Efficient computation of ℓ_2 distance between weighted automata

Guillaume Rabusseau

April 8, 2018

Let $A = (\mathbb{R}^m, \boldsymbol{\alpha}_0, \{\mathbf{A}^\sigma\}_{\sigma\in\Sigma}, \boldsymbol{\alpha}_\infty)$ and $B = (\mathbb{R}^n, \boldsymbol{\beta}_0, \{\mathbf{B}^\sigma\}_{\sigma\in\Sigma}, \boldsymbol{\beta}_\infty)$ be two WAs. We are interested in computing the ℓ_2 distance between the functions they compute:

$$\|f_A - f_B\|_2^2 = \sum_{x\in\Sigma^*}(f_A(x) - f_B(x))^2.$$

The naive way

Construct a WA C computing the function $x \mapsto (f_A(x) - f_B(x))^2$ and compute its sum $\sum_{x\in\Sigma^*} f_C(x)$ using classical constructions (note that C has $(m+n)^2$ states).

The naive way slightly improving on `splearn`

To compute the sum $\sum_{x\in\Sigma^*} f_C(x)$ `splearn` uses

$$\sum_{x\in\Sigma^*} f_C(x) = \boldsymbol{\gamma}_0^\top(\mathbf{I} - \sum_{\sigma\in\Sigma}\mathbf{C}_\sigma)^{-1}\boldsymbol{\gamma}_\infty$$

and computes explicitly the inverse. Solving the linear system

$$(\mathbf{I} - \sum_{\sigma\in\Sigma}\mathbf{C}_\sigma)\mathbf{x} = \boldsymbol{\gamma}_\infty$$

using `numpy.linalg.solve` and getting the sum with $\boldsymbol{\gamma}_0^\top\mathbf{x}$ seems more efficient.

Using the Gramians

Define the following Gramians:

$$\begin{aligned}
\mathbf{G}_A &= \sum_{x\in\Sigma^*}\mathbf{A}_x\boldsymbol{\alpha}_\infty\boldsymbol{\alpha}_\infty^\top\mathbf{A}_x^\top\\
\mathbf{G}_B &= \sum_{x\in\Sigma^*}\mathbf{B}_x\boldsymbol{\beta}_\infty\boldsymbol{\beta}_\infty^\top\mathbf{B}_x^\top\\
\mathbf{G}_{A,B} &= \sum_{x\in\Sigma^*}\mathbf{A}_x\boldsymbol{\alpha}_\infty\boldsymbol{\beta}_\infty^\top\mathbf{B}_x^\top.
\end{aligned}$$

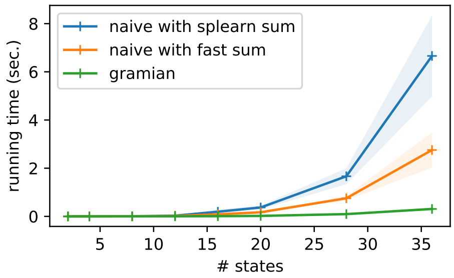
We have

$$\begin{aligned}
\|f_A - f_B\|_2^2 &= \sum_{x\in\Sigma^*}(f_A(x) - f_B(x))^2\\
&= \sum_{x\in\Sigma^*}(\boldsymbol{\alpha}_0^\top\mathbf{A}_x\boldsymbol{\alpha}_\infty - \boldsymbol{\beta}_0^\top\mathbf{B}_x\boldsymbol{\beta}_\infty)^2\\
&= \sum_{x\in\Sigma^*}(\boldsymbol{\alpha}_0^\top\mathbf{A}_x\boldsymbol{\alpha}_\infty - \boldsymbol{\beta}_0^\top\mathbf{B}_x\boldsymbol{\beta}_\infty)(\boldsymbol{\alpha}_\infty^\top\mathbf{A}_x^\top\boldsymbol{\alpha}_0 - \boldsymbol{\beta}_\infty^\top\mathbf{B}_x^\top\boldsymbol{\beta}_0)\\
&= \boldsymbol{\alpha}_0^\top\mathbf{G}_A\boldsymbol{\alpha}_0 + \boldsymbol{\beta}_0^\top\mathbf{G}_B\boldsymbol{\beta}_0 - 2\boldsymbol{\alpha}_0^\top\mathbf{G}_{A,B}\boldsymbol{\beta}_0.
\end{aligned}$$

The Gramians can be computed by solving linear systems, see [1, Theorem 20, Lemma 23].

A small numeric simulation

Time to compute the ℓ_2 distance between two random probabilistic automata (average over 3 runs):



References

[1] Borja Balle, Prakash Panangaden, and Doina Precup. Singular value automata and approximate minimization. *arXiv preprint arXiv:1711.05994*, 2017.