Links between Probabilistic Automata and Hidden Markov Models

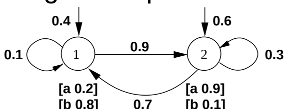
Pierre Dupont

pdupont@info.ucl.ac.be

– Typeset by FoilTEX –

Motivation

Hidden Markov Models (HMMs) are widely used in many pattern recognition areas (speech recognition, biological sequence modeling, etc.)



In most cases, the ***HMM structure***, also referred to as topology, is defined according to some ***prior knowledge*** of the application domain

Automatic techniques for ***inducing HMM topology*** are interesting as the structures are sometimes hard to define a priori or need to be tuned after some task adaptation

Several induction techniques have been developed for ***probabilistic automata (PA)***

> Stressing the ***links*** between PA and HMMs offers the possibility to apply PA induction techniques to learn HMM structures

Some classical misconceptions

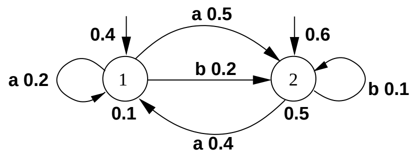
- The only difference between HMMs and PA is that symbols are attached to states in HMMs while they are attached to transitions in PA
- HMMs are more powerful than PA as they include transition probabilities ***and*** emission probabilities
- HMMs and PA are incomparable (except for very special cases)
- HMMs and PA are strictly equivalent and one can always transform a HMM into a PA with the same number of states ***and*** conversely
- HMMs with or without silent states are defining the same types of distributions
- . . .

Outline

- Probabilistic automata
  - Sufficient and necessary conditions to define a distribution
  - PDFA are strictly less general than PNFA
  - PNFA without final probabilities
- HMMs
  - HMM with state emission
  - HMM with transition emission (HMMT)
- Links between PA and HMMs
- Learnability results
- Open questions

Main results are quoted here. Detailed proofs available in additional reference.

A semi-probabilistic automaton (semi-PA)



A semi-probabilistic automaton $A = \langle \Sigma, Q, \phi, \iota, \tau \rangle$

- Σ finite ***alphabet***
- Q finite ***set of states***
- $\phi : Q \times \Sigma \times Q \rightarrow [0,1]$ ***transition probability function***
- $\iota : Q \rightarrow [0,1]$ ***initial probability*** $\sum_{q \in Q} \iota(q) = 1$
- $\tau : Q \rightarrow [0,1]$ ***final probability***

$$\forall q \in Q, \tau(q) + \sum_{a \in \Sigma} \sum_{q' \in Q} \phi(q, a, q') = 1$$

A state q is initial if $\iota(q) > 0$ and final if $\tau(q) > 0$
Note: ϕ also denotes *extensions* of the transition probability function

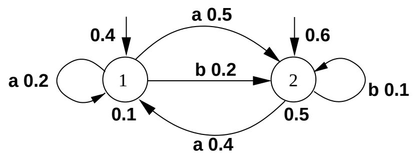
Generation Probability

Probability of generating ***prefix*** u

$$\overline{P}_A(u) = \sum_{q,q' \in Q} \iota(q)\phi(q,u,q')$$

Probability of generating ***word*** u

$$P_A(u) = \sum_{q,q' \in Q} \iota(q)\phi(q,u,q')\tau(q')$$



$$\begin{aligned} P_A(b) &= \iota(1)\phi(1,b,1)\tau(1) + \iota(1)\phi(1,b,2)\tau(2) \\ &+ \iota(2)\phi(2,b,1)\tau(1) + \iota(2)\phi(2,b,2)\tau(2) \\ &= 0.07 \end{aligned}$$

Theorem 1. *A semi-PA defines a* ***semi-distribution*** *over* Σ^**:*
$$P_A(\Sigma^*) = \sum_{u \in \Sigma^*} P_A(u) \leq 1$$

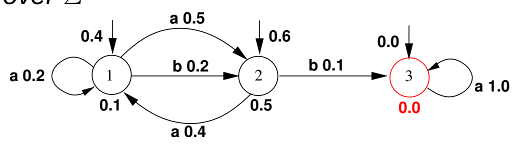
Probabilistic automaton

A state q is ***accessible*** if there is a strictly positive probability of reaching q from an initial state
$$\phi(Q_I, \Sigma^*, q) > 0$$

A semi-PA A is a ***probabilistic automaton*** (PA) if for any accessible state q there is a strictly positive probability of reaching a final state

$$P_{A_q}(\Sigma^*) = \sum_{q'} \phi(q, \Sigma^*, q')\tau(q') > 0.$$

Theorem 2. *Let A be a semi-PA, A is a* ***probabilistic automaton*** *if and only if P_A is a distribution over Σ^**

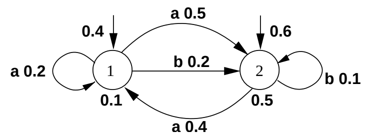


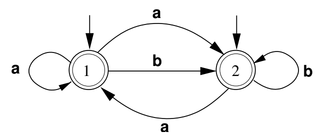
A non-probabilistic semi-PA

Support automaton, PNFA, PDFA

The ***support automaton*** of a PA $A = \langle \Sigma, Q, \phi, \iota, \tau \rangle$ is a non-deterministic finite automaton (NFA) $\underline{A} = \langle \Sigma, Q, \delta, I, F \rangle$ where

- I the set of ***initial states***
- F the set of ***final states***
- $\delta \subseteq Q \times \Sigma \times Q$ the transition function: $(q, a, q') \in \delta \Leftrightarrow \phi(q, a, q') > 0$





Property 1. *The language L generated by the support automaton of a PA A is the support of the distribution P_A*

A ***PNFA*** (respectively PDFA) is a PA the support of which is a non-deterministic finite automaton (NFA) (respectively a DFA)

Probabilistic regular languages

A ***probabilistic language*** is a distribution ψ over Σ^*

A probabilistic language is ***regular*** if it can be generated by a PNFA or, equivalently, by a probabilistic regular grammar

There exist probabilistic languages, with regular support languages, that are not regular:

$$L = \{a^*\} \text{ and the distribution } \psi(a^n) = \frac{1}{e.n!}, \forall n \geq 0$$

PDFA are strictly less general than PNFA

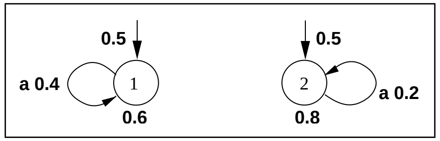
Theorem 3. $\mathcal{PDFA} \subsetneq \mathcal{PNFA}$

Proof (sketch):
Define $\rho(u)$

$$\forall u \in \Sigma^*, \rho(u) = \begin{cases} \frac{P_A(u)}{\overline{P}_A(u)} & \text{, if } \overline{P}_A(u) > 0 \\ 0 & \text{, otherwise.} \end{cases}$$

If A is a PDFA, the set $\{\rho(u), u \in \Sigma^*\}$ is necessarily finite

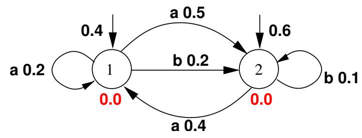
Consider the following PNFA:



$\rho(a^n) = 0.6 + \frac{0.2}{1+2^n}$ is a strictly decreasing series for strictly increasing values of n

$$\Rightarrow \{\rho(u), u \in \Sigma^*\} \text{ cannot be finite} \qquad \square$$

PNFA with no final probabilities

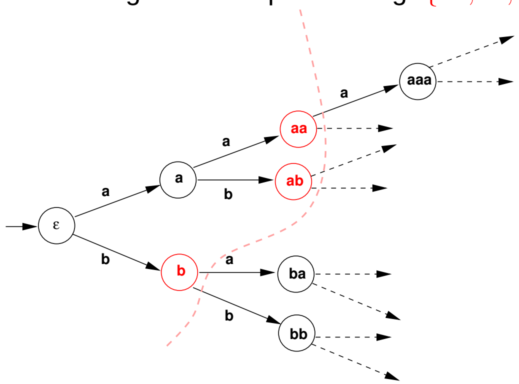


$\forall q \in Q, \tau(q) = 0$

- $\forall u \in \Sigma^*, P_A(u) = 0$
- such a machine defines probabilities on space of ***infinite*** words Σ^∞
- $\overline{P}_A(u) = \sum_{q,q' \in Q} \iota(q)\phi(q, u, q')$ can be interpreted as the ***probability of generating a finite prefix*** u of an infinite word
- a PNFA with no final probabilities defines a distribution over any ***complete finite prefix-free set***

Complete finite prefix-free sets

A ***complete finite prefix-free set*** can be represented as a ≪***cut***≫ in a infinite prefix tree of all possible strings on the alphabet: e.g. $\{aa, ab, b\}$



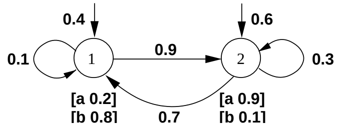
A PNFA with no final probabilities generates a family of distributions, one distribution for each complete finite prefix-free set

A particular case of interest: Σ^n, for any $n \in \mathbb{N}$

Hidden Markov Models

A discrete *Hidden Markov Model (HMM)* (with state emission) $M = \langle \Sigma, Q, A, B, \iota \rangle$

- Σ is a finite ***alphabet***
- Q is a ***set of states***
- $A : Q \times Q \rightarrow [0, 1]$ ***transition probability*** $\quad \forall q \in Q, \sum_{q' \in Q} A(q, q') = 1$
- $B : Q \times \Sigma \rightarrow [0, 1]$ ***state emission probability*** $\quad \forall q \in Q, \sum_{a \in \Sigma} B(q, a) = 1$
- $\iota : Q \rightarrow [0, 1]$ ***initial probability*** $\quad \sum_{q \in Q} \iota(q) = 1$



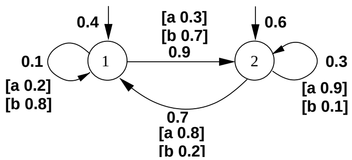
Hidden Markov Models with Emissions on Transitions

A discrete *Hidden Markov Model with transition emission (HMMT)* $M = \langle \Sigma, Q, A, B, \iota \rangle$

- Σ is a finite ***alphabet***
- Q is a ***set of states***
- $A : Q \times Q \rightarrow [0, 1]$ ***transition probability*** $\quad \forall q \in Q, \sum_{q' \in Q} A(q, q') = 1$
- $B : Q \times \Sigma \times Q \rightarrow [0, 1]$ ***transition emission probability***

$$\forall q, q' \in Q, \sum_{a \in \Sigma} B(q, a, q') = \begin{cases} 1 \text{ if } A(q, q') > 0 \\ 0 \text{ otherwise.} \end{cases}$$

- $\iota : Q \rightarrow [0, 1]$ ***initial probability*** $\quad \sum_{q \in Q} \iota(q) = 1$



Links between PA and HMMs

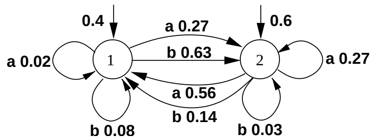
Theorem 4. *HMMs are equivalent to probabilistic automata with no final probabilities*

Constructive proof: PNFA $\Rightarrow$ HMMT $\Rightarrow$ HMM $\Rightarrow$ PNFA

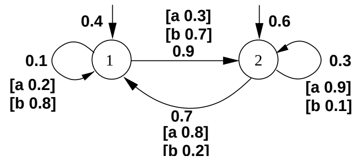
Corollary 1.
A HMM can be transformed into an equivalent PNFA with the ***same number of states***

A PNFA can be transformed into an equivalent HMM but ***generally not*** *with the* ***same number of states***

Transformation of a PNFA into an equivalent HMMT



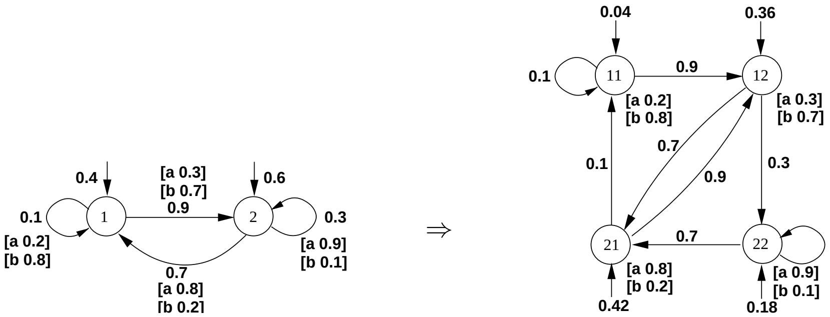
$\Rightarrow$



$$A(q, q') = \sum_{a \in \Sigma} \phi(q, a, q')$$

$$B(q, a, q') = \begin{cases} \frac{\phi(q,a,q')}{\sum_{a \in \Sigma} \phi(q,a,q')} & \text{if } \sum_{a \in \Sigma} \phi(q, a, q') > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Transformation of a HMMT into an equivalent HMM (1)



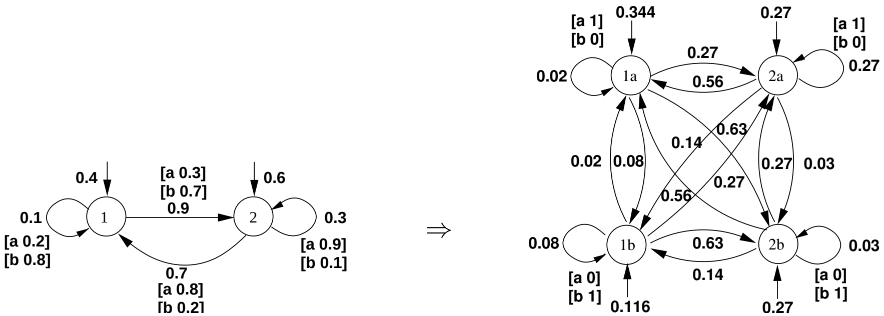
$Q' = \{(q, q') \in Q \times Q | A(q, q') > 0\}$. The states of Q' represents pairs of states in Q that are connected by a strictly positive transition probability ($\Rightarrow$ $|Q'| = \mathcal{O}(|Q^2|)$)

$$A((q,q'),(q'',q''')) = \begin{cases} A(q'',q''') & \text{if } q' = q'' \\ 0 & \text{otherwise.} \end{cases}$$

$B((q,q'),a) = B(q,a,q')$

$\iota'((q,q')) = \iota(q)A(q,q')$

Transformation of a HMMT into an equivalent HMM (2)



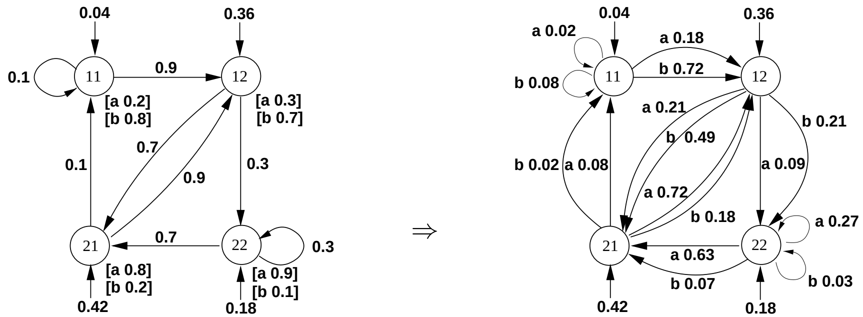
$Q' = Q \times \Sigma \Rightarrow |Q'| = \mathcal{O}(|Q| \times |\Sigma|)$

$\iota'((q,a)) = \sum_{q' \in Q} \iota(q')A(q',q)B(q',a,q)$

$B'((q,a),x) = 1$ if $x = a$, and 0 otherwise

$A'((q,a),(q',b)) = A(q,q')B(q,b,q')$

Transformation of a HMM into an equivalent PNFA



$\phi(q,a,q') = B(q,a)A(q,q')$

$\forall q, \tau(q) = 0$

Degrees of freedom

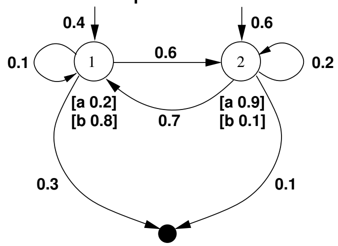
Consider machines (without final probabilities) with n states and an alphabet of m letters

Model	Parameters	Degrees of freedom	Total
PNFA	$\iota(q)$	$n-1$	
	$\phi(q,a,q')$	n^2m-n	
		n^2m-1	$\mathcal{O}(n^2 \times m)$
HMMT	$\iota(q)$	$n-1$	
	$A(q,q')$	n^2-n	
	$B(q,a,q')$	n^2m-n^2	
		n^2m-1	$\mathcal{O}(n^2 \times m)$
HMM	$\iota(q)$	$n-1$	
	$A(q,q')$	n^2-n	
	$B(q,a)$	$nm-n$	
		$n^2+nm-n-1$	$\mathcal{O}(n \times \max(n,m))$

A HMM can be transformed into an equivalent PNFA with the ***same number of states***, but the converse is not true in general.

PNFA are equivalent to HMMs with final probabilities

A HMM including final probabilities represented with a ***final silent state***



Theorem 5. *HMMs with final probabilities are equivalent to semi-PA*

Corollary 2. *HMMs with final probabilities, and such that the probabilities of reaching a final state from any accessible state is strictly positive, generate distributions over* Σ^*

- Probabilistic automata
  - Sufficient and necessary conditions to define a distribution
  - PDFA are strictly less general than PNFA
  - PNFA without final probabilities
- HMMs
  - HMM with state emission
  - HMM with transition emission (HMMT)
- Links between PA and HMMs
- **Learnability results**
- Open questions

Learning models

Learning a PNFA or a HMM aims at inducing a machine generating a distribution $\hat{P}$ from a ***sample*** S drawn according to some ***unknown target distribution*** P

A ***learning model*** includes a learning protocol specifying:

- the ***prior knowledge*** given to the learner
- the required quality of the learned hypothesis $\hat{P}$ ($\Rightarrow$ ***performance criterion***)
- some possible constraints on the sample S
- some possible ***bounds*** on the ***computational complexity*** of learning

Once a learning model is defined, one can ask

- whether a specific class of distributions can be learned?
- how much data is needed to reach a certain quality?
- what is the complexity of learning?

PAC learning model for distribution learning

Probably ***A***pproximately ***C***orrect learning

- Assume the data is an independent and identically distributed ***(iid) sample*** from P
- Consider a distance measure $D(P, \hat{P})$ between distributions P and $\hat{P}$
  An hypothesis is ***ϵ-good*** if $D(P, \hat{P}) \leq \epsilon$
- Given a *precision parameter* $\epsilon > 0$ and a confidence parameter $0 < \delta < 1$, the learning algorithm must output, with probability $1 - \delta$, an ϵ-good hypothesis $\hat{P}$
- the ***time complexity*** must be a ***polynomial*** function of $\frac{1}{\epsilon}, \frac{1}{\delta}$ and $|P|$

Notes:

- $|P|$ typically denotes the *number of parameters* to define the distributions (see degree of freedoms)
- A typical ≪distance≫ is the *Kullback-Leibler divergence* between P and $\hat{P}$
- Possible prior knowledges: P can be generated by a HMM, some constraints on the structure

(Simplified) learnability results

PAC learnability:

- Distributions defined by PDFA over an alphabet of 2 letters are ***not*** efficiently PAC ***learnable***
- Specific ***subclasses*** of PDFA are learnable
  - μ-distinguishable acyclic PDFA are ***learnable*** when μ is known
  - Probabilistic finite suffix automata of order L, equivalent to variable order Markov chains, are ***learnable*** when L is known

When the topology is assumed to be known, the learning problem is reduced to the problem of ***training*** a fixed set of parameters. ***Polynomial trainability*** requires to be able to approximate a model maximizing the sample likelihood in polynomial time:

- PDFA are polynomially ***trainable***
- 2-states PNFA are ***not*** polynomially ***trainable***
- EM algorithm outputs a ***locally optimal*** ML solution

(Simplified) learnability results (contd.)

- PNFA are ***identifiable in the limit*** with probability 1
  but this learning model requires an asymptotic identification of the structure without bounding the total complexity of learning

- Several practical induction algorithms do not fit in a learning model but a ***Bayesian learning*** framework.
  The goal is to build a model $\hat{M}$ maximizing the product of the prior probability $P(M)$ and the sample likelihood $P(S|M)$

Summary

- PNFA with no final probabilities are ***equivalent*** to HMMs
  They define distributions over complete finite prefix-free sets
- HMMs with final probabilities are ***equivalent*** to PNFA
  They define (semi-)distributions over Σ^*
- HMMs can be converted into PNFA and conversely, but not necessarily with the same number of states
- General HMMs (equivalent to PNFA) are ***hard to learn***
- PDFA form a ***restricted class***, ***hard*** to *learn* but ***easy*** to *train*
- Most practical algorithms induce PDFA, often in a Bayesian framework

Open questions

- New interesting subclasses efficiently learnable or polynomially trainable? Subclasses of PNFA, left-to-right HMMs, *etc*?
- Most negative PAC learnability results consider automata with no final probabilities. Can we come up with positive results for learning distributions over Σ^*?
- Relaxation of the PAC framework? Distance measure different from divergence but non trivial learning?
- Characterization of the local optimum produced by the EM algorithm in some cases?
- New robust and fast learning algorithms?
- Links with the learning of probabilistic acceptors defining conditional distributions $P(Y = y|u)$ with $u \in \Sigma^*$?

Additional information

- proofs
- more details on learnability results
- a presentation of several PA/HMM induction algorithms
- many references

P. Dupont, F. Denis and Y. Esposito, *Links between Probabilistic Automata and Hidden Markov Models: probability distributions, learning models and induction algorithms*, to appear in Pattern Recognition: Special Issue on Grammatical Inference Techniques & Applications, 2004.

See http://www.info.ucl.ac.be/∼pdupont/