Interactive Oracle Proofs of Proximity for Algebraic Geometry codes

Sarah Bordage Jade Nardi

November 30, 2021

In this work, we initiate the study of proximity testing to Algebraic Geometry (AG) codes. An AG code $C = C(\mathcal{C}, \mathcal{P}, D)$ is a vector space associated to evaluations on $\mathcal{P}$ of functions in the Riemann-Roch space $L_{\mathcal{C}}(D)$. The problem of testing proximity to an error-correcting code C consists in distinguishing between the case where an input word, given as an oracle, belongs to C and the one where it is far from every codeword of C. AG codes are good candidates to construct *short* proof systems, but there exists no efficient proximity tests for them.

We construct an Interactive Oracle Proof of Proximity (IOPP) for some families of AG codes by generalizing an IOPP for Reed-Solomon codes, known as the FRI protocol [Ben-Sasson et al., ICALP 2018]. We identify suitable requirements for designing efficient IOPP systems for AG codes. We take advantage of the algebraic geometry framework that makes any group action on the curve that fixes the divisor D translate into a decomposition of the code C.

Our approach relies on Kani's result that splits the Riemann-Roch space of any invariant divisor under a group action on a curve into several explicit Riemann-Roch spaces on the quotient curve. Under some hypotheses, a proximity test to C can thus be reduced to one to a simpler code C'. Iterating this process thoroughly, we end up with a membership test to a code with significantly smaller length. In addition to proposing the first proximity test targeting AG codes, our IOPP admits quasilinear prover arithmetic complexity and sublinear verifier arithmetic complexity with constant soundness for meaningful classes of AG codes.

As a concrete instantiation, we study AG codes on Kummer curves, which are potentially much longer than Reed-Solomon codes. For this type of curves, we manage to extend our generic construction to reach a strictly linear proving time and a strictly logarithmic verification time.

The talk will be given by Sarah Bordage.