Affine finite automata

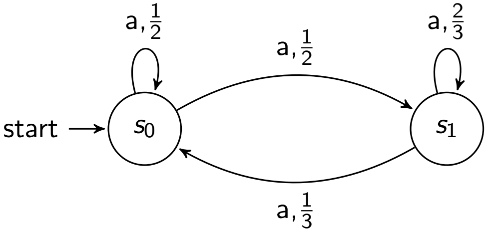
A quantum-like classical finite automata

Abuzer Yakaryılmaz
abuzer.yakaryilmaz@gmail.com

October 16, 2016
Theory Days at Lilaste, Latvia

Joint work with
Alejandro Díaz-Caro, Universidad Nacional de Quilmes (Argentina)
and
Marcos Villagra, Universidad Nacional de Asuncion (Paraguay)

A probabilistic finite automaton (PFA) is a generalization of deterministic finite automaton (DFA) that can make random choices:



Framework for probabilistic systems.

- A probabilistic state is defined on $(\mathbb{R}^+ \cup \{0\})^n$ for some $n > 0$.
- The l_1 norm of a probabilistic state is 1 and the probability of observing a state is its contribution in the l_1 norm, which is simply the value in the corresponding entry.
- The summation of probabilities is always 1.
- They evolve linearly (i.e. stochastic matrices) and l_1-norm is preserved on nonnegative vectors.

A probabilistic state v:

$$v = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}, \quad 0 \leq p_i \leq 1, \quad |v| = \sum_{i=1}^{n} p_i = 1.$$

Each column of a stochastic matrix (A) is a probabilistic state.

$$v' = Av \to \begin{pmatrix} p'_1 \\ \vdots \\ p'_n \end{pmatrix} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1,1} & p_{1,2} & \cdots & p_{1,n} \end{pmatrix} \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}, \quad |v'| = 1.$$

The (j, i)-th entry of A, $p_{j,i}$, represents the probability of going from the i-th state to j-th state.

Framework for **a more general** probabilistic systems.

- A general probabilistic state is defined on $(\mathbb{R})^n$ for some $n > 0$.
- The l_1 norm of a probabilistic state is 1 and the probability of observing a state is its contribution in the l_1 norm, which is the absolute value of the corresponding entry.
- The summation of probabilities is always 1.
- They evolve linearly (i.e. YYY matrices) and l_1-norm is preserved.

YYY?

New framework based on l_2-norm:

- The summation of probabilities is always 1.
- The l_2 norm of a new kind state is 1 and the probability of observing a state is its contribution in the l_2 norm, i.e. the square of the corresponding entry.
- A state is defined on $\mathbb{R}^n$ for some $n > 0$.

New framework based on l_2-norm:

- The summation of probabilities is always 1.
- The l_2 norm of a new kind state is 1 and the probability of observing a state is its contribution in the l_2 norm, i.e. the square of the corresponding entry.
- A state is defined on $\mathbb{R}^n$ for some $n > 0$.
- They evolve linearly (i.e. ZZZ matrices) and l_2-norm is preserved.

ZZZ?

An n-dimensional system can have the following state:

$$v = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \in \mathbb{R}^n, \quad |v| = \sum_{i=1}^{n} |\alpha_n|^2 = 1,$$

where the probability of observing the i-th state is $|\alpha_i|^2$.

The column of a orthogonal matrix (O) is also a norm-1 vector.

$$v' = Ov \rightarrow \begin{pmatrix} \alpha'_1 \\ \vdots \\ \alpha'_n \end{pmatrix} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,n} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,n} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}, \ |v'| = 1.$$

The (j, i)-th entry of O, $\alpha_{j,i}$, represents the transition value of going from the i-th state to j-th state.

New **updated** framework based on l_2-norm:

- A state is defined on $\mathbb{C}^n$ for some $n > 0$.
- The l_2 norm of a new kind state is 1 and the probability of observing a state is its contribution in the l_2 norm, which is square of the value in the corresponding entry.
- The summation of probabilities is always 1.
- They evolve linearly (i.e. unitary matrices) and l_2-norm is preserved.

How can we defined a quantum-like (using negative values) system classically?

- The state should be a vector in $\mathbb{R}^n$.
- But there is no linear operator preserving l_1-norm.
- On the other hand, another property of stochastic vectors is that the summation of all entries is 1.
- Is there any such linear operator?

How can we defined a quantum-like (using negative values) system classically?

- The state should be a vector in $\mathbb{R}^n$.
- But there is no linear operator preserving l_1-norm.
- On the other hand, another property of stochastic vectors is that the summation of all entries is 1.
- Is there any such linear operator?

Yes, affine operators, preserving the summation!

An affine state v:

$$v = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, \quad a_i \in \mathbb{R}, \quad \sum_{i=1}^{n} a_i = 1.$$

Each column of an affine matrix (A) is an affine state.

$$v' = Av \rightarrow \begin{pmatrix} a'_1 \\ \vdots \\ a'_n \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,1} & a_{1,2} & \cdots & a_{1,n} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, \sum_{i=1}^{n} a_i = 1.$$

The (j, i)-th entry of A, $a_{j,i}$, represents the transition value of going from the i-th state to j-th state.

How can we determine the observing probability of i-th state?

$$v = \begin{pmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{pmatrix}, \quad \sum_{i=1}^{n} a_i = 1.$$

Remark that $|v| \geq 1$!

How can we determine the observing probability of i-th state?

$$v = \begin{pmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{pmatrix}, \quad \sum_{i=1}^{n} a_i = 1.$$

Remark that $|v| \geq 1$!

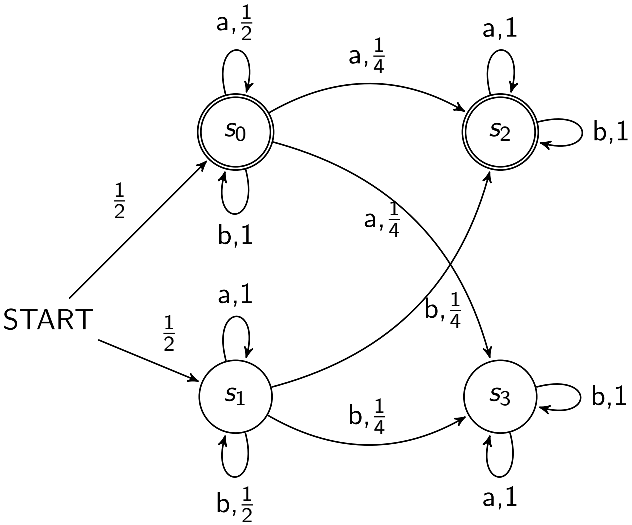
We use a non-linear operator called weighting that returns the weight of each state in $|v|$.

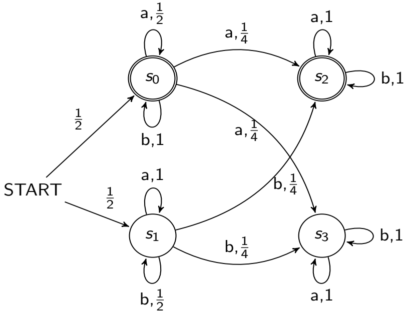
$$Pr[a_i] = \frac{|a_i|}{|v|}.$$

Framework for affine systems.

- An affine state is defined on $\mathbb{R}^n$ for some $n > 0$.
- The cumulative sum is 1 and the probability of observing a state is its contribution in the l_1 norm, i.e. the normalized absolute value of the corresponding entry.
- The summation of probabilities is always 1.
- They evolve linearly (i.e. affine matrices) and cumulative sum is preserved but l_1-norm does not to be preserved.

Consider a PFA example: 4-state PFA P defined over $\{a, b\}$:





After reading $a^m b^n$, the probabilities:

$$s_0 : p_0 = \left(\frac{1}{2}\right)^{m+1} \qquad s_1 : p_1 = \left(\frac{1}{2}\right)^{n+1}$$

$$s_2 : p_2 = (1 - p_0 - p_1)/2 \qquad s_3 : p_3 = (1 - p_0 - p_1)/2$$

The accepting and rejecting probabilities are

$$f_P(a^m b^n) = \left(\frac{1}{2}\right)^{m+1} + \frac{1 - p_0 - p1}{2}$$

and

$$1 - f_P(a^m b^n) = \left(\frac{1}{2}\right)^{n+1} + \frac{1 - p_0 - p1}{2}.$$

The accepting and rejecting probabilities are

$$f_P(a^m b^n) = \left(\frac{1}{2}\right)^{m+1} + \frac{1 - p_0 - p1}{2}$$

and

$$1 - f_P(a^m b^n) = \left(\frac{1}{2}\right)^{n+1} + \frac{1 - p_0 - p1}{2}.$$

How can we define a language recognized by P?

Remark that the automaton P defines a probability distributions over all strings.

The accepting and rejecting probabilities are

$$f_P(a^m b^n) = \left(\frac{1}{2}\right)^{m+1} + \frac{1 - p_0 - p1}{2}, \text{ and}$$

$$1 - f_P(a^m b^n) = \left(\frac{1}{2}\right)^{n+1} + \frac{1 - p_0 - p1}{2}.$$

We can pick a threshold called cutpoint $\lambda \in [0, 1)$ and then classify all strings under three sets. Let's pick $\lambda = \frac{1}{2}$:

- $L(P, < \frac{1}{2}) = \{w \mid f_P(w) < \frac{1}{2}\}$, formed by the string accepted with probability less than $\frac{1}{2}$
- $L(P, = \frac{1}{2}) = \{w \mid f_P(w) = \frac{1}{2}\}$, formed by the string accepted with probability equal to $\frac{1}{2}$
- $L(P, > \frac{1}{2}) = \{w \mid f_P(w) > \frac{1}{2}\}$, formed by the string accepted with probability greater than $\frac{1}{2}$

Any of them or any two of them form a language recognized by P.

Any language recognized by a PFA P with cutpoint λ is called stochastic:

$$L(P, > \lambda) = \{w \mid f_P(w) > \lambda\}.$$

The class of stochastic languages is denoted SL.

Any language defined in the following way is called exclusive stochastic languages:

$$L(P, \neq \lambda) = \{w \mid f_P(w) \neq \lambda\}.$$

The class of exclusive stochastic languages is denoted $\mathsf{SL}^{\neq}$, a proper superset of regular languages (REG).

The complement class of $\mathsf{SL}^{\neq}$ is $\mathsf{SL}^{=}$:

- $\mathtt{EQ} = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ is in $\mathsf{SL}^{=}$ and
- $\mathtt{NEQ} = \{w \in \{a,b\}^* \mid |w|_a \neq |w|_b\}$ is in $\mathsf{SL}^{\neq}$.

QFAs with cutpoints define exactly the same class as PFAs: SL.

Exclusive quantum languages are identical to stochastic one $\mathrm{SL}^{\neq}$. However QFAs and PFAs can differ in the following case:

- The class of languages $L(P, \neq 0) = \{w | f_P(w) > 0\}$, where P is a PFA.
- The class of languages $L(M, \neq 0) = \{w | f_P(w) > 0\}$, where M is a QFA.

The class of languages $L(P, \neq 0) = L(P, > \lambda) = \{w | f_P(w) > 0\}$, where P is a PFA.

A PFA, say P, example for the following language:

$$\texttt{MOD}_{3,5,7} = \{a^i \mid i \mod 3 \equiv 0 \text{ or } i \mod 5 \equiv 0 \text{ or } i \mod 7 \equiv 0\}$$

With equal probability split into 3 paths at the beginning and then make each modular check separately. If one check is successful, then accept the input.

- P accepts all members with probability at least $\frac{1}{3}$.
- P accepts each non-member with zero probability.

The class of languages $L(P, \neq 0) = L(P, > \lambda) = \{w | f_P(w) > 0\}$, where P is a PFA.

A PFA, say P, example for the following language:

$$\texttt{MOD}_{3,5,7} = \{a^i \mid i \mod 3 \equiv 0 \text{ or } i \mod 5 \equiv 0 \text{ or } i \mod 7 \equiv 0\}$$

With equal probability split into 3 paths at the beginning and then make each modular check separately. If one check is successful, then accept the input.

- P accepts all members with probability at least $\frac{1}{3}$.
- P accepts each non-member with zero probability.

Any PFA P fixed to define a single language $L(P, > 0)$ is a nondeterministic finite automaton (NFA).

Any PFA P fixed to define a single language $L(P, > 0)$ is a nondeterministic finite automaton (NFA).

Any QFA M fixed to define a single language $L(M, > 0)$ is a nondeterministic QFA (NQFA).

Any PFA P fixed to define a single language $L(P, > 0)$ is a nondeterministic finite automaton (NFA).

Any QFA M fixed to define a single language $L(M, > 0)$ is a nondeterministic QFA (NQFA).

- NFAs define only REG.
- NQFAs define $\mathsf{SL}^{\neq}$.

Bounded-error computation:

- There is a constant gap between the accepting probabilities of members and non-members.
- The PFA algorithm for $\texttt{MOD}_{3,5,7}$, where the gap is $\frac{1}{3}$.
- So, this algorithm is also one-sided bounded-error. One answer is always correct!

Bounded-error PFAs and QFAs define exactly REG.

An n-state affine finite automaton (AfA) M is a 5-tuple

$$M = (E, \Sigma, \{A_\sigma \mid \sigma \in \Sigma \cup \{\#\}\}, v_0, E_a),$$

where

- $E = \{e_1, \ldots, e_n\}$ is the set of states,
- Σ is the input alphabet not containing the right end-marker $\#$,
- A_σ is the affine operator applied when reading symbol $\sigma \in \Sigma \cup \{\#\}$,
- v_0 is the initial affine state, and
- $E_a \subset E$ is the set of accepting state.

For a given input $w \in \Sigma^*$, the computation is traced as

$$v_f = A_\# A_{w_{|w|}} \cdots A_{w_1} v_0.$$

For a given input $w \in \Sigma^*$, the computation is traced as

$$v_f = A_{\#} A_{w_{|w|}} \cdots A_{w_1} v_0.$$

Then, the accepting probability of w by M is

$$f_M(w) = \frac{\sum_{e_i \in E_a} |v_f[i]|}{|v_f|}.$$

Bounded-error: Consider the nonregular language
$\mathtt{EQ} = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$:

- The initial affine state $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$.
- Apply $A_a = \begin{pmatrix} 2 & 0 \\ -1 & 1 \end{pmatrix}$ for each a.
- Apply $A_b = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 1 \end{pmatrix}$ for each b.

If $|w|_a = m$ and $|w|_b = n$, then the final state is

$$\begin{pmatrix} 2^{m-n} \\ 1 - 2^{m-n} \end{pmatrix}.$$

- Each member ($m = n$) is accepted with probability 1.
- Each non-member ($m \neq n$) is accepted with probability at most $\frac{2}{3}$.

Bounded-error AfAs are more powerful than bounded-error PFAs and QFAs.

Nondeterministic affine languages: NAfAs and NQFAs can simulate each other and so they are equivalent and more powerful than PFAs.

- When focusing on a single non-zero accepting path, the degree of norm is not important $(l_1, l_2, \ldots, l_i, \ldots, l_\infty)$.
- The restriction of PFAs is using only non-negative values.

Now we have:

$$\mathsf{NQAL} = \mathsf{SL}^{\neq} = \mathsf{NAfL} = \mathsf{NQAL}^{\neq}$$

Moreover, we can follow that

- The class of languages recognized by one-sided bounded-error (rational) AfAs are identical to

  $$\mathsf{SL}^{=}_{\mathbb{Q}} \cup \mathsf{SL}^{\neq}_{\mathbb{Q}},$$

  where the classes are defined with PFAs using only rational numbers.
- In other words, nondeterminism is useless for AfAs when restricted to rational numbers.

What can we say about exclusive affine language, i.e. $\mathsf{AfL}^{\neq}$ (and its complement class $\mathsf{AfL}^{=}$)?

The power of weighting operator:

The language ABS-EQ is defined on $\{a, b\}$ such that $w \in$ ABS-EQ if and only if

$$|m - n| + |m - 4n| = |m - 2n| + |m - 3n|, \tag{1}$$

where $|w|_a = m$ and $|w|_b = n$.

- Without absolute values, the equality is trivial.
- Interestingly, ABS-EQ $\notin \mathsf{SL}^=$.
- On the other hand, we can easily show that ABS-EQ $\in \mathsf{AfL}^=$.

The language ABS-EQ is defined on $\{a, b\}$ such that $w \in$ ABS-EQ if and only if

$$|m - n| + |m - 4n| = |m - 2n| + |m - 3n|, \tag{2}$$

where $|w|_a = m$ and $|w|_b = n$.

- We can encode the followings in the values of some states, $m, n, 2n, 3n, 4n$.
- Then, we can easily set the followings to the values of some states at the end of the computation:

$$\begin{pmatrix} m - n \\ m - 4n \\ m - 2n \\ m - 3n \\ \frac{1-T}{2} \\ \frac{1-T}{2} \end{pmatrix}, \text{ where } T \text{ is the summation of first entries.}$$

- By setting e_1, e_2, e_5 as the accepting states, we can get the desired result.

Remark that the computational power comes from weighting operator in the previous example!

$$\mathsf{SL}^{=} = \mathsf{QAL}^{=} \subsetneq \mathsf{AfL}^{=} \text{ and } \mathsf{SL}^{\neq} = \mathsf{QAL}^{\neq} \subsetneq \mathsf{AfL}^{\neq}$$

In classical case:

$$\mathsf{SL}^{\neq 0} = \mathsf{REG} \subsetneq \mathsf{SL}^{\neq}.$$

In quantum case:

$$\mathsf{QAL}^{\neq 0} = \mathsf{NQAL} = \mathsf{QAL}^{\neq}.$$

In affine case:

$$\mathsf{AfL}^{\neq 0} = \mathsf{NAfL} \subsetneq \mathsf{AfL}^{\neq}.$$

AfAs with cutpoints:

- They are more powerful than PFAs and QFAs since they can recognize some nonstochastic languages:

$$\texttt{LAPIN\v{S}}' = \{w \in \{a,b,c\}^* \mid |w|_a^4 > |w|_b^2 > |w|_c\}$$

or equivalently

$$\texttt{LAPIN\v{S}}' = \{w \in \{a,b,c\}^* \mid |w|_a^2 > |w|_b \text{ and } |w|_b^2 > |w|_c\}.$$