Maximum Likelihood Estimation of Factored Regular Deterministic Stochastic Languages

Chihiro Shibata and Jeffrey Heinz

University of Toronto
July 19, 2019

We thank JSPS KAKENHI #JP18K11449 (CS) and
NIH #R01HD87133-01 (JH)

The Problem in General

Stochastic languages are probability distributions over strings.

The Problem in General

Stochastic languages are probability distributions over strings.

$$f : \Sigma^* \to [0,1] \text{ and } \sum_{w} f(w) = 1$$

The Problem in General

Stochastic languages are probability distributions over strings.

$$f : \Sigma^* \to [0,1] \text{ and } \sum_{w} f(w) = 1$$

A class C of stochastic languages is often defined parametrically: an assignment of values to parameters Θ uniquely determines some stochastic language $f_\Theta \in C$.

THE PROBLEM IN GENERAL

Stochastic languages are probability distributions over strings.

$$f : \Sigma^* \rightarrow [0, 1] \text{ and } \sum_w f(w) = 1$$

A class C of stochastic languages is often defined parametrically: an assignment of values to parameters Θ uniquely determines some stochastic language $f_\Theta \in C$.

An important learning criterion

For *any* data sequence D drawn i.i.d. from *any* stochastic language, a Maximum-Likelihood Estimator finds parameter values $\hat{\Theta}$ which maximize $P(D)$ with respect to C.

The Problem in General

Stochastic languages are probability distributions over strings.

$$f : \Sigma^* \rightarrow [0, 1] \text{ and } \sum_{w} f(w) = 1$$

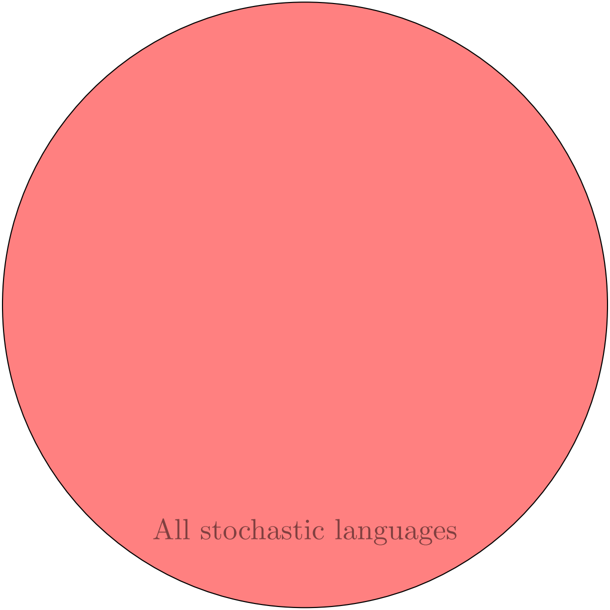
A class C of stochastic languages is often defined parametrically: an assignment of values to parameters Θ uniquely determines some stochastic language $f_\Theta \in C$.

An important learning criterion

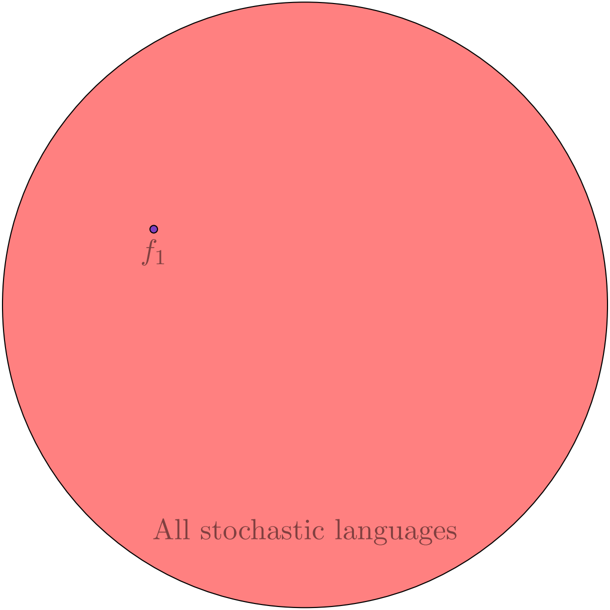
For *any* data sequence D drawn i.i.d. from *any* stochastic language, a Maximum-Likelihood Estimator finds parameter values $\hat{\Theta}$ which maximize $P(D)$ with respect to C.

For a class of stochastic languages C, is there an algorithm which reliably returns a Maximum-Likelihood Estimate (MLE) of an observed data sample D?

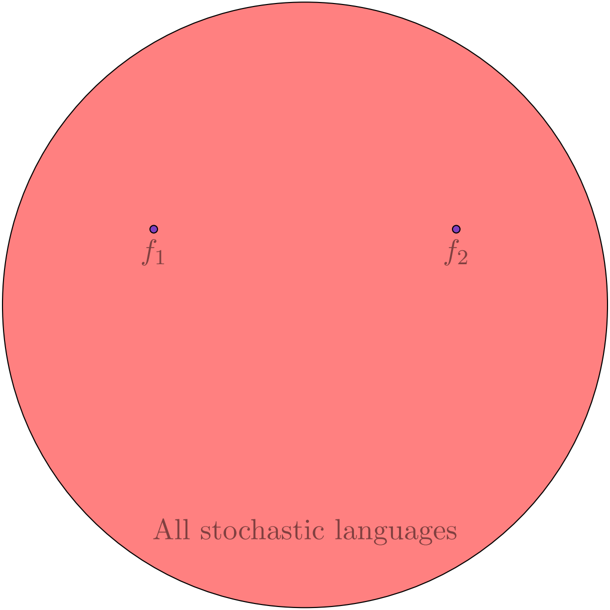
In Pictures



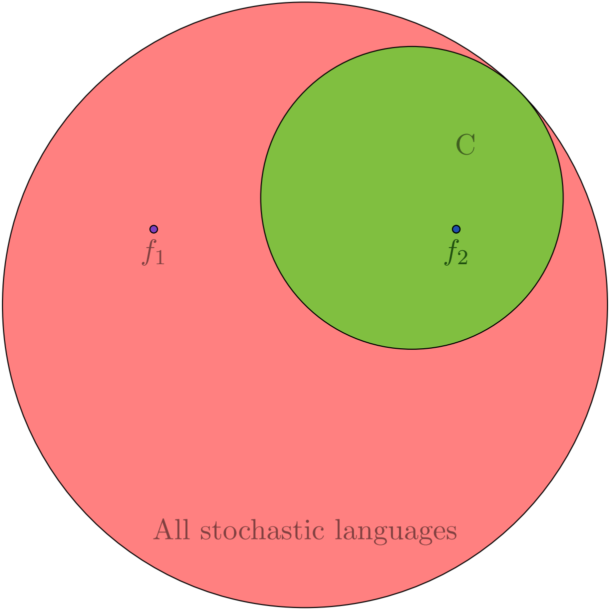
In Pictures



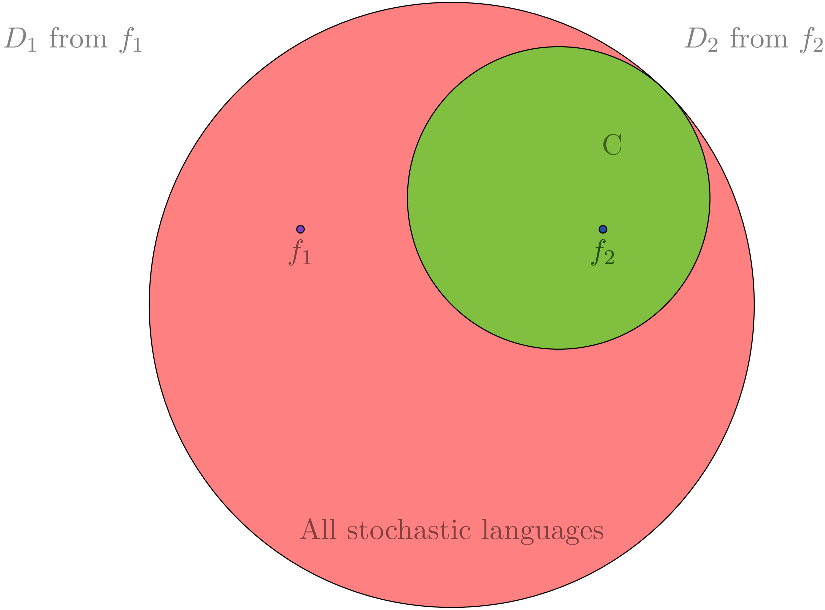
In Pictures



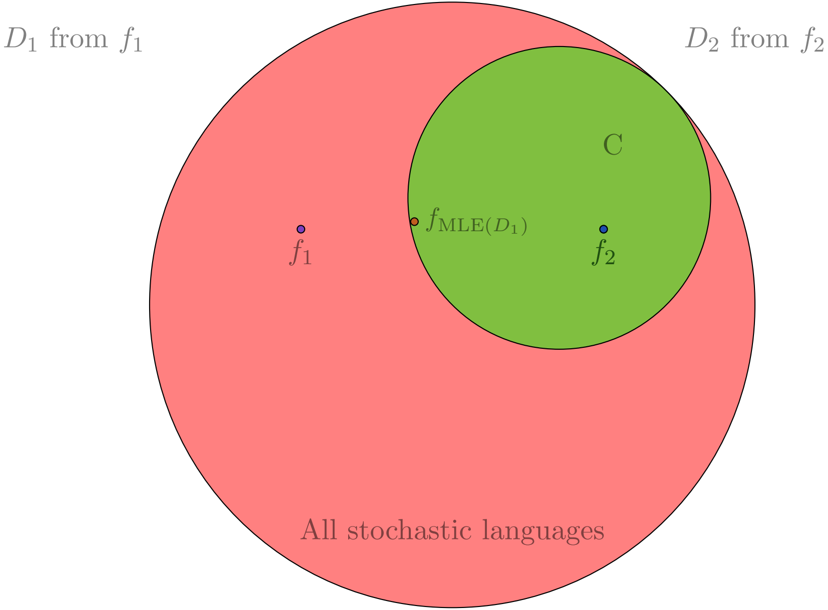
In Pictures



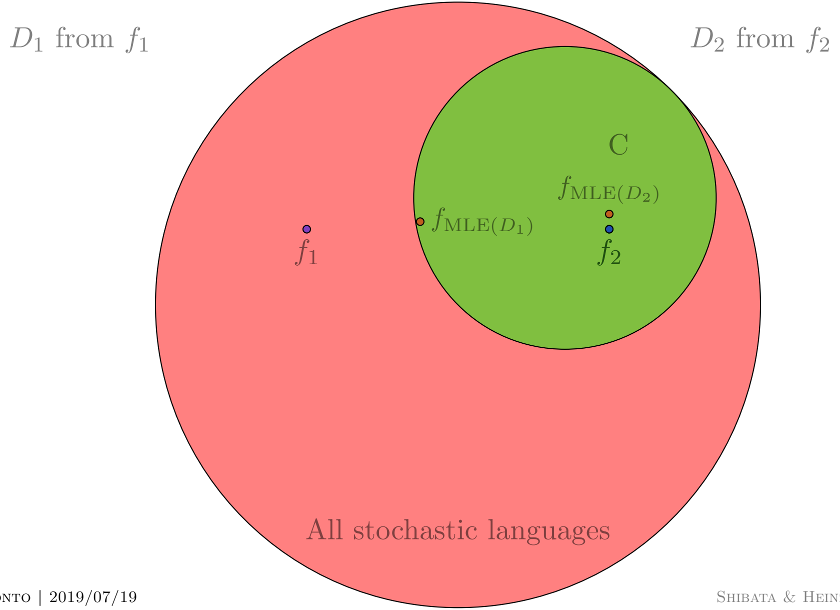
In Pictures



In Pictures


D_1 from f_1
D_2 from f_2
C
$f_{\text{MLE}(D_1)}$
f_1
f_2
All stochastic languages

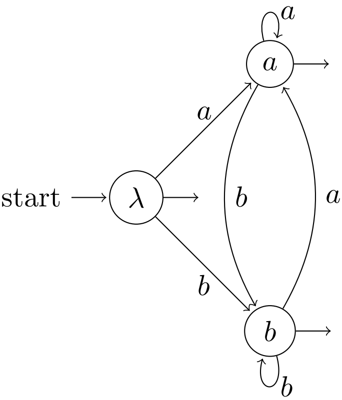
In Pictures



D_1 from f_1
D_2 from f_2
C
$f_{\mathrm{MLE}(D_2)}$
f_1
$f_{\mathrm{MLE}(D_1)}$
f_2
All stochastic languages

Classes defined by Single DFAs

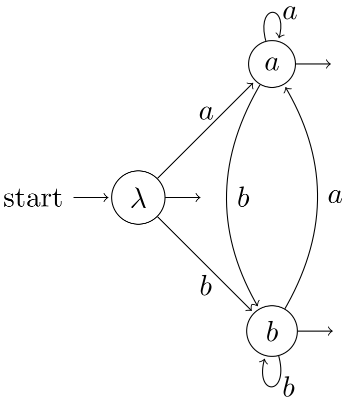
Example: Bigram model



Parameters
$\theta_{\rtimes a}$
$\theta_{\rtimes b}$
$\theta_{\rtimes \ltimes}$
θ_{aa}
θ_{ab}
$\theta_{a\ltimes}$
θ_{ba}
θ_{bb}
$\theta_{b\ltimes}$

Classes defined by Single DFAs

Example: Bigram model

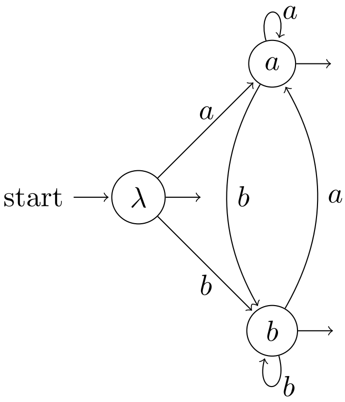


Parameters
$\theta_{\rtimes a}$
$\theta_{\rtimes b}$
$\theta_{\rtimes \ltimes}$
θ_{aa}
θ_{ab}
$\theta_{a\ltimes}$
θ_{ba}
θ_{bb}
$\theta_{b\ltimes}$

$$D = \langle ab, aabb \rangle$$

Classes defined by Single DFAs

Example: Bigram model

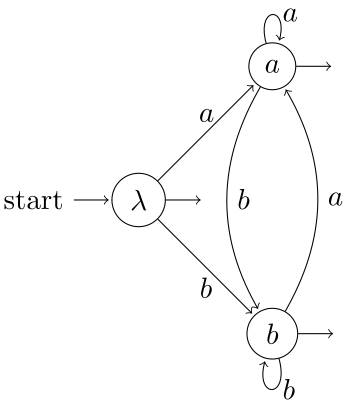


Parameters	
$\theta_{\rtimes a}$	1
$\theta_{\rtimes b}$	
$\theta_{\rtimes \ltimes}$	
θ_{aa}	
θ_{ab}	1
$\theta_{a\ltimes}$	
θ_{ba}	
θ_{bb}	
$\theta_{b\ltimes}$	1

$$D = \langle ab, aabb \rangle$$

MLE is obtained by passing D through DFA and normalizing.
(Vidal et al. 2005)

Example: Bigram model



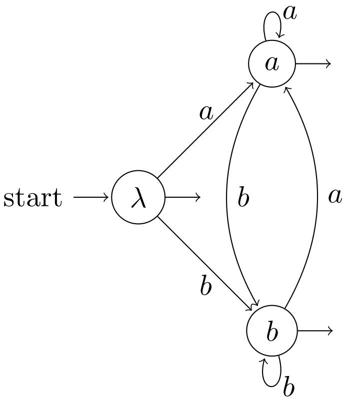
Parameters	
$\theta_{\rtimes a}$	2
$\theta_{\rtimes b}$	
$\theta_{\rtimes \ltimes}$	
θ_{aa}	1
θ_{ab}	2
$\theta_{a \ltimes}$	
θ_{ba}	
θ_{bb}	1
$\theta_{b \ltimes}$	2

$$D = \langle ab, \color{blue}{aabb}\rangle$$

MLE is obtained by passing D through DFA and normalizing.
(Vidal et al. 2005)

Classes defined by Single DFAs

Example: Bigram model



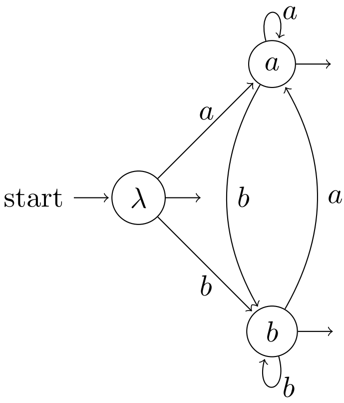
Parameters	
$\theta_{\rtimes a}$	1
$\theta_{\rtimes b}$	0
$\theta_{\rtimes \ltimes}$	0
θ_{aa}	1/3
θ_{ab}	2/3
$\theta_{a\ltimes}$	0
θ_{ba}	0
θ_{bb}	1/3
$\theta_{b\ltimes}$	2/3

$$D = \langle ab, aabb \rangle$$

MLE is obtained by passing D through DFA and normalizing.
(Vidal et al. 2005)

Classes defined by Single DFAs

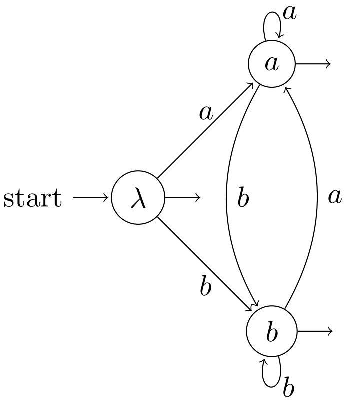
Example: Strictly 2-Local Languages



Parameters
$\theta_{\rtimes a}$
$\theta_{\rtimes b}$
$\theta_{\rtimes \ltimes}$
θ_{aa}
θ_{ab}
$\theta_{a\ltimes}$
θ_{ba}
θ_{bb}
$\theta_{b\ltimes}$

Classes defined by Single DFAs

Example: Strictly 2-Local Languages

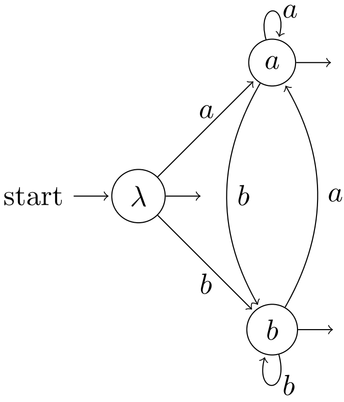


Parameters
$\theta_{\rtimes a}$
$\theta_{\rtimes b}$
$\theta_{\rtimes \ltimes}$
θ_{aa}
θ_{ab}
$\theta_{a \ltimes}$
θ_{ba}
θ_{bb}
$\theta_{b \ltimes}$

$$D = \langle ab, aabb \rangle$$

Classes defined by Single DFAs

Example: Strictly 2-Local Languages



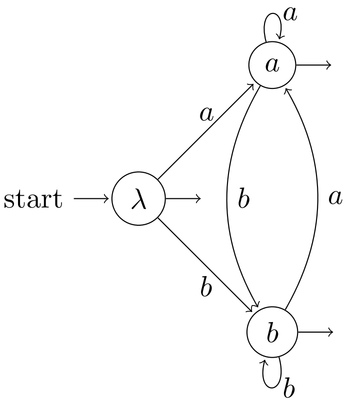
Parameters	
$\theta_{\rtimes a}$	1
$\theta_{\rtimes b}$	
$\theta_{\rtimes\ltimes}$	
θ_{aa}	
θ_{ab}	1
$\theta_{a\ltimes}$	
θ_{ba}	
θ_{bb}	
$\theta_{b\ltimes}$	1

$$D = \langle ab, aabb \rangle$$

Smallest language consistent with D in C is obtained by passing D through DFA and ‘activating’ parsed transitions. (Heinz and Rogers 2013)

Classes defined by Single DFAs

Example: Strictly 2-Local Languages



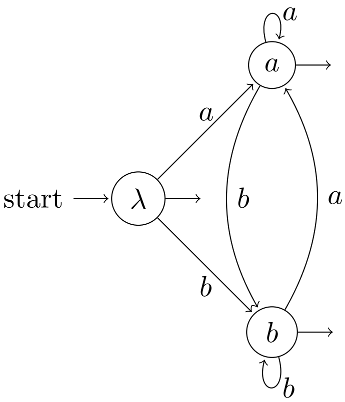
Parameters	
$\theta_{\rtimes a}$	1
$\theta_{\rtimes b}$	
$\theta_{\rtimes \ltimes}$	
θ_{aa}	1
θ_{ab}	1
$\theta_{a\ltimes}$	
θ_{ba}	
θ_{bb}	1
$\theta_{b\ltimes}$	1

$$D = \langle ab, \textcolor{blue}{aabb} \rangle$$

Smallest language consistent with D in C is obtained by passing D through DFA and 'activating' parsed transitions. (Heinz and Rogers 2013)

Classes defined by Single DFAs

Example: Strictly 2-Local Languages



Parameters	
$\theta_{\rtimes a}$	1
$\theta_{\rtimes b}$	0
$\theta_{\rtimes \ltimes}$	0
θ_{aa}	1
θ_{ab}	1
$\theta_{a\ltimes}$	0
θ_{ba}	0
θ_{bb}	1
$\theta_{b\ltimes}$	1

$$D = \langle ab, aabb \rangle$$

Smallest language consistent with D in C is obtained by passing D through DFA and 'activating' parsed transitions. (Heinz and Rogers 2013)

Overview of Related Results

			Class C defined with	
			single DFA	**finitely many DFA**
type of	$f : \Sigma^* \to \{0, 1\}$			
language	$f : \Sigma^* \to [\,0, 1\,]$			

1. For Boolean languages, the learning algorithms return the smallest language in C which includes D.
2. For Stochastic languages, the MLE returns the language in C which maximizes likelihood of D.

(Vidal et al. 2005, Heinz and Rogers 2013)

Overview of Related Results

		Class C defined with	
		single DFA	**finitely many DFA**
type of language	$f : \Sigma^* \to \{0, 1\}$	✓	
	$f : \Sigma^* \to [\,0, 1\,]$		

1. For Boolean languages, the learning algorithms return the smallest language in C which includes D.
2. For Stochastic languages, the MLE returns the language in C which maximizes likelihood of D.

(Vidal et al. 2005, Heinz and Rogers 2013)

Overview of Related Results

		Class C defined with	
		single DFA	finitely many DFA
type of	$f : \Sigma^* \to \{0, 1\}$	✓	
language	$f : \Sigma^* \to [\,0, 1\,]$	✓	

1. For Boolean languages, the learning algorithms return the smallest language in C which includes D.
2. For Stochastic languages, the MLE returns the language in C which maximizes likelihood of D.

(Vidal et al. 2005, Heinz and Rogers 2013)

Overview of Related Results

		Class C defined with	
		single DFA	**finitely many DFA**
type of	$f : \Sigma^* \to \{0, 1\}$	✓	✓
language	$f : \Sigma^* \to [\,0, 1\,]$	✓	

1. For Boolean languages, the learning algorithms return the smallest language in C which includes D.
2. For Stochastic languages, the MLE returns the language in C which maximizes likelihood of D.

(Vidal et al. 2005, Heinz and Rogers 2013)

Overview of Related Results

		Class C defined with	
		single DFA	**finitely many DFA**
type of	$f : \Sigma^* \to \{0, 1\}$	✓	✓
language	$f : \Sigma^* \to [0, 1]$	✓	this paper

1. For Boolean languages, the learning algorithms return the smallest language in C which includes D.
2. For Stochastic languages, the MLE returns the language in C which maximizes likelihood of D.

(Vidal et al. 2005, Heinz and Rogers 2013)

Overview of Related Results (part 2)

1. The class of all DFAs is not identifiable in the limit from positive data (Gold 1967).

Overview of Related Results (part 2)

1. The class of all DFAs is not identifiable in the limit from positive data (Gold 1967).
2. It is NP-hard to find the minimal DFA consistent with a finite sample of positive and negative examples (Gold 1978).

Overview of Related Results (part 2)

1. The class of all DFAs is not identifiable in the limit from positive data (Gold 1967).
2. It is NP-hard to find the minimal DFA consistent with a finite sample of positive and negative examples (Gold 1978).
3. Each DFA admits a characteristic sample D of positive and negative examples such that RPNI identifies the DFA from any superset of D in cubic time (Oncina and Garcia 1992, DuPont 1996).

Overview of Related Results (part 2)

1. The class of all DFAs is not identifiable in the limit from positive data (Gold 1967).
2. It is NP-hard to find the minimal DFA consistent with a finite sample of positive and negative examples (Gold 1978).
3. Each DFA admits a characteristic sample D of positive and negative examples such that RPNI identifies the DFA from any superset of D in cubic time (Oncina and Garcia 1992, DuPont 1996).
4. ALEGRIA/RLIPS (based on RPNI) (Carrasco and Oncina 1994, 1999) learns the class of PDFAs in polynomial time with probability one (de la Higuera and Thollard 2001).

Overview of Related Results (part 2)

1. The class of all DFAs is not identifiable in the limit from positive data (Gold 1967).
2. It is NP-hard to find the minimal DFA consistent with a finite sample of positive and negative examples (Gold 1978).
3. Each DFA admits a characteristic sample D of positive and negative examples such that RPNI identifies the DFA from any superset of D in cubic time (Oncina and Garcia 1992, DuPont 1996).
4. ALEGRIA/RLIPS (based on RPNI) (Carrasco and Oncina 1994, 1999) learns the class of PDFAs in polynomial time with probability one (de la Higuera and Thollard 2001).
5. Clark and Thollard (2004) present an algorithm which learns the class of PDFAs in a modified PAC setting. (See also Parekh and Hanover 2001.)

Overview of Related Results (part 2)

1. The class of all DFAs is not identifiable in the limit from positive data (Gold 1967).
2. It is NP-hard to find the minimal DFA consistent with a finite sample of positive and negative examples (Gold 1978).
3. Each DFA admits a characteristic sample D of positive and negative examples such that RPNI identifies the DFA from any superset of D in cubic time (Oncina and Garcia 1992, DuPont 1996).
4. ALEGRIA/RLIPS (based on RPNI) (Carrasco and Oncina 1994, 1999) learns the class of PDFAs in polynomial time with probability one (de la Higuera and Thollard 2001).
5. Clark and Thollard (2004) present an algorithm which learns the class of PDFAs in a modified PAC setting. (See also Parekh and Hanover 2001.)
6. Maximization-Expectation techniques are used to learn the class of PNFAs, but there is no guarantee to find a global optimum (Rabiner 1989).

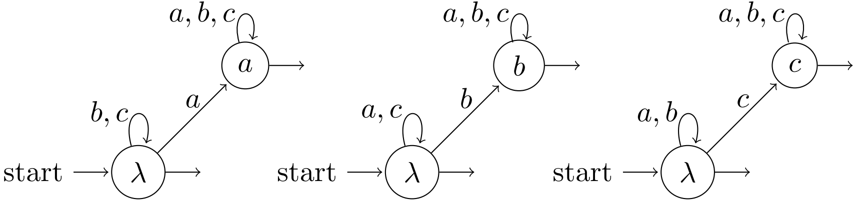
Defining C with finitely many DFA

Defining C with finitely many DFA

How do you define a class C with finitely many DFA?

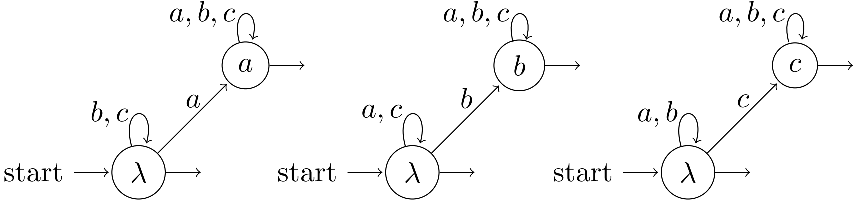
DEFINING C WITH FINITELY MANY DFA

How do you define a class C with finitely many DFA?



Defining C with finitely many DFA

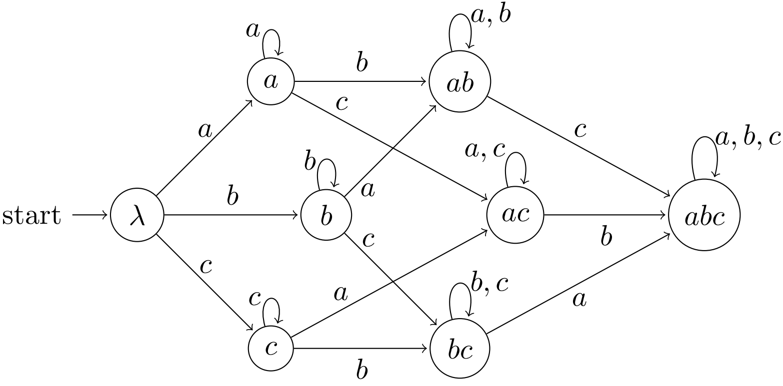
How do you define a class C with finitely many DFA?



Product Operations

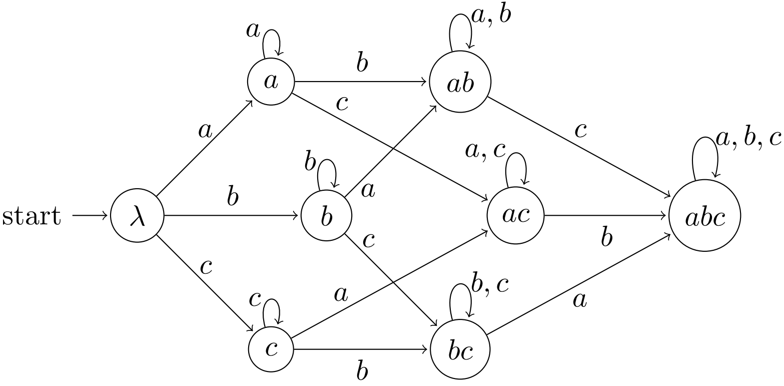
1. For Boolean languages, use **acceptor product** (yields intersection)
2. For Stochastic languages, use **co-emission product** (yields joint distribution)

The product of those three acceptors



(exit/accepting arrow at each state is not shown)

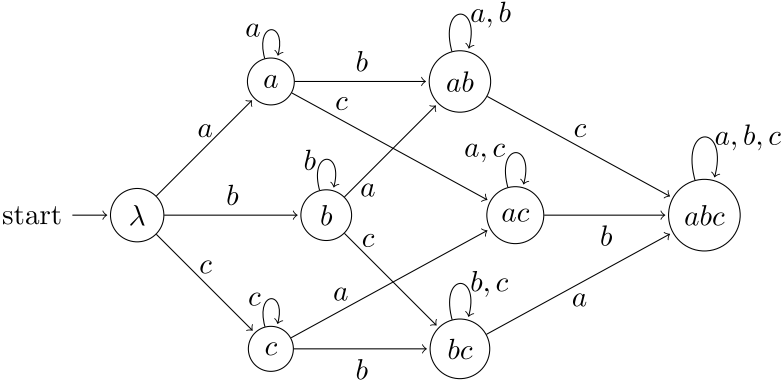
The product of those three acceptors



(exit/accepting arrow at each state is not shown)

- If C is defined by this DFA, then C = Piecewise 2-Testable.

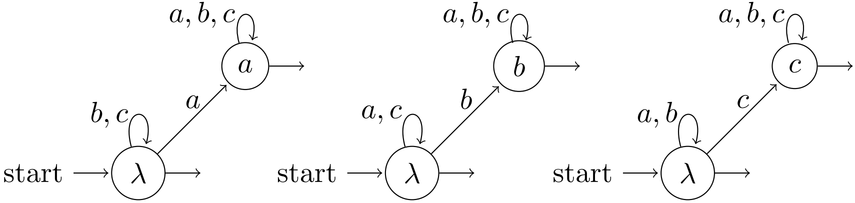
The product of those three acceptors



(exit/accepting arrow at each state is not shown)

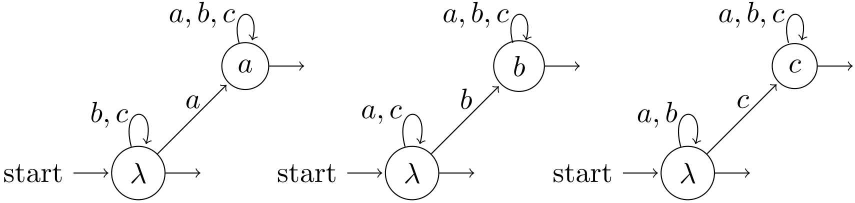
- If C is defined by this DFA, then C = Piecewise 2-Testable.
- If C is defined by the 3 atomic DFAs, then C = Strictly 2-Piecewise.

Cause . . .



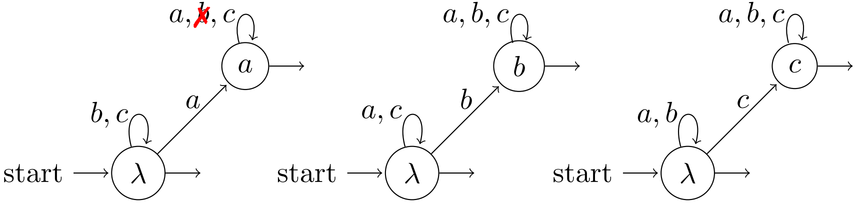
start
λ
b, c
a
a, b, c
a
start
λ
a, c
b
a, b, c
b
start
λ
a, b
c
a, b, c
c

CAUSE . . .



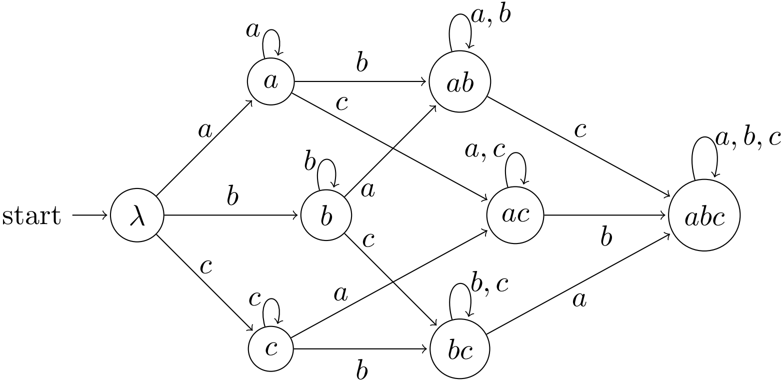
The parameters of the model are set at the level of the individual DFA.

Cause . . .



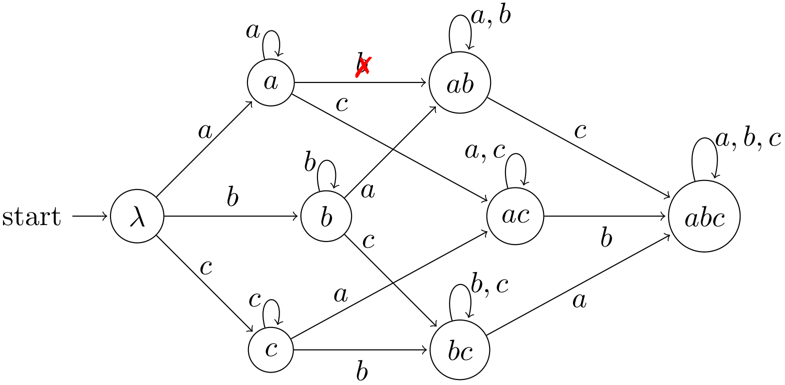
The parameters of the model are set at the level of the individual DFA.

. . . and Effect



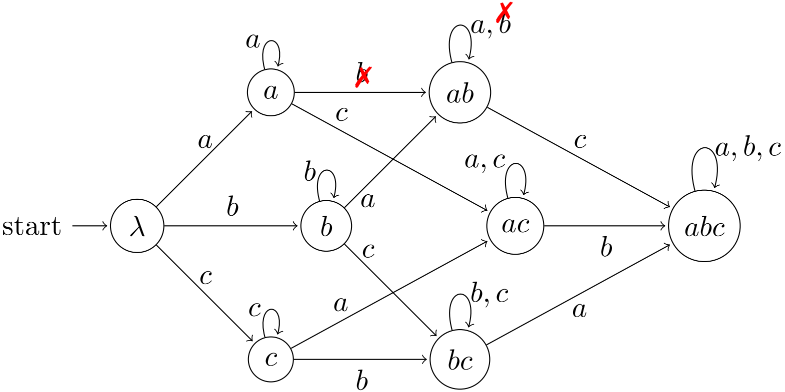
(exit/accepting arrow at each state is not shown)

. . . and Effect



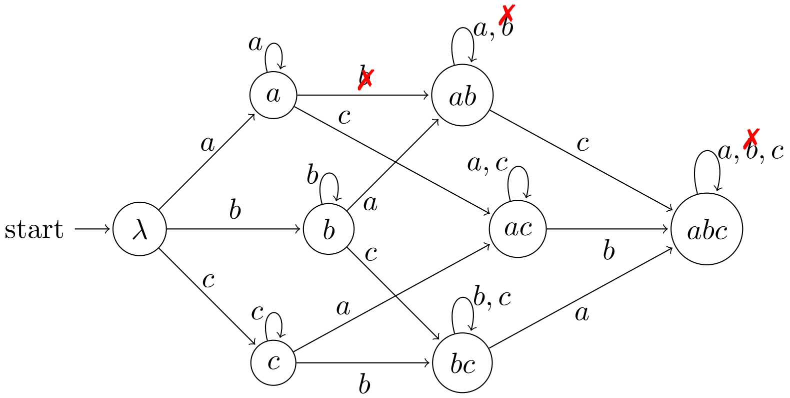
(exit/accepting arrow at each state is not shown)

. . . and Effect



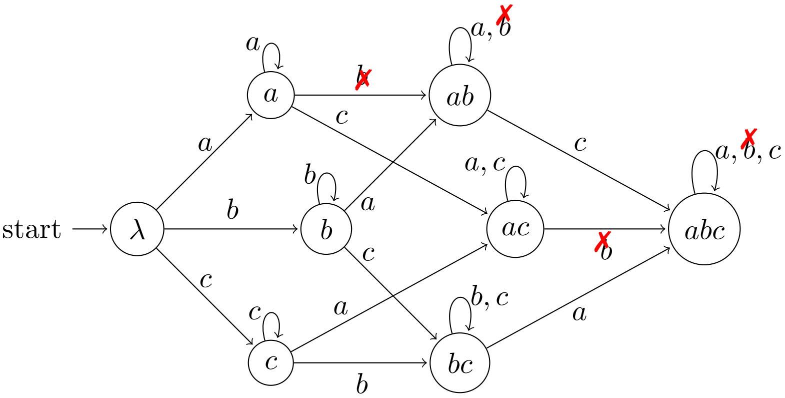
(exit/accepting arrow at each state is not shown)

. . . and Effect



(exit/accepting arrow at each state is not shown)

. . . and Effect



(exit/accepting arrow at each state is not shown)

Comparing the Representations

The Product DFA

The Atomic DFAs

Comparing the Representations

The Product DFA

1. In the worst case, it has $\prod_i |Q_i|$ states and $(|\Sigma| + 1) \prod_i |Q_i|$ parameters.

The Atomic DFAs

Comparing the Representations

The Product DFA

1. In the worst case, it has $\prod_i |Q_i|$ states and $(|\Sigma| + 1) \prod_i |Q_i|$ parameters.

The Atomic DFAs

1. The atomic DFAs have a total of $\sum_i |Q_i|$ states and $(|\Sigma| + 1) \sum_i |Q_i|$ parameters.

Comparing the Representations

The Product DFA

1. In the worst case, it has $\prod_i |Q_i|$ states and $(|\Sigma| + 1) \prod_i |Q_i|$ parameters.
2. Transitions/parameters are independent of others.

The Atomic DFAs

1. The atomic DFAs have a total of $\sum_i |Q_i|$ states and $(|\Sigma| + 1) \sum_i |Q_i|$ parameters.

Comparing the Representations

The Product DFA

1. In the worst case, it has $\prod_i |Q_i|$ states and $(|\Sigma| + 1) \prod_i |Q_i|$ parameters.
2. Transitions/parameters are independent of others.

The Atomic DFAs

1. The atomic DFAs have a total of $\sum_i |Q_i|$ states and $(|\Sigma| + 1) \sum_i |Q_i|$ parameters.
2. The transitions in the product are NOT independent.

Pluses and Minuses

Pluses and Minuses

+ Fewer parameters means more accurate estimation of model parameters with less data.

Pluses and Minuses

+ Fewer parameters means more accurate estimation of model parameters with less data.
− Fewer parameters means the model is less expressive.

Pluses and Minuses

+ Fewer parameters means more accurate estimation of model parameters with less data.
− Fewer parameters means the model is less expressive.

- Heinz and Rogers (2013, MoL) extend the method of ‘activating’ data-parsed transitions to learn classes of Boolean languages defined with single DFA to classes of Boolean languages defined with finitely many DFA.

Pluses and Minuses

+ Fewer parameters means more accurate estimation of model parameters with less data.
– Fewer parameters means the model is less expressive.

- Heinz and Rogers (2013, MoL) extend the method of 'activating' data-parsed transitions to learn classes of Boolean languages defined with single DFA to classes of Boolean languages defined with finitely many DFA.
- They show it always returns the smallest Boolean language in the class consistent with the data, and thus identifies the class in the limit from positive data.

The Co-emission Product

The Co-emission Product

- The co-emission product defines how PDFA-definable stochastic languages can be multiplied together to yield a well-defined stochastic language.

The Co-emission Product

- The co-emission product defines how PDFA-definable stochastic languages can be multiplied together to yield a well-defined stochastic language.
- Heinz and Rogers 2010 defined stochastic Strictly k-Piecewise languages using a *variant* of the co-emission product.

The Co-emission Product

- The co-emission product defines how PDFA-definable stochastic languages can be multiplied together to yield a well-defined stochastic language.
- Heinz and Rogers 2010 defined stochastic Strictly k-Piecewise languages using a *variant* of the co-emission product.
- They claimed they could find the MLE, but nobody seemed convinced.

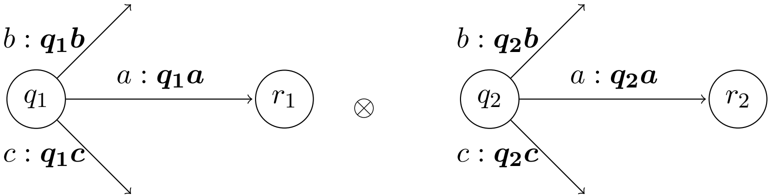
The Co-emission Product

- The co-emission product defines how PDFA-definable stochastic languages can be multiplied together to yield a well-defined stochastic language.
- Heinz and Rogers 2010 defined stochastic Strictly k-Piecewise languages using a *variant* of the co-emission product.
- They claimed they could find the MLE, but nobody seemed convinced.

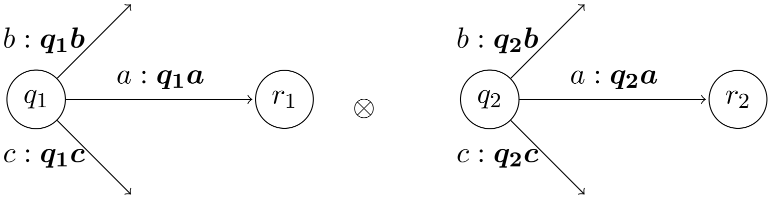
$Pr(x \mid P_{\leq 1}(y))$		x			
		s	$\widehat{\text{ts}}$	ʃ	$\widehat{\text{tʃ}}$
y	s	0.0335	0.0051	0.0011	0.0002
	$\widehat{\text{ts}}$	0.0218	0.0113	0.0009	0.
	ʃ	0.0009	0.	0.0671	0.0353
	$\widehat{\text{tʃ}}$	0.0006	0.	0.0455	0.0313

Table: Results of SP_2 estimation on the Samala corpus. Only sibilants are shown. (Heinz and Rogers 2010, p. 894)

The Co-Emission Product (definition)



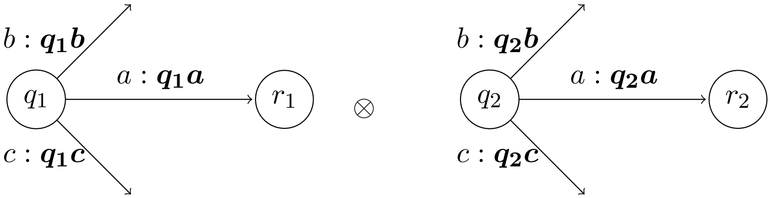
The Co-Emission Product (definition)



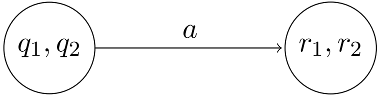
$$= \quad (q_1, q_2) \xrightarrow{\;a\;} (r_1, r_2) \quad \text{where}$$

$$P(a \mid q_1, q_2) \quad \stackrel{\text{def}}{=} \quad \frac{\prod_i \boldsymbol{q_i a}}{\sum_\sigma \prod_i \boldsymbol{q_i \sigma}}$$

The Co-Emission Product (definition)



$=$



where

$$P(a \mid q_1, q_2) \stackrel{\text{def}}{=} \frac{\prod_i \boldsymbol{q_i a}}{\sum_\sigma \prod_i \boldsymbol{q_i \sigma}}$$

For fixed σ, the co-emission product treats the parameters $\boldsymbol{q_i \sigma}$ as *independent*.

Contributions

1. We extend Heinz and Rogers 2010 analysis to classes defined with

Contributions

1. We extend Heinz and Rogers 2010 analysis to classes defined with
    1. the standard co-emission product (not the variant introduced by Heinz and Rogers)

Contributions

1. We extend Heinz and Rogers 2010 analysis to classes defined with
    1. the standard co-emission product (not the variant introduced by Heinz and Rogers)
    2. of arbitrary sets of finitely many PDFAs (not just the ones which define stochastic SP_k languages)

Contributions

1. We extend Heinz and Rogers 2010 analysis to classes defined with
   1. the standard co-emission product (not the variant introduced by Heinz and Rogers)
   2. of arbitrary sets of finitely many PDFAs (not just the ones which define stochastic SP_k languages)
2. Essentially, we prove that parameters which maximize the probability of the data with respect to such models are found by running the corpus through each of the individual factor PDFAs and calculating the relative frequencies.

Some details of the analysis

1. Probability of Words
2. Relative Frequency of Emissions
3. Empirical Mean of co-emission probabilities
4. Main Theorems

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \ldots M_K$.

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \ldots M_K$.
- Suppose that $w = \sigma_1 \cdots \sigma_N$

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \dots M_K$.
- Suppose that $w = \sigma_1 \cdots \sigma_N$
- Let $q(j, i)$ denote a state in Q_j that is reached after M_j reads the prefix $\sigma_1 \cdots \sigma_{i-1}$.

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \ldots M_K$.
- Suppose that $w = \sigma_1 \cdots \sigma_N$
- Let $q(j, i)$ denote a state in Q_j that is reached after M_j reads the prefix $\sigma_1 \cdots \sigma_{i-1}$.
- If $i = 1$ then $q(j, i)$ represents the initial state of M_j.

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \ldots M_K$.
- Suppose that $w = \sigma_1 \cdots \sigma_N$
- Let $q(j,i)$ denote a state in Q_j that is reached after M_j reads the prefix $\sigma_1 \cdots \sigma_{i-1}$.
- If $i = 1$ then $q(j,i)$ represents the initial state of M_j.
- Let $T_j(q,\sigma)$ denote a parameter (transitional probabability) in PDFA M_j.

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \dots M_K$.
- Suppose that $w = \sigma_1 \cdots \sigma_N$
- Let $q(j,i)$ denote a state in Q_j that is reached after M_j reads the prefix $\sigma_1 \cdots \sigma_{i-1}$.
- If $i = 1$ then $q(j,i)$ represents the initial state of M_j.
- Let $T_j(q,\sigma)$ denote a parameter (transitional probabability) in PDFA M_j.
- Then the probability that σ is emitted after the product machine $\bigotimes_{1 \leq j \leq K} \mathcal{M}_j$ reads the prefix $\sigma_1 \cdots \sigma_{i-1}$ is the following:

$$\mathsf{Coemit}(\sigma, i) = \frac{\prod_{j=1}^{K} T_j(q(j,i),\sigma)}{\sum_{\sigma' \in \Sigma} \prod_{j=1}^{K} T_j(q(j,i),\sigma').} \qquad (1)$$

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \dots M_K$.
- Suppose that $w = \sigma_1 \cdots \sigma_N$
- Let $q(j,i)$ denote a state in Q_j that is reached after M_j reads the prefix $\sigma_1 \cdots \sigma_{i-1}$.
- If $i = 1$ then $q(j,i)$ represents the initial state of M_j.
- Let $T_j(q,\sigma)$ denote a parameter (transitional probabability) in PDFA M_j.
- Then the probability that σ is emitted after the product machine $\bigotimes_{1 \le j \le K} \mathcal{M}_j$ reads the prefix $\sigma_1 \cdots \sigma_{i-1}$ is the following:

$$\mathsf{Coemit}(\sigma, i) = \frac{\prod_{j=1}^{K} T_j(q(j,i),\sigma)}{\sum_{\sigma' \in \Sigma} \prod_{j=1}^{K} T_j(q(j,i),\sigma').} \tag{1}$$

- We assume that there is a end marker $\ltimes \in \Sigma$ which uniquely occurs at the end of words.

Probability of words

- Consider a class C defined with the co-emission product of K machines $M_1 \dots M_K$.
- Suppose that $w = \sigma_1 \cdots \sigma_N$
- Let $q(j,i)$ denote a state in Q_j that is reached after M_j reads the prefix $\sigma_1 \cdots \sigma_{i-1}$.
- If $i = 1$ then $q(j,i)$ represents the initial state of M_j.
- Let $T_j(q,\sigma)$ denote a parameter (transitional probabability) in PDFA M_j.
- Then the probability that σ is emitted after the product machine $\bigotimes_{1 \le j \le K} \mathcal{M}_j$ reads the prefix $\sigma_1 \cdots \sigma_{i-1}$ is the following:

$$\mathsf{Coemit}(\sigma, i) = \frac{\prod_{j=1}^{K} T_j(q(j,i),\sigma)}{\sum_{\sigma' \in \Sigma} \prod_{j=1}^{K} T_j(q(j,i),\sigma').} \qquad (1)$$

- We assume that there is a end marker $\ltimes \in \Sigma$ which uniquely occurs at the end of words.

$$P(w\ltimes) = \prod^{N+1} \mathsf{Coemit}(\sigma_i, i) \qquad (2)$$

Relative Frequency of Emission

Relative Frequency of Emission

- Let $m_w(M_j, q, \sigma) \in \mathbb{Z}^+$ denote how many times σ is emitted at the state q while the machine M_j emits w.

Relative Frequency of Emission

- Let $m_w(M_j, q, \sigma) \in \mathbb{Z}^+$ denote how many times σ is emitted at the state q while the machine M_j emits w.
- Let $n_w(M_j, q) \in \mathbb{Z}^+$ denote how many times the state q is visited while the machine M_j emits w.

Relative Frequency of Emission

- Let $m_w(M_j, q, \sigma) \in \mathbb{Z}^+$ denote how many times σ is emitted at the state q while the machine M_j emits w.
- Let $n_w(M_j, q) \in \mathbb{Z}^+$ denote how many times the state q is visited while the machine M_j emits w.

Then

$$\mathsf{freq}_w(\sigma|M_j, q) = \frac{m_w(M_j, q, \sigma)}{n_w(M_j, q)}, \tag{3}$$

represents the relative frequency that M_j emits σ at q during emission of w.

Relative Frequency of Emission

- Let $m_w(M_j, q, \sigma) \in \mathbb{Z}^+$ denote how many times σ is emitted at the state q while the machine M_j emits w.
- Let $n_w(M_j, q) \in \mathbb{Z}^+$ denote how many times the state q is visited while the machine M_j emits w.

Then

$$\mathsf{freq}_w(\sigma | M_j, q) = \frac{m_w(M_j, q, \sigma)}{n_w(M_j, q)}, \tag{3}$$

represents the relative frequency that M_j emits σ at q during emission of w.

It is straightforward to lift this definition to data sequences $D = \langle w_1 \ltimes, w_2 \ltimes, \ldots w_{|D|} \ltimes \rangle$ by letting $w = w_1 \ltimes w_2 \ltimes \ldots w_{|D|} \ltimes$.

Empirical Mean of co-emission probabilities

Empirical Mean of co-emission probabilities

$$\mathsf{sumCoemit}_w(\sigma, M_j, q) = \sum_{i \text{ s.t. } q(j,i)=q} \mathsf{Coemit}(\sigma, i).$$

Empirical Mean of co-emission probabilities

$$\mathsf{sumCoemit}_w(\sigma, M_j, q) = \sum_{i \text{ s.t. } q(j,i)=q} \mathsf{Coemit}(\sigma, i).$$

The *empirical mean of a co-emission probability* is defined as follows:

$$\overline{\mathsf{Coemit}}_w(\sigma | M_j, q) = \frac{\mathsf{sumCoemit}_w(\sigma, M_j, q)}{n_w(M_j, q)}, \tag{4}$$

Empirical Mean of co-emission probabilities

$$\mathsf{sumCoemit}_w(\sigma, M_j, q) = \sum_{i \text{ s.t. } q(j,i)=q} \mathsf{Coemit}(\sigma, i).$$

The *empirical mean of a co-emission probability* is defined as follows:

$$\overline{\mathsf{Coemit}}_w(\sigma | M_j, q) = \frac{\mathsf{sumCoemit}_w(\sigma, M_j, q)}{n_w(M_j, q)}, \tag{4}$$

This is the sample average of the co-emission probability when $q \in Q_j$ is visited.

Main Theorem

Main Theorem

Consider any parameter $T_j(q, \sigma)$ in PDFA M_j.

Theorem

$\partial P(D)/\partial T_j(q, \sigma) = 0$ holds for all j if and only if the following equation is satisfied for all $1 \leq j \leq K$:

$$\mathsf{freq}_w(\sigma | M_j, q) = \overline{\mathsf{Coemit}}_w(\sigma | M_j, q) \ .$$

Example

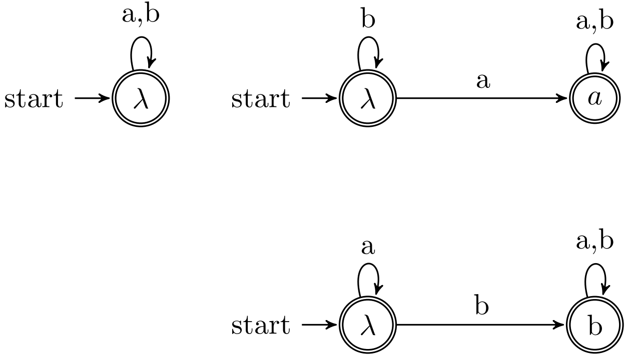


Figure: The 2-set of of SD-PDFAs with $\Sigma = \{a, b\}$. There are 15 parameters. Suppose $D = abb \ltimes bbb\ltimes$.

Example

$$\begin{array}{rrr}
\mathsf{freq}_D(a|\mathcal{M}_\lambda,\lambda) = 1/8 & \mathsf{freq}_D(a|\mathcal{M}_a,\lambda) = 1/5 & \mathsf{freq}_D(a|\mathcal{M}_a,a) = 0/3, \\
\mathsf{freq}_D(b|\mathcal{M}_\lambda,\lambda) = 5/8 & \mathsf{freq}_D(b|\mathcal{M}_a,\lambda) = 3/5 & \mathsf{freq}_D(b|\mathcal{M}_a,a) = 2/3, \\
\mathsf{freq}_D(\ltimes|\mathcal{M}_\lambda,\lambda) = 2/8 & \mathsf{freq}_D(\ltimes|\mathcal{M}_a,\lambda) = 1/5 & \mathsf{freq}_D(\ltimes|\mathcal{M}_a,a) = 1/3, \\
 & \mathsf{freq}_D(a|\mathcal{M}_b,\lambda) = 1/3 & \mathsf{freq}_D(a|\mathcal{M}_b,b) = 3/5, \\
 & \mathsf{freq}_D(b|\mathcal{M}_b,\lambda) = 2/3 & \mathsf{freq}_D(b|\mathcal{M}_b,b) = 0/5, \\
 & \mathsf{freq}_D(\ltimes|\mathcal{M}_b,\lambda) = 0/3 & \mathsf{freq}_D(\ltimes|\mathcal{M}_b,b) = 2/5,
\end{array}$$

FIGURE: Frequency computations with $D = abb \ltimes bbb\ltimes$ and the 2-set of of SD-PDFAs on previous slide.

Convexity of the Negative Log Likelihood

Convexity of the Negative Log Likelihood

Let $\tau_{j,q,\sigma}$ denote $\log T_j(q, \sigma)$; i.e. the log of a parameter of C defined with $\bigotimes_j M_j$.

Convexity of the Negative Log Likelihood

Let $\tau_{j,q,\sigma}$ denote $\log T_j(q, \sigma)$; i.e. the log of a parameter of C defined with $\bigotimes_j M_j$.

Then τ can be thought of as a vector in $\mathbb{R}^n$ where n is the number of parameters.

Convexity of the Negative Log Likelihood

Let $\tau_{j,q,\sigma}$ denote $\log T_j(q,\sigma)$; i.e. the log of a parameter of C defined with $\bigotimes_j M_j$.

Then τ can be thought of as a vector in $\mathbb{R}^n$ where n is the number of parameters.

Theorem

$-\log P(w\ltimes)$ is convex with respect to $\tau \in \mathbb{R}^n$.

Convexity of the Negative Log Likelihood

Let $\tau_{j,q,\sigma}$ denote $\log T_j(q,\sigma)$; i.e. the log of a parameter of C defined with $\bigotimes_j M_j$.

Then τ can be thought of as a vector in $\mathbb{R}^n$ where n is the number of parameters.

Theorem

$-\log P(w \ltimes)$ is convex with respect to $\tau \in \mathbb{R}^n$.

Thus the solution obtained by the previous theorem is a MLE.

Discussion

Discussion

At a high level, the problem we considered is a decomposition of complex probability distributions into simpler factors.

Discussion

At a high level, the problem we considered is a decomposition of complex probability distributions into simpler factors.

This has also been studied in the context of Bayesian networks, Markov random fields, and probabilistic graphical models more generally (Bishop, 2006; Koller and Friedman, 2009).

Discussion

At a high level, the problem we considered is a decomposition of complex probability distributions into simpler factors.

This has also been studied in the context of Bayesian networks, Markov random fields, and probabilistic graphical models more generally (Bishop, 2006; Koller and Friedman, 2009).

A reviewer points out that this literature may simplify our proofs.

Future Work

1. Language Modeling with various sets of specific factors and various corpora such as ...
   1. $\mathrm{SL}_k + \mathrm{SP}_k$
   2. $\mathrm{SLP}_{k,\ell}$ (Rogers and Lambert 2019, MoL)
   3. Atomic PDFA based on phonological features (Chandlee et al. 2019, MoL)
2. ...and compare to NNs, ALERGIA, and other algorithms on various benchmarks.
3. Connections to probabababilistic graphical models
4. Extend results to weighted deterministic automata.

Thanks

We acknowledge Canaan Breiss, Morgan Cassels, Huteng Dai, Danny DeSantiago, Anton Kukhto, Jon Rawski, Yang Wang, and Yuhong Zhu for valuable feedback on a draft presentation.

Questions?