Incremental Computation of Infix Probabilities for Probabilistic Finite Automata

Marco Cognetta Yo-Sub Han Soon Chan Kwon

Yonsei University, Seoul, Republic of Korea

http://toc.yonsei.ac.kr

Problem

We study the problem of *incrementally* computing infix probabilities of strings.

$$\underbrace{\mathcal{P}(\Sigma^* w\Sigma^*)}_{\text{Infix probability of } w} = \sum_{x\in\Sigma^* w\Sigma^*} \overbrace{\mathcal{P}(x)}^{\text{Probability of } x}$$

Use the infix probability of w to compute the infix probability of wa.

$$\mathcal{P}(\Sigma^* w\Sigma^*) \rightarrow \mathcal{P}(\Sigma^* wa\Sigma^*)$$

Given $w = w_1w_2\dots w_n$, calculate the infix probability of each prefix of w.

Previous Approach

Step	Time Complexity
·Construct DFA $\mathcal{D}$ for $\Sigma^* w\Sigma^*$	$O(\lvert w\rvert)$, Knuth-Morris-Pratt
·Create the automaton $\mathcal{D}\cap\mathcal{P}$	$O(\lvert Q_\mathcal{D}\rvert\lvert Q_\mathcal{P}\rvert) = O(\lvert w\rvert\lvert Q_\mathcal{P}\rvert)$

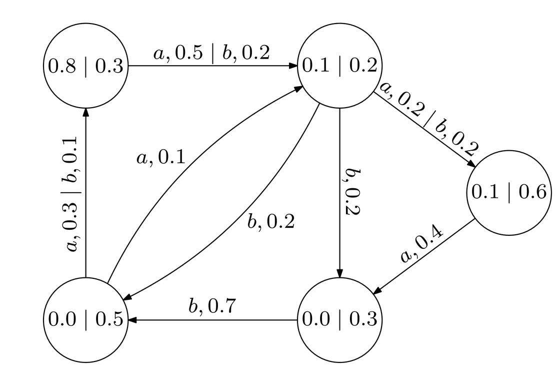
$$[\mathcal{D}\cap\mathcal{P}](w) = \begin{cases}\mathcal{P}(w), & w\in L(\mathcal{D})\\ 0, & \text{otherwise}\end{cases}$$

Step	Time Complexity
·Compute $\sum_{x\in\Sigma^*}[\mathcal{D}\cap\mathcal{P}](x)$	$O((\lvert w\rvert\lvert Q_\mathcal{P}\rvert)^m)$
·Repeat for each prefix of w	$O(\lvert w\rvert(\lvert w\rvert\lvert Q_\mathcal{P}\rvert)^m)$

Notation

PFA $\mathcal{P} = (Q_\mathcal{P}, \Sigma, \{\mathbb{M}_\mathcal{P}(c)\}_{c\in\Sigma}, \mathbb{I}_\mathcal{P}, \mathbb{F}_\mathcal{P})$

- $\mathbb{M}_\mathcal{P}(c)$ – $|Q_\mathcal{P}|\times|Q_\mathcal{P}|$ transition matrix
- $\mathbb{I}_\mathcal{P}$ – $1\times|Q_\mathcal{P}|$ initial weight vector
- $\mathbb{F}_\mathcal{P}$ – $|Q_\mathcal{P}|\times 1$ final weight vector



$$\mathcal{P}(w) = \mathbb{I}_\mathcal{P}\prod_{i=1}^{|w|}\mathbb{M}_\mathcal{P}(w_i)\mathbb{F}_\mathcal{P}$$

State Elimination

Label DFA states 1 to $|Q| = n$. Add two new states q_0 and q_{n+1} and λ transitions from q_0 to q_1 and final states to q_{n+1}.

$\alpha^k_{i,j}$ = paths leading from state i to state j that only visit states up to k

Eliminate state k: $\alpha^k_{i,j} = \alpha^{k-1}_{i,j} + \alpha^{k-1}_{i,k}(\alpha^{k-1}_{k,k})^*\alpha^{k-1}_{k,j}$

$$\text{Base cases: } \alpha^0_{i,j} = \begin{cases}\lambda, & i = 0\wedge j = 1\\ \lambda, & q_i\in F\wedge j = n+1\\ \{c \mid \delta(q_i, c) = q_k\}, & \text{otherwise.}\end{cases}$$

Unambiguous Regular Expressions

An *unambiguous* regular expression is one that can only be matched to a string in one way. State elimination on a DFA gives an unambiguous regular expression.

Two regular expressions for all strings containing aa as an infix.

Ambiguous	Unambiguous
$(a+b)^*aa(a+b)^*$	$b^*a(bb^*a)^*a(a+b)^*$
$aaaaa$	$aaaaa$
$aaaaa$	$baaab$
$aaaaa$	$abaab$

Regex → Transition Matrix

We extract a transition matrix from a regular expression.

Regex	Matrix
$\emptyset$	0
λ	1
c	$\mathbb{M}_\mathcal{P}(c)$

Regex	Matrix
$R\cup S$	$\mathbb{M}_\mathcal{P}(R)+\mathbb{M}_\mathcal{P}(S)$
RS	$\mathbb{M}_\mathcal{P}(R)\mathbb{M}_\mathcal{P}(S)$
R^*	$(1-\mathbb{M}_\mathcal{P}(R))^{-1}$

$$R \text{ is unambiguous} \rightarrow \mathbb{I}\mathbb{M}_\mathcal{P}(R)\mathbb{F} = \sum_{w\in L(R)}\mathcal{P}(w)$$

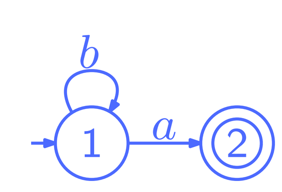
Incrementally Generating Infix Regex

Let $\mathcal{F}(w)$ be the set of strings ending in the <u>first</u> occurrence of w. Thus, $\mathcal{F}(w)\Sigma^* = \Sigma^* w\Sigma^*$.

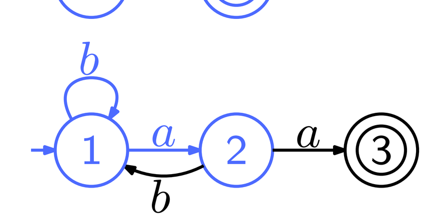
Left Quotient: $R\backslash S = \{y \mid \exists x\in R \text{ s.t. } xy\in S\}$

$\mathcal{F}(wa) = \mathcal{F}(w)\cdot\mathcal{F}(w)\backslash\mathcal{F}(wa)$

$\mathcal{F}(a) = b^*a$

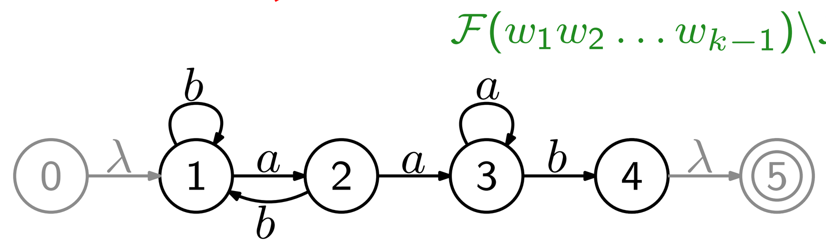


$\mathcal{F}(aa) = b^*a(bb^*a)^*a$



Key Idea

$$\alpha^k_{0,k+1} = \mathcal{F}(w_1w_2\dots w_k) = \overset{\emptyset}{\cancel{\boxed{\alpha^{k-1}_{0,k+1}}}} + \overset{\mathcal{F}(w_1w_2\dots w_{k-1})}{\boxed{\alpha^{k-1}_{0,k}}}\underset{\mathcal{F}(w_1w_2\dots w_{k-1})\backslash\mathcal{F}(w_1w_2\dots w_{k-1}w_k)}{\boxed{(\alpha^{k-1}_{k,k})^*\alpha^{k-1}_{k,k+1}}}$$



α^0

	0	1	2	3	4	5
0	$\emptyset$	λ	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\emptyset$	b	a	$\emptyset$	$\emptyset$	$\emptyset$
2	$\emptyset$	b	$\emptyset$	a	$\emptyset$	$\emptyset$
3	$\emptyset$	$\emptyset$	$\emptyset$	a	b	$\emptyset$
4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	λ
5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

α^1

	0	1	2	3	4	5
0	$\emptyset$	$\lambda+b^*b$	b^*a	$\emptyset$	$\emptyset$	$\emptyset$
1	$\emptyset$	$b+bb^*b$	$a+bb^*a$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\emptyset$	$b+bb^*b$	bb^*a	a	$\emptyset$	$\emptyset$
3	$\emptyset$	$\emptyset$	$\emptyset$	a	b	$\emptyset$
4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	λ
5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

$\mathcal{F}(a) = b^*a$ $\mathcal{F}(aa) = b^*a(bb^*a)^*a$

Algorithm 1: Incremental Infix Probability

Data: PFA $\mathcal{P}$, String w

$\mathcal{D}\leftarrow$ KMP DFA for w
$n\leftarrow|Q_\mathcal{D}|$
$T, T'\leftarrow(n+2)\times(n+2)$ table
for $i, j\in[0, n+1]$ **do**
  if $(i = 0\wedge j = 1)\vee q_i\in F\wedge j = n+1$ **then**
    $T_{i,j} = 1$
  else
    for c *such that* $\delta(q_i, c) = q_j$ **do**
      $T_{i,j} = T_{i,j} + \mathbb{M}_\mathcal{P}(c)$
$\mathbb{V}\leftarrow\mathbb{I}_\mathcal{P}$
for $k\in[1, n]$ **do**
  $\mathbb{V}\leftarrow\mathbb{V}(T_{k,k})^*T_{k,k+1}$
  yield $\mathbb{V}\mathbb{M}_\mathcal{P}(\Sigma^*)\mathbb{F}_\mathcal{P}$
  for $i, j\in[0, n+1]$ **do**
    $T'_{i,j} = T_{i,j} + T_{i,k}(T_{k,k})^*T_{k,j}$
  $T\leftarrow T'$

Time Complexity: $O(|w|^3|Q_\mathcal{P}|^m)$

Experimental Results

States	$\lvert Q\rvert = 614$		$\lvert Q\rvert = 1028$		$\lvert Q\rvert = 1455$	
Infix Length	Incremental	Intersection	Incremental	Intersection	Incremental	Intersection
1	0.226	0.147	0.857	0.468	2.383	1.079
2	0.272	0.316	1.072	1.235	3.000	3.112
3	0.334	0.637	1.327	2.634	3.693	6.997
4	0.399	1.133	1.586	4.864	4.442	13.250
5	0.465	1.934	1.855	8.104	5.124	22.357
6	0.527	3.375	2.088	12.562	5.815	35.065
7	0.584	4.129	2.347	18.414	6.593	51.709
8	0.649	5.791	2.591	25.614	7.224	72.512
9	0.711	7.879	2.851	34.959	7.950	99.347
Total	4.169	25.342	16.574	108.853	46.224	305.428

One iteration of the incremental algorithm is much faster than scrapping the computation and starting over (intersection method).

The total time to compute all infixes by the incremental method is less than the time to compute just the longest infix by intersection.

Future Directions

- Backwards incremental infix computation.

$$\mathcal{P}(\Sigma^* w_iw_{i+1}\dots w_n\Sigma^*) \rightarrow \mathcal{P}(\Sigma^* w_{i-1}w_iw_{i+1}\dots w_n\Sigma^*)$$

- Streaming incremental infix computation.

  Instead of knowing all of w at the beginning, receive characters one-by-one and compute the current infix probability on the fly.

- Two sided incremental infix probability.

  In the streaming setting, allow characters to be prepended or appended at will, instead of always being added to the end.

- Incremental infix probability calculation for PCFGs.

The first three can be solved non-incrementally in $O(|w|(|w||Q_\mathcal{P}|)^m)$ time using the intersection algorithm. Can we acheive a similar speedup with a modified incremental approach?

- Mark-Jan Nederhof and Giorgio Satta. *Computation of infix probabilities for probabilistic context-free grammars.* EMNLP 2011, pp 1213–1221.
- Ronald Book et al. 1971. *Ambiguity in graphs and expressions.* IEEE Transactions on Computers, 20:149–153.

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIP) (2018-0-00247, 2018-0-00276).