CONJURA · OPEN PROBLEM

Conflict Checkable Codes with Local-to-Global Consistency Beyond Half the Singleton Bound

Which of two properties costs the factor of two — and the source says it is the other one

Status. Statement: AI-written from Benny Applebaum and Eliran Kachlon, *Conflict Checkable and Decodable Codes and Their Applications*, IACR ePrint 2023/627. Not yet checked by a human. Their Theorem 1.8 proves $k \le (n - d + 2)/2$ for codes that are both comparison-based and local-to-global consistent. Their Theorem 1.3 gives an almost-MDS conflict checkable code, $k \ge n - d + 1 - \epsilon$, which bypasses that bound — but it is neither comparison-based nor known to be local-to-global consistent. The conjecture names which of the two properties is the culprit.

Category. *Coding Theory.*

Conjecture (informal). *Store a codeword across n servers, one symbol each, and have every pair of servers announce publicly whether their two symbols are consistent with some legal codeword. What can be inferred from that graph alone? Enough, it turns out, to check membership in the code, and — with an extra property — to decode. Two structural properties keep appearing together in the constructions: conflicts decided by comparing a function of each symbol, and consistency on any large enough set lifting to a genuine codeword. Together they cost exactly a factor of two in rate against the Singleton bound. The source has an almost-optimal code with neither, and conjectures the first property is what costs the factor.*

1 The setting

Let C be a code over an alphabet of size q with block length n, dimension k and distance d, and let a possibly corrupted codeword be distributed among n servers. The *conflict functions* $G = (G_{i,j})_{1 \le i < j \le n}$ record, for each pair of positions, whether the two symbols could

co-occur in a codeword; the resulting graph is the *conflict graph*. A code is *conflict checkable* if from the conflict graph of a vector one can decide whether the vector is a codeword.

Two further properties are the subject of the statement.

Definition 1 (Local-to-global consistency, [AK23, Lemma 1.6]). An $(n,k,d)_q$ conflict checkable code C provides *local-to-global consistency* if for every set $I \subseteq [n]$ of at least $n-d+1$ indices and every symbols $(\sigma_i)_{i\in I}$, if $G_{i,j}(\sigma_i,\sigma_j)=0$ for every $i<j$ in I, then there exists a codeword $c\in C$ with $c[i]=\sigma_i$ for every $i\in I$.

Definition 2 (Comparison-based, [AK23, §4]). C is *comparison-based* if for every $1\le i<j\le n$ there exist functions $f_{i,j}, f_{j,i} : [q]\to[q]$ such that $G_{i,j}(\sigma,\tau)=\mathrm{NEQ}(f_{i,j}(\sigma), f_{j,i}(\tau))$, where NEQ returns 1 if its arguments differ and 0 otherwise.

Why these matter: local-to-global consistency is a sufficient condition for conflict decodability (the source's Lemma 1.6), with an efficiently computable decoder, and it is *necessary* for robust conflict decodability. And every linear code over a finite field is comparison-based (its Theorem 4.7). So for linear codes the two properties come as a package, and the package has a price.

2 The conjecture

Theorem 1 (The rate bound, [AK23, Theorem 1.8]). *For every $(n,k,d)_q$ comparison-based conflict checkable code with $1<d<n$ that satisfies local-to-global consistency, $k\le \frac{n-d+2}{2}$. In particular the bound holds for any linear conflict checkable code that satisfies local-to-global consistency.*

Theorem 2 (The almost-MDS code, [AK23, Theorem 1.3]). *For every $n\ge 3$, every $2\le d\le n-1$ and every $\epsilon>0$ there is an alphabet size $q=q(n,d,\epsilon)$ for which there exists an $(n,k,d)_q$ conflict checkable code with $k\ge n-d+1-\epsilon$.*

Conjecture 1 ([AK23, §1.2]). *There is a conflict checkable code that satisfies local-to-global consistency and bypasses the bound of*

Theorem 1: that is, for some n and some $1 < d < n$, an $(n, k, d)_q$ conflict checkable code with local-to-global consistency and $k > \frac{n-d+2}{2}$.

Remark 1 (How the source states it, and what it is really claiming). Page 11: "We conjecture that the former property is the important one and that the bound from Theorem 1.8 can be bypassed by some conflict checkable code with local-to-global consistency." "The former property" is comparison-basedness. So the content is an attribution of blame: of the two hypotheses of Theorem 1, the source predicts that comparison-basedness is what forces the factor of two, and that local-to-global consistency alone is compatible with an almost-MDS rate. By Theorem 4.7 such a code cannot be linear.

Remark 2 (The conjecture is already true at one endpoint). The source records a special case in which it holds: "Observe that this conjecture holds for the special case of $d = n - 1$ since, in this case, the conflict checkable code from Theorem 1.3 trivially satisfies local-to-global consistency (any pair of consistent entries must be consistent with a unique codeword)." So the statement is not open across the whole parameter range, and a resolution should say which range of d it covers. The degenerate reason it holds at $d = n - 1$ — sets I of size $n - d + 1 = 2$ — also shows why that case carries no information about the rest.

Remark 3 (What is honestly not known about the candidate code). Theorem 2's code is the natural candidate, and the source is careful about its status: "We do not know whether the code from Theorem 1.3 satisfies local-to-global consistency." It is also non-explicit. So the shortest route to Conjecture 1 is to settle that one question about an existing object — and a proof that this particular code does *not* have the property would leave the conjecture open while removing its most obvious candidate.

Remark 4 (What the factor of two currently costs). It is not an abstract loss. The source's bivariate-polynomial code $C_{\text{bivariate}}$ is linear, conflict checkable and local-to-global consistent, and it "half-meets" the Singleton bound at $k = (n - d + 2)/2$ — so by Theorem 1 it is rate-optimal in its class. Downstream, Lemma 1.12 gives a Singleton-type bound $k \leq (n - 2t + 1)/2$ for comparison-based t-robust conflict decodable codes, met by the same code. A

code satisfying Conjecture 1 would break that whole chain of tight bounds, which is what makes the attribution of blame worth settling.

Remark 5 (A separate open question in the same paper). The source shows how to find a $(1+\epsilon)$-approximation of t-vertex covers in polynomial time in the graphs it works with, and notes that "[t]he question of finding an exact vertex cover in polynomial-time algorithm in such graphs remains as an interesting open question." That bears on decoder efficiency, not on rate, and it is not this statement.

3 Bibliography

[AK23] Benny Applebaum and Eliran Kachlon. *Conflict checkable and decodable codes and their applications.* IACR ePrint 2023/627. The source. Theorem 1.3 is on page 7, Lemma 1.6 on page 9, Theorem 1.8 and the conjecture on page 11, Lemma 1.12 on page 13, and the comparison-based definition and Theorem 4.7 in Section 4, pages 29 and 36.

[CDM00] Ronald Cramer, Ivan Damgård and Ueli Maurer. *General secure multi-party computation from any linear secret-sharing scheme.* EUROCRYPT 2000. One of the two works in which the symmetric-bivariate-polynomial code the source uses appears implicitly. [UNVERIFIED] — cited in the source as [CDM00]; the bibliographic detail here is inferred.

[RS97] Ran Raz and Shmuel Safra. *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP.* STOC 1997. The source notes that closely related arguments appear here, used to derive better soundness for a local test — an information-theoretic goal — whereas it uses the same combinatorial structure for an algorithmic one.