Unitary Branching Programs: Learnability and Lower Bounds

Fidel Ernesto Díaz Andino
University of São Paulo (Brazil)

Maria Kokkou
Chalmers University of Technology (Sweden)

Mateus de Oliveira Oliveira
University of Bergen (Norway)

Farhad Vadiee
University of Bergen (Norway)

This work was supported by the Research Council of Norway (Proj no. 288761).

Main Computational Model

- We introduced the notion of a unitary branching program (UBP).
  - Builds on the notion of a program over a monoid defined by Barrington.
  - A slightly different acceptance condition.
- Computational power.
  - Constant-dimension UBPs generalize the traditional model of constant-width BPs.
  - Therefore, any function computable by polynomial-size circuits of logarithmic depth can be computed by a constant dimension UBP of polynomial length.
- Given the power of this model, nontrivial lower bounds are hard to obtain.

Some Quantitative Results

- $\Omega(\frac{n^2}{k^2 \log n})$ lower bound on the length of UBPs computing the n-bit element distinctness function.
- Any binary function $f : \{0,1\}^n \to \{0,1\}$ computable by a read-once dimension-k δ-gapped UBP can be represented by a DFA with $\left(\frac{n}{\delta}\right)^{O(k^2)}$ states.
- The class of dimension-k read-once δ-gapped UBPs of class size 2 can be exactly learned with $\left(\frac{n}{\delta}\right)^{O(k^2)}$ queries using the representation class of DFAs.
- The n-bit triangle-freeness function requires read-once δ-gapped UBPs of dimension $k = \Omega(\sqrt{n/(\log \frac{n}{\delta})})$.

Some Quantitative Results

- $\Omega(\frac{n^2}{k^2 \log n})$ lower bound on the length of UBPs computing the n-bit element distinctness function.
- Any binary function $f : \{0,1\}^n \to \{0,1\}$ computable by a read-once dimension-k δ-gapped UBP can be represented by a DFA with $\left(\frac{n}{\delta}\right)^{O(k^2)}$ states.
- The class of dimension-k read-once δ-gapped UBPs of class size 2 can be exactly learned with $\left(\frac{n}{\delta}\right)^{O(k^2)}$ queries using the representation class of DFAs.
- The n-bit triangle-freeness function requires read-once δ-gapped UBPs of dimension $k = \Omega(\sqrt{n/(\log \frac{n}{\delta})})$.

Some Quantitative Results

- $\Omega(\frac{n^2}{k^2 \log n})$ lower bound on the length of UBPs computing the n-bit element distinctness function.
- Any binary function $f : \{0,1\}^n \to \{0,1\}$ computable by a read-once dimension-k δ-gapped UBP can be represented by a DFA with $\left(\frac{n}{\delta}\right)^{O(k^2)}$ states.
- The class of dimension-k read-once δ-gapped UBPs of class size 2 can be exactly learned with $\left(\frac{n}{\delta}\right)^{O(k^2)}$ queries using the representation class of DFAs.
- The n-bit triangle-freeness function requires read-once δ-gapped UBPs of dimension $k = \Omega(\sqrt{n/(\log \frac{n}{\delta})})$.

A Heuristic for Learning UBPs

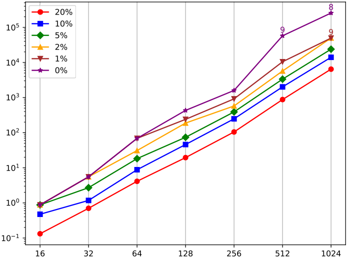
- The set of unitary matrices of dimension k forms a group that has the structure of a compact *connected* manifold known as the complex Stiefel manifold $V_k(\mathbb{C}^k)$.
- A branching program with l instructions, alphabet size s and class size c can be viewed as a point in $V_k(\mathbb{C}^k)^{l \cdot s + c}$.
- We formulate the problem of learning a UBP consistent with a given dataset as a minimization problem over this manifold.

Improvements

- Ideally, we would like to compute an optimal solution using off-the-shelf tools for Riemannian gradient descent. In practice this is too slow.
- Speed up the process significantly by using...
  - local optimization: Riemannian gradient descent is applied to a small window of instructions at a time (a much smaller space).
  - pre-computation: allows us to evaluate intermediate UBPs against the input dataset only at the beginning of each window optimization cycle.
- Implementation:
  - LUBP: Learning Unitary Branching Programs
  - Source code: `https://github.com/AutoProving/LUBP`

Experimental Results

- n-dataset: n positive strings and n negative strings from $\{0,1\}^n$. We refer to n as the *size* of the dataset.
- Point (n, t): t is the average time to learn a read-once dimension-3 UBP consistent with a randomly sampled n-dataset (10 sampled datasets) with a given error tolerance.



- Yellow Line: All datasets were learned with at most 2% error. Purple Line: Almost all datasets were learned with 0% error. Except one dataset of size 512 (average taken over 9 datasets) and 2 datasets of size 1024 (average taken over 8 datasets).

Open Problems

- Analytic proof of convergence for the task of learning dimension-3 UBPs consistent with a given n-dataset?
- In the opposite direction, an n-dataset that requires super-constant dimension when represented by read-once UBPs?
- Polynomial dimension lower-bounds for non-gapped read-once UBPs?