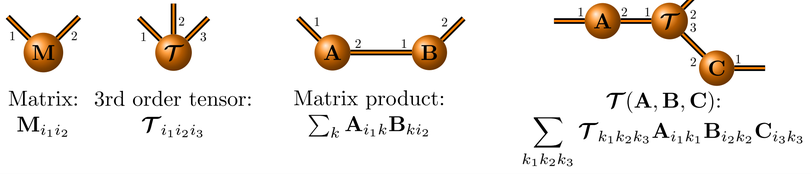
Low-Rank Approximation of Weighted Tree Automaton

Guillaume Rabusseau (Aix-Marseille University) Borja Balle (Lancaster University) Shay B. Cohen (University of Edinburgh)

Overview

- Weighted Tree Automata: model subsuming PCFG and L-PCFG.
- Aim: model reduction to speed-up inference (e.g. parsing).
- Method: similar to PCA for weighted context-free grammar.
- Iterative algorithm that resembles the power method for SVD.

Tensor Networks



Weighted Tree Automata

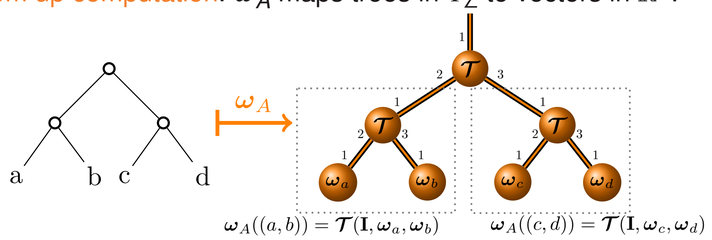
- A weighted tree automaton (WTA) is a tuple $\langle \boldsymbol{\alpha}, \mathcal{T}, \{\boldsymbol{\omega}_\sigma\}_{\sigma\in\Sigma}\rangle$

  $\boldsymbol{\alpha} \in \mathbb{R}^n$: vector of initial weights

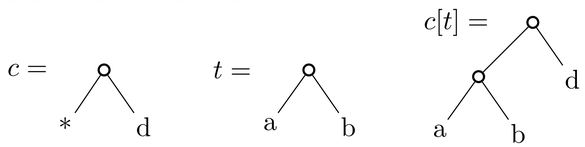
  $\mathcal{T} \in \mathbb{R}^{n\times n\times n}$: tensor of transition weights

  $\boldsymbol{\omega}_\sigma \in \mathbb{R}^n$: vector of final weights associated with $\sigma \in \Sigma$

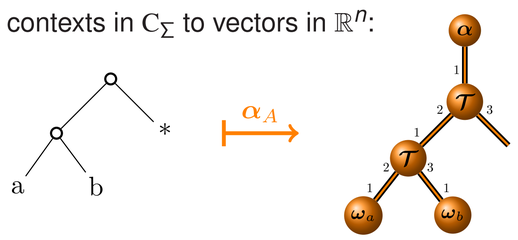
- Bottom up computation. ω_A maps trees in T_Σ to vectors in $\mathbb{R}^n$:



- The WTA A compute the function $f_A : t \mapsto \boldsymbol{\alpha}^\top \omega_A(t) \in \mathbb{R}$.
- Contexts are trees with a *hole:*



- α_A maps contexts in C_Σ to vectors in $\mathbb{R}^n$:



- For any tree context $c \in \mathrm{C}_\Sigma$ and tree $t \in \mathrm{T}_\Sigma$:

$$f_A(c[t]) = \alpha_A(c)^\top \omega_A(t) \ .$$

Rank Factorization of the Hankel Matrix

- The rank of a function $f : \mathrm{T}_\Sigma \to \mathbb{R}$ is the number of states of the smallest WTA computing f.
- The Hankel matrix $\mathbf{H}_f \in \mathbb{R}^{\mathrm{C}_\Sigma \times \mathrm{T}_\Sigma}$ is the bi-infinite matrix defined by

$$\mathbf{H}_f(c,t) = f(c[t]) \text{ for any } c \in \mathrm{C}_\Sigma,\ t \in \mathrm{T}_\Sigma.$$

Theorem (Bozapalidis and Louscou-Bozapalidou [1983]).

For any $f : \mathrm{T}_\Sigma \to \mathbb{R}$ we have $rank(f) = rank(\mathbf{H}_f)$.

- A WTA A with n states induces a rank n factorization $\mathbf{H}_f = \mathbf{P}_A\mathbf{S}_A$:
  - $\mathbf{P}_A \in \mathbb{R}^{\mathrm{C}_\Sigma \times n}$ defined by $\mathbf{P}_A(c,:) = \boldsymbol{\alpha}_A(c)$
  - $\mathbf{S}_A \in \mathbb{R}^{n \times \mathrm{T}_\Sigma}$ defined by $\mathbf{P}_S(:,t) = \boldsymbol{\omega}_A(t)$.
- One-to-one correspondence between rank factorizations of $\mathbf{H}_f$ and WTAs computing f:

Theorem.

Let $f : \mathrm{T}_\Sigma \to \mathbb{R}$ be of finite rank. If $\mathbf{H}_f = \mathbf{PS}$ is a rank factorization, then there exists a WTA A computing f such that $\mathbf{P}_A = \mathbf{P}$ and $\mathbf{S}_A = \mathbf{S}$.

Low-Rank Approximation of WTA

Problem: Given a WTA A with n states, find a WTA with $\hat{n} < n$ states that is close to A.

Idea: (following Balle et al. [2015] in the string case)

- Find a WTA $\hat{A}$ such that the rank-n factorization $\mathbf{H}_f = \mathbf{P}_{\hat{A}}\mathbf{S}_{\hat{A}}$ coincides with the SVD of $\mathbf{H}_f = \mathbf{UDV}^\top$:

$$\mathbf{P}_{\hat{A}} = \mathbf{U}\mathbf{D}^{1/2} \text{ and } \mathbf{S}_{\hat{A}} = \mathbf{D}^{1/2}\mathbf{V}^\top. \qquad (1)$$

- The ith state of $\hat{A}$ corresponds to the ith singular value of $\mathbf{H}_f$.
- Remove the $n - \hat{n}$ states corresponding to the smallest singular values.

We call a WTA satisfying (1) a singular value tree automaton (SVTA).

Computing the SVTA

Given a WTA $A = \langle \boldsymbol{\alpha}, \mathcal{T}, \{\boldsymbol{\omega}_\sigma\}_{\sigma\in\Sigma}\rangle$, how to find an SVTA computing the same function?

- Obvious way: compute the SVD of the bi-inifinite matrix $\mathbf{H}_f = \mathbf{PS}$.
- In practice:
  - *Gram matrices* $\mathbf{G}_\mathrm{C} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{n\times n}$ and $\mathbf{G}_\mathrm{T} = \mathbf{SS}^\top \in \mathbb{R}^{n\times n}$.
  - Compute $\mathbf{Q}$ from the *eigendecompositions* of $\mathbf{G}_\mathrm{C}$ and $\mathbf{G}_\mathrm{T}$.

$\Rightarrow$ The WTA $\hat{A} = \langle \mathbf{Q}^\top\boldsymbol{\alpha}, \mathcal{T}(\mathbf{Q}^{-\top}, \mathbf{Q}, \mathbf{Q}), \{\mathbf{Q}^{-1}\boldsymbol{\omega}_\sigma\}\rangle$ is a SVTA computing f.

Computing the Gram Matrices

- Leverage the *fixed point equations*:

$$\mathbf{G}_\mathrm{T} = \sum_{\sigma\in\Sigma} \boldsymbol{\omega}_\sigma\boldsymbol{\omega}_\sigma^\top + \mathbf{T}_{(1)}(\mathbf{G}_\mathrm{T} \otimes \mathbf{G}_\mathrm{T})\mathbf{T}_{(1)}^\top$$

$$\mathbf{G}_\mathrm{C} = \boldsymbol{\alpha}\boldsymbol{\alpha}^\top + \mathbf{T}_{(2)}(\mathbf{G}_\mathrm{C} \otimes \mathbf{G}_\mathrm{T})\mathbf{T}_{(2)}^\top + \mathbf{T}_{(3)}(\mathbf{G}_\mathrm{C} \otimes \mathbf{G}_\mathrm{T})\mathbf{T}_{(3)}^\top$$

$\Rightarrow$ Iterative algorithm converging exponentially fast:

Theorem.

There exists $0 < \rho < 1$ such that after k iterations , the approximations $\hat{\mathbf{G}}_\mathrm{C}$ and $\hat{\mathbf{G}}_\mathrm{T}$ satisfy $\|\mathbf{G}_\mathrm{C} - \hat{\mathbf{G}}_\mathrm{C}\|_F \leq \mathcal{O}(\rho^k)$ and $\|\mathbf{G}_\mathrm{T} - \hat{\mathbf{G}}_\mathrm{T}\|_F \leq \mathcal{O}(\rho^k)$.

Experiments

- PCFG with $n = 211$ nonterminals learnt from the annotated corpus of German newspaper texts NEGRA [Skut et al., 1997].
- Comparison with Cohen et al. [2013] (decomposition of the tensors of the PCFG).
- Number of parameters: $\hat{n}^3$ for a WTA with $\hat{n}$ states, and $3Rn$ when the tensor $\mathcal{T}$ is approximated with a tensor of CP-rank R .

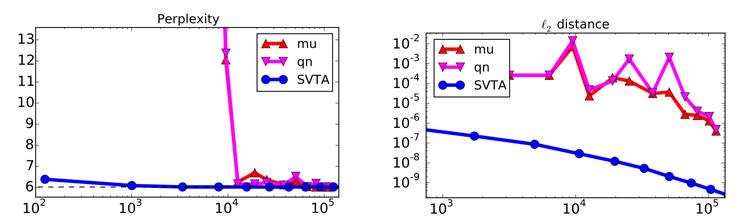


Figure: Perplexity and ℓ_2 distance w.r.t. number of parameters.

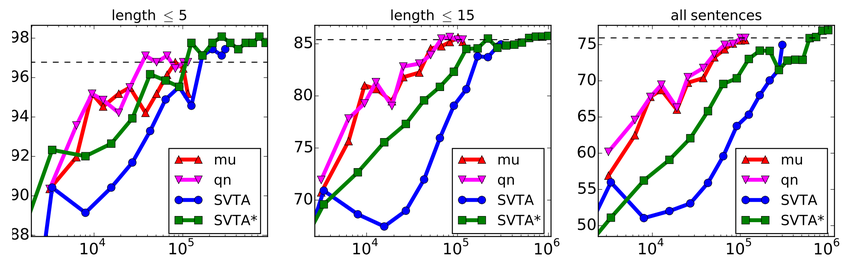


Figure: Parsing accuracy w.r.t. number of parameters.

References

B. Balle, P. Panangaden, and D. Precup. A canonical form for weighted automata and applications to approximate minimization. In *Proceedings of LICS*, 2015.

S. Bozapalidis and O. Louscou-Bozapalidou. The rank of a formal tree power series. *Theoretical Computer Science*, 1983.

S. B. Cohen, G. Satta, and M. Collins. Approximate PCFG parsing using tensor decomposition. In *Proceedings of NAACL*, 2013.

W. Skut, B. Krenn, T. Brants, and H. Uszkoreit. An annotation scheme for free word order languages. In *Conference on Applied Natural Language Processing*, 1997.

Emails: guillaume.rabusseau@lif.univ-mrs.fr b.deballepigem@lancaster.ac.uk scohen@inf.ed.ac.uk