** can be gathered in ***blocks*** w.r.t. their unique emission
- ***FPT*** observed in the ***sequences*** concern these ***blocks***

POMM dynamics is poorly approximated by MC



FPT(a,b) — *Time to absorption* — Time to absorption

Order 1 MC modeling of the POMM's distribution:
(estimated from 1000 sequences of length 100 from the POMM)



FPT(a,b) — *Time to absorption* — Time to absorption

POMM induction: POMMStruct

Algorithm POMMSTRUCT(S,r,p,n_r)

$EP_0 \leftarrow$ `initialize`(S, r);
$FPT \leftarrow$ `extractFPT`(S);
$\mathcal{P} \leftarrow$ `selectDivPairs`(EP, FPT, p);
$EP_0 \leftarrow$ POMMPHIT$(EP_0, FPT, \mathcal{P}, n_r)$;
$Lik \leftarrow$ `FPTLikelihood`(EP_0, FPT);
$i \leftarrow 0$

repeat
  $Lik_{last} \leftarrow Lik$;
  $\kappa_j \leftarrow$ `probeBlocks`(EP_i, FPT);
  $EP_{i+1} \leftarrow$ `addStateInBlock`(EP_i, κ_j);
  $EP_{i+1} \leftarrow$ POMMPHIT$(EP_{i+1}, FPT, \mathcal{P}, n_r)$;
  $Lik \leftarrow$ `FPTLikelihood`(EP_{i+1}, FPT);
  $i \leftarrow i + 1$
until $\frac{|Lik - Lik_{last}|}{|Lik_{last}|} < \epsilon$;
return $\{EP_0, \ldots, EP_i\}$

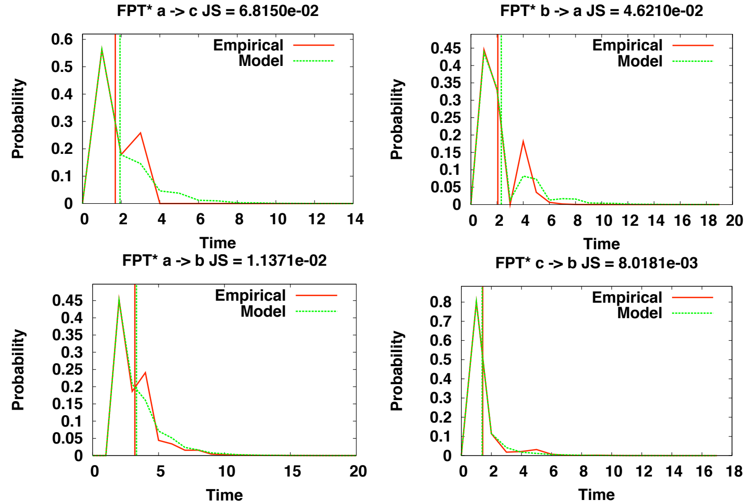
POMM induction: POMMStruct

Algorithm POMMSTRUCT(S,r,p,n_r)

$EP_0 \leftarrow$ `initialize`(S, r);
$FPT \leftarrow$ `extractFPT`(S);
$\mathcal{P} \leftarrow$ `selectDivPairs`(EP, FPT, p);
$EP_0 \leftarrow$ POMMPHIT$(EP_0, FPT, \mathcal{P}, n_r)$;
$Lik \leftarrow$ `FPTLikelihood`(EP_0, FPT);
$i \leftarrow 0$

repeat
  $Lik_{last} \leftarrow Lik$;
  $\kappa_j \leftarrow$ `probeBlocks`(EP_i, FPT);
  $EP_{i+1} \leftarrow$ `addStateInBlock`(EP_i, κ_j);
  $EP_{i+1} \leftarrow$ POMMPHIT$(EP_{i+1}, FPT, \mathcal{P}, n_r)$;
  $Lik \leftarrow$ `FPTLikelihood`(EP_{i+1}, FPT);
  $i \leftarrow i + 1$
until $\frac{|Lik - Lik_{last}|}{|Lik_{last}|} < \epsilon$;
return $\{EP_0, \ldots, EP_i\}$

Feature selection/weighting



- $D_{JS}(P_1||P_2) = H(M) - \frac{1}{2}H(P_1) - \frac{1}{2}H(P_2)$
  where $M = \frac{1}{2}(P_1 + P_2)$ and $H(.)$ is the Shannon entropy
- FPT pairs are ***filtered/weighted*** according to their JS divergence

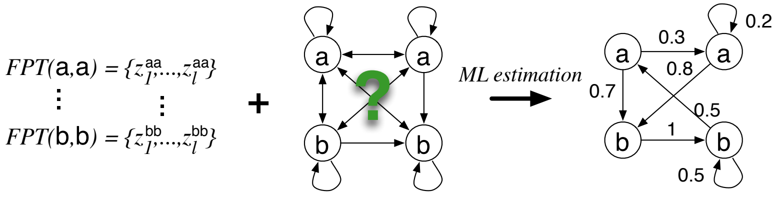
POMM induction: POMMStruct

Algorithm POMMSTRUCT(S,r,p,n_r)

$EP_0 \leftarrow$ `initialize`(S, r);
$FPT \leftarrow$ `extractFPT`(S);
$\mathcal{P} \leftarrow$ `selectDivPairs`(EP, FPT, p);
$EP_0 \leftarrow$ POMMPHIT$(EP_0, FPT, \mathcal{P}, n_r)$;
$Lik \leftarrow$ `FPTLikelihood`(EP_0, FPT);
$i \leftarrow 0$

repeat
  $Lik_{last} \leftarrow Lik$;
  $\kappa_j \leftarrow$ `probeBlocks`(EP_i, FPT);
  $EP_{i+1} \leftarrow$ `addStateInBlock`(EP_i, κ_j);
  $EP_{i+1} \leftarrow$ POMMPHIT$(EP_{i+1}, FPT, \mathcal{P}, n_r)$;
  $Lik \leftarrow$ `FPTLikelihood`(EP_{i+1}, FPT);
  $i \leftarrow i + 1$
until $\frac{|Lik - Lik_{last}|}{|Lik_{last}|} < \epsilon$;
return $\{EP_0, \ldots, EP_i\}$

Parameter estimation: POMMPHit



- POMMPHIT is a novel ***EM-based algorithm*** to maximize the FPT likelihood
- Each z_i^{ab} is a ***partial observation*** of a couple (z_i^{ab}, h_i) where h_i is the sequence of states reached during the FPT z_i^{ab}
- ***Re-estimation formulas*** are derived to maximize $\mathbb{E}\left[P(Z, H \mid \rho) \mid Z\right]$
- Additionally, a ***trimming procedure*** removes the transitions with the lowest expected passage times

Reestimation formula

Expectation step

$$\overline{S^{\mathsf{a,b}}}(q) = \sum_{k=1}^{l} c_k \frac{\sigma_q^{\mathsf{a}} \beta^{\mathsf{b}}(q, z_k)}{\sum_{q \in \kappa_{\mathsf{b}}} \alpha^{\mathsf{a,b}}(q, z_k)}$$

$$\overline{N^{\mathsf{a,b}}}(q, q') = \sum_{k=1}^{l} c_k \sum_{t=0}^{z_k - 1} \frac{\alpha^{\mathsf{a,b}}(q, t) A_{qq'} \beta^{\mathsf{b}}(q', z_k - t - 1)}{\sum_{q \in \kappa_{\mathsf{b}}} \alpha^{\mathsf{a,b}}(q, z_k)}$$

Maximization step

$$\sigma_q^{\kappa_{\mathsf{a}}} = \begin{cases} \dfrac{\sum_{\mathsf{b} \in \{\mathsf{b} \mid (\mathsf{a,b}) \in \mathcal{P}\}} \overline{S^{\mathsf{a,b}}}(q)}{\sum_{q \in \kappa_{\mathsf{a}}} \sum_{\mathsf{b} \in \{\mathsf{b} \mid (\mathsf{a,b}) \in \mathcal{P}\}} \overline{S^{\mathsf{a,b}}}(q)} & \text{if } q \in \kappa_{\mathsf{a}} \\ 0 & \text{otherwise} \end{cases}$$

$$A_{qq'} = \frac{\sum_{(\mathsf{a,b}) \in \mathcal{P}} \overline{N^{\mathsf{a,b}}}(q, q')}{\sum_{q' \in Q} \sum_{(\mathsf{a,b}) \in \mathcal{P}} \overline{N^{\mathsf{a,b}}}(q, q')}$$

Computational complexity: $\mathcal{O}(p\ L^2\ n_t)$ per iteration where L is the longest FPT and n_t is the number of transitions

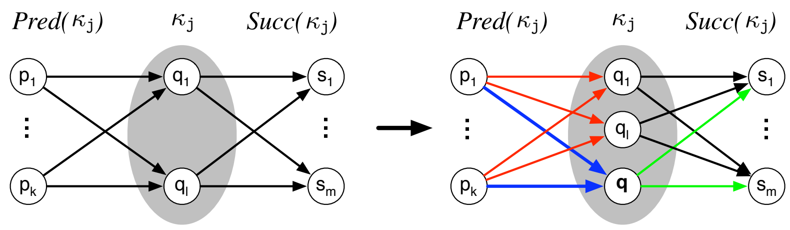
POMM induction: POMMStruct

Algorithm POMMSTRUCT(S,r,p,n_r)

$EP_0 \leftarrow$ initialize(S, r);
$FPT \leftarrow$ extractFPT(S);
$\mathcal{P} \leftarrow$ selectDivPairs(EP, FPT, p);
$EP_0 \leftarrow$ POMMPHIT$(EP_0, FPT, \mathcal{P}, n_r)$;
$Lik \leftarrow$ FPTLikelihood(EP_0, FPT);
$i \leftarrow 0$

repeat
$\quad Lik_{last} \leftarrow Lik$;
$\quad \kappa_{\mathsf{j}} \leftarrow$ probeBlocks(EP_i, FPT);
$\quad EP_{i+1} \leftarrow$ addStateInBlock$(EP_i, \kappa_{\mathsf{j}})$;
$\quad EP_{i+1} \leftarrow$ POMMPHIT$(EP_{i+1}, FPT, \mathcal{P}, n_r)$;
$\quad Lik \leftarrow$ FPTLikelihood(EP_{i+1}, FPT);
$\quad i \leftarrow i + 1$
until $\frac{|Lik - Lik_{last}|}{|Lik_{last}|} < \epsilon$;
return $\{EP_0, \ldots, EP_i\}$

Adding state in block



- $Pred(\kappa_j)$, κ_j and $Succ(\kappa_j)$ ***need not be disjoint***
- Local transitions are initialized following their type (color here)
- **POMMPHIT** first only estimates ***local transitions***. The trimming procedure is applied to these transitions.
- The ***complete model*** is then ***reestimated*** with **POMMPHIT** and all transitions are candidate to be trimmed

POMM induction: POMMStruct

Algorithm POMMSTRUCT(S,r,p,n_r)

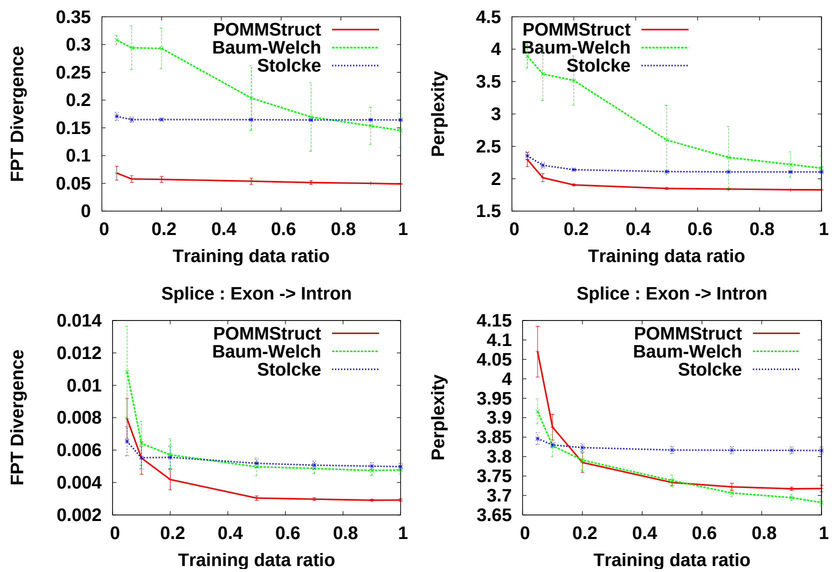
$EP_0 \leftarrow$ `initialize`(S, r);
$FPT \leftarrow$ `extractFPT`(S);
$\mathcal{P} \leftarrow$ `selectDivPairs`(EP, FPT, p);
$EP_0 \leftarrow$ POMMPHIT$(EP_0, FPT, \mathcal{P}, n_r)$;
$Lik \leftarrow$ `FPTLikelihood`(EP_0, FPT);
$i \leftarrow 0$

repeat

$Lik_{last} \leftarrow Lik$;
$\kappa_j \leftarrow$ `probeBlocks`(EP_i, FPT);
$EP_{i+1} \leftarrow$ `addStateInBlock`(EP_i, κ_j);
$EP_{i+1} \leftarrow$ POMMPHIT$(EP_{i+1}, FPT, \mathcal{P}, n_r)$;
$Lik \leftarrow$ `FPTLikelihood`(EP_{i+1}, FPT);
$i \leftarrow i + 1$

until $\frac{|Lik - Lik_{last}|}{|Lik_{last}|} < \epsilon$;
return $\{EP_0, \ldots, EP_i\}$

Experimental results



Conclusion and future work

- We proposed a ***novel approach to induce POMMs*** based on the FPT dynamics observed in the sample
- The FPT are ***informative*** about the ***structure*** of the model
- ***Structural induction*** is made by iterative state addition and by transition trimming
- ***Parameter estimation*** is performed by POMMPHIT which maximizes the model likelihood w.r.t. the observed FPT

Future work

- Return a HMM rather than a POMM
- Fit FPT between higher order events

HMM induction overview