arXiv:2601.01372v1 [cs.IT] 4 Jan 2026

线性码的概率验证算法

李明超*[1] 和**李吉友**†[1]

[1]上海交通大学数学科学学院，中华人民共和国上海

摘要

本文提出一种适用于任意线性码 C 的概率算法，用于判断给定向量 $\mathbf{x}$ 是否属于 C。该算法的时间复杂度为 $O(n\log\ n)$，空间复杂度为 $O(n^2)$。渐近意义下，错误概率小于 $1/\text{poly}(n)$。

关键词： 概率算法，线性码。

1 引言

几十年来，线性码得到了广泛应用。针对特定码族开发解码算法已取得显著进展，其中包括用于 Reed–Solomon（RS）、BCH 和 Goppa 码的 Berlekamp-Massey算法 [Mas69] ；用于 RS 码的模方法和快速傅里叶变换（快速傅里叶变换）[TH22]; 以及用于代数几何（AG）码的多数表决法 [FR93] 。在启动解码过程之前，必须检查收到的码字是否包含错误。如果未检测到错误，就无需解码。如果发现错误，则可以采用重传系统等机制，尽可能减少调用解码器的次数。因此，在实际应用中，快速错误检测至关重要。此外，一些码具有极高效的成员资格测试，这类码称为局部可测试码（LTC）；它们还与概率可检验证明（PCP）有着内在联系。

以往的研究主要集中于 RS 码的验证和 LTC 的研究。由于 RS 码可以看作在适当子集上对多项式求值，因此验证 RS 码本质上等同于从接收向量中恢复系数。用于此目的的主要技术是有限域上的快速傅里叶变换技术。有关乘法FFT、加法FFT、分圆FFT 和基于伽罗瓦理论的FFT（G-FFT）的更多细节，请参阅 [LTC 07] [WYL12] [BFT11][TH22] [LX24] [GM10] [Pol71] 。LTC 也可以被快速检测。它们是配备概率测试器的纠错码，只需对接收向量进行少量查询，便可验证其成员关系

*: lmc3051@sjtu.edu.cn, †: lijiyou@sjtu.edu.cn

[GS06]。测试器应以概率 1 接受码字，并以显著概率拒绝那些与该码在汉明距离上相距较远的字[KV10]。一些知名的码，例如哈达玛码、二元 Reed-Muller（RM）码 [AKK+05]、近似正交码（包括 Dual-BCH、Goppa 码和代数几何码的迹子码） [KL05], ，都是 LTC。然而，许多其他码并不是 LTC，例如随机码和某些类型的代数几何码 [Che09] [KV10]。

尽管大量研究集中于针对特定线性码的快速验证算法，但如何加速一般线性码的码字验证仍未得到充分研究。如果线性码缺乏丰富的代数结构，码字验证有时可能会成为计算瓶颈。一个典型例子是 RM码的列表译码算法。2004 年，Pellikaan 和 Wu [PW04] 将 RM码视为 RS 码的子域子码，并提出了一种两步列表译码方法：第一步调用 RS 码列表译码器，第二步检查所得列表中的每个码字，以确定其是否属于 RM码。尽管 Cohn 和 Heninger[CH15] 于 2015 年提出的算法在一定假设下使第一步达到了准线性时间复杂度，但第二步仍需要计算伴随式，这需要$O(n^2)$时间。最近，Kuang 等人[KLL+25] 通过为 RM码引入一种新的基，将第二步优化到了准线性时间。其算法的关键在于，新基与标准基之间的变换具有准线性复杂度。

在陈述主要问题之前，我们先介绍一些记号。设 C 是有限域 $\mathbb{F}_q$ 上的任意 $[n,k]$线性码，并设 H 为其校验矩阵。假设收到向量 $\mathbf{x} \in \mathbb{F}_q^n$，我们需要尽快确定$\mathbf{x} \in C$。显然，我们不能期望在准线性时间内直接计算 $H\mathbf{x}^T$，因为这将意味着矩阵乘法复杂度的改进。相反，我们旨在设计一种概率算法，其思想与 LTC 背后的思路类似。下面简要分析该算法所需的参数。这里，所有操作都是有限域 $\mathbb{F}_q$ 上的乘法或加法，存储空间以 $\mathbb{F}_q$中元素的数量衡量。该算法的首要目标是在 $O(n\log\ n)$ 时间内产生结果，因为直接计算 $H\mathbf{x}^T$需要 $O(n^2)$ 次操作。其次，由于校验矩阵需要存储 $n(n-k)$ 个元素（在系统形式情形下为 $k(n-k)$ 个元素），因此算法的存储空间必须限制在 $O(n^2)$ 以内。换言之，与标准方法相比，我们最多只能容忍存储空间增加一个常数倍。最后，我们希望算法的错误概率被限制在 $1/\text{poly}(n)$ 以内。如果错误概率过高，该算法的实际用途将十分有限。由此，我们得到以下核心问题：

问题 1.1。是否存在一种概率算法，能够在$\mathbf{x} \in C$ *in* $O(n\log\ n)$ 时间内判定，使用 $O(n^2)$ 空间，且错误概率不超过 $1/\text{poly}(n)$？

设$\mathbf{h}_1$、.. .、$\mathbf{h}_{n-k}$表示校验矩阵 H的行向量。要证明$\mathbf{x} \notin C$，只需找到一个与$\mathbf{x}$不正交的行向量$\mathbf{h}_i$。一种自然的

方法是从行向量$\mathbf{h}_1$、. . .、$\mathbf{h}_{n-k}$中随机抽取一个，计算它与$\mathbf{x}$的内积，并力求快速识别出这样的非正交向量。然而，这种直接的策略通常效果并不好。由于 $n > n-k$ 和 H 具有列满秩，存在一个非零向量$\mathbf{e}$，使得 $H\mathbf{e}^T$ 的重量等于 $\mathbf{1}$。设 $\mathbf{x} = \mathbf{c} + \mathbf{e}$，其中$\mathbf{c} \in C$为某个向量。那么，选中一个与$\mathbf{x}$ 不正交的行向量的概率仅为 $1/(n-k)$。平均而言，我们必须独立执行 $n-k$ 次内积操作才能检测出 $\mathbf{e}$，与直接计算伴随式相比并没有改进。在系统形式情形下，情况会变得更糟。假设 H 以分块形式 $[H_1 \mid I_{n-k}]$给出。即使要检测发生在最后 $n-k$ 个位置中的单个突发错误（对应于单位子矩阵），也需要计算所有内积 $(\mathbf{h}_1, \mathbf{x})$. . .、$(\mathbf{h}_{n-k}$、$\mathbf{x})$。

尽管这种朴素的随机化方法无法达到预期的效率，但它凸显了提高与$\mathbf{x}$不正交的向量所占比例的重要性。这一观察促使我们在下一节中介绍相应的算法。

2 算法设计

校验矩阵的行向量集合的一个自然推广是下面的测试集概念。

定义 2.1。设 C 是 $\mathbb{F}_q^n$ 中的一个 $[n,k]$ 线性码，$\mathbb{F}_{q^u}$ 是 $\mathbb{F}_q$ 的 u 次扩域。S 的子集 $\mathbb{F}_{q^u}^n$ 是码 C 的测试集，其设计概率为 $p \in$（0，1），如果对每个$\mathbf{x} \in \mathbb{F}_q^n$:

1. $\mathbf{x} \in C$ 当且仅当对所有$\mathbf{y} \in S$，（$\mathbf{x}$，$\mathbf{y}$）$= 0$。
2. 如果存在$\mathbf{y} \in S$ 使得（$\mathbf{x}$，$\mathbf{y}$）$\neq 0$，则

$$\sharp\{\mathbf{y} \in S \mid (\mathbf{x}, \mathbf{y}) \neq 0\} \geq (1-p)\sharp S. \tag{1}$$

注意，第二个条件等价于 $\sharp\{\mathbf{y} \in S \mid (\mathbf{x}, \mathbf{y}) = 0\} \leq p\sharp S$。测试集 S 与原码 C 具有如下关系。

命题 2.2. 如果 $C_S \subset \mathbb{F}_{q^u}^n$ 是由 S中的向量生成的线性码，则有 $C^\perp = \mathrm{Tr}(C_S)$，其中 $\mathrm{Tr}(C_S)$ 表示 C_S的迹码。

证明。首先，根据 S 的第一条性质，对于所有 $\mathbf{x} \in C_S^\perp$，都有这一结论；这意味着 $C \subset C_S^\perp \cap \mathbb{F}_q^n$；对于所有 $\mathbf{z} \in C_S^\perp \cap \mathbb{F}_q^n$，我们再次使用第一条性质，得到 $\mathbf{z} \in C$。最终得到$C = C_S^\perp \cap \mathbb{F}_q^n = C_S^\perp|_{\mathbb{F}_q^n}$。现在使用 Delsarte定理，得到 $\left(C_S^\perp|_{\mathbb{F}_q^n}\right)^\perp = \mathrm{Tr}\left((C_S^\perp)^\perp\right)$。由此可推出 $C^\perp = \mathrm{Tr}(C_S)$。 □

利用测试集的概念，我们提出以下通用算法框架。测试集 S 可以预先计算，并假定在收到向量 $\mathbf{x}$ 时可用。

算法 1 DetermineCodeword

输入： $\mathbf{x} \in \mathbb{F}^n{}_q$

输出： 真或假

1: **对于** $i = 1$ 到 R，**执行**
2: 从 S中随机选择一个向量$\mathbf{y}$
3: 计算内积 $z = (\mathbf{x}, \mathbf{y})$
4: **如果** $z \neq 0$ **则**
5: **返回** False
6: **结束如果**
7: **结束循环**
8: **返回**True

命题 2.3。 算法 *1* 可以在 $O(Run)$ 次操作内，使用 $O((\sharp S)un)$ 空间，以至多为 p^R 的错误概率解决判定问题“$\mathbf{x} \in C$”。

证明。 由于该算法每轮只执行一次内积操作，并且注意到 $\mathbb{F}_{q^u}$ 中的元素可以视为 $\mathbb{F}_q$上的多项式，因此，计算 $\mathbb{F}_{q^u}^n$ 和 $\mathbb{F}_q^n$ 中向量的内积不涉及 $\mathbb{F}_{q^u}$中的乘法操作。因此，算法每轮执行$O(un)$次操作。最多执行 R 轮时，操作总数至多为 $O(Run)$。集合 S 总共包含来自 $\mathbb{F}_{q^u}$的$(\sharp S)n$ 个元素。因此，存储 S 需要 $O((\sharp S)nu)$个 $\mathbb{F}_q$ 元素的空间。最后，由于与$\mathbf{x}$ 正交的向量比例至多为 p，当$\mathbf{x} \notin C$时，算法每轮的失败概率至多为 p 。因此，经过 R 次独立重复后，错误概率至多为 p^R。当$\mathbf{x} \in C$时，算法不会出错。

□

现在我们给出测试集 S的一种显式构造方法。该方法本质上是通过对伴随式进行编码来提高非正交向量的比例。设 $H \in \mathbb{F}_q^{(n-k)\times n}$为 C的校验矩阵， $\bar{G}$ 为 $\mathbb{F}_{q^u}^{\bar{n}}$ 上的一个 $[\bar{n}, n-k, \bar{d}]$ 码的生成矩阵。设 S为 $\bar{G}^T H$的所有行向量组成的集合，则有

$$\mathbf{x} \in C \Leftrightarrow H\mathbf{x}^T = \mathbf{0} \Leftrightarrow \bar{G}^T H\mathbf{x}^T = \mathbf{0}$$

and

$$\frac{\sharp\{\mathbf{y} \in S \mid (\mathbf{x}, \mathbf{y}) \neq 0\}}{\sharp S} = \frac{\mathrm{wt}(\bar{G}^T H\mathbf{x}^T)}{\bar{n}} \geq \frac{\bar{d}}{\bar{n}}. \tag{2}$$

因此， S是一个设计概率为 $1 - d/\bar{n}$的测试集。这种构造 S 的方法效率很高，因为预先计算 S 只需要进行一次矩阵乘法。通过仔细选择参数 $\bar{n}$ 和 d，算法 **1** 可以显著提高

验证效率。通过选择不同类型的 G，可以得到两种不同类型的 S，下文将对此进行说明。

$\bar{G}$ 源自 MDS码 对于满足 $\frac{n-k-1}{q^u-1} < 1$ 的固定 u，在范围 $\left(\frac{n-k-1}{q^u-1}, 1\right)$ 内任意选择 p。令 $m = \lceil (n-k-1)/p \rceil$，并在 $\mathbb{F}_{q^u}$ 上构造参数为 $[m, n-k, m-(n-k)+1]$ 的MDS码。设 G 为其生成矩阵，则由 $\bar{G}^T H$ 的行向量构造的测试集 S 的设计概率为

$$1 - \frac{m-(n-k)+1}{m} = \frac{n-k-1}{\left\lceil \frac{n-k-1}{p} \right\rceil} \leq p$$

根据不等式 (1) 和 (2)。换言之，对于 $\left(\frac{n-k-1}{q^u-1}, 1\right)$ 中任意的 p，我们都可以构造一个设计概率为 p 的测试集。这就是我们将 p 称为设计概率的原因。

$\underline{G}$ 源自渐近良码 现在我们转而求解问题1.1，该问题要求从渐近良码中选择 G 作为生成矩阵。

定理 2.4. 设 $\{C_i\}$ 是 *over* $\mathbb{F}_{q^u}$ 上的渐近良好的 $[n_i, k_i, d_i]$ 码族，这意味着

$$\liminf_{i\to\infty} \frac{k_i}{n_i} \geq \alpha,\ \liminf_{i\to\infty} \frac{d_i}{n_i} \geq \beta,\ 0 < \alpha, \beta < 1.$$

我们进一步假设存在常数 W，使得对于所有 i均有 $n_i \leq W n_{i-1}$。通过使用 C_i 的生成矩阵构造良好的测试集，算法 *1* 可以在 $O(un\log\, n)$ 次操作内解决判定问题“$\mathbf{x} \stackrel{?}{\in} C$”，使用 $O(un^2)$ 空间，且错误概率不超过 $1/\mathrm{poly}(n)$。

证明. 由于 C_i 是渐近良好的，我们可以假设

$$\alpha - \epsilon \leq \frac{k_i}{n_i},\ \beta - \epsilon \leq \frac{d_i}{n_i} \tag{3}$$

对于所有 i，其中 ϵ 属于 $(0,\ \min\{\alpha, \beta\})$。给定任意 $[n, k]$ 码 C 及其校验矩阵 H，并假设对于某个 i有 $k_{i-1} < n-k \leq k_i$，$\underline{C_i}$ 的生成矩阵 G_i 形状为 $k_i \times n_i$。我们删除 G_i 的若干行，并将其转换为一个 $(n-k) \times n_i$ 矩阵 G_i'（当 $n-k = k_i$、$G_i' = G_i$ 时），并将测试集 S 定义为 $\bar{G}_i'^T H$ 的行向量集合。S 的设计概率 p 满足

$$p \leq 1 - \frac{d_i}{n_i} \leq 1 - \beta + \epsilon < 1$$

根据 (1) 和 (3)。取轮数为 $R = R_1 \log n$，其中 R_1 是一个常数，其值将很快确定，经过 R 轮后的错误概率为

$$p^R \leq (1 - \beta + \epsilon)^{R_1 \log n} = \frac{1}{n^{R_1 \log(1-\beta+\epsilon)^{-1}}},$$

其中 $\log(1 - \beta + \epsilon)^{-1} > 0$。因此，通过适当增大常数 R_1，使得 $R_1 \log(1 - \beta - \epsilon)^{-1}$ 大于 $\deg \text{poly}(n)$，可以将错误概率界定为 $1/\text{poly}(n)$。显然，操作总数为 $O(unR_1 \log n) = O(un \log n)$。最后，我们证明空间复杂度为 $O((\sharp S)un) = O(un^2)$，方法是

$$\sharp S = n_i \leq W n_{i-1} \leq W \frac{k_{i-1}}{\alpha - \epsilon} < \frac{W}{\alpha - \epsilon}(n - k).$$

□

值得注意的是，C_i 可以利用 Garcia-Stichtenoth（GS）塔构造。关于 C_i 的更多细节，可参见 [GS95] 以及 [Sti09]的第 7 章和第 8 章。GS 塔要求有限域的大小为平方数，因此我们选择 $u = 2$。这样，该算法将满足问题 1.1 中的所有条件。

我们的算法高度依赖于从 S 中随机选取 $O(\log n)$ 个向量。这种随机性保证了 $O(n \log n)$ 的时间复杂度以及较低的错误概率。对于一般的线性码 C 和任意向量 $\mathbf{x}$，显然，我们无法仅通过测试 $\mathbf{x}$ 与 C 中 $O(\log n)$ 个向量的正交性，来判断$\mathbf{x}$ 是否属于 C。因此，将这一过程去随机化以获得确定性算法，是一个有趣且富有挑战性的问题。

问题 2.5. 是否存在一种确定性算法，能够在 $O(n \log n)$ 时间内解决任意线性码的码字判定问题?

3 应用

本节介绍我们算法的一些应用。

3.1 RM码的列表译码

第一个应用是 RM码的列表译码。我们沿用 Wu 的思路 [PW04]，将 RM码视为 RS 码的子域子码。

引理 *3.1*（[KLL+25]）。假设 $r = u(q-1) + s$，其中 $0 \leq s < q - 1$。令$k = q^m - q^{m-u-1}(q - s) + 1$。长度为 $N = q^m$ 的 *Reed-Muller*码 $RM_q(r, m)$ 可视为 $RS_{q^m}(q^m, k)$ 的子域子码。此外，$RM_q(r, m)$ 与 $RS_{q^m}(q^m, k)$ 具有相同的最小距离 $D = q^{m-u-1}(q - s)$。

为了有效地对 $RS_{q^m}(q^m, k)$ 进行列表译码，我们还需要 [CH15] 中提出的 RS列表译码器，它可实现准线性复杂度。

引理 3.2 ([CH15] [KLL+25])。一个 q 元 $[n,k]$ *Reed-Solomon* 码 $\mathrm{RS}(n,k)$ 是(ρ,l)-列表可译码的，其译码半径 ρ 和列表大小 l 定义如下：

$$\rho = 1 - \sqrt{1-\delta} - \epsilon,\ l = O(1/\epsilon),$$

其中 $\epsilon > 0$是一个较小的实数，且 $\delta = (n-k+1)/n$。解码复杂度为 $O(\epsilon^{-\omega-1}\log^2 n \log\log n)$，其中 ω 是矩阵乘法的指数。

对 [KLL+25] 中算法 2 稍作修改，可得到以下改进后的算法。

算法 2 RM 列表译码器(ρ, l)

输入： $\mathbf{r} \in \mathbb{F}_q^N$ **输出：** $L = \{\mathbf{a} \in \mathbb{F}_q^N \mid \mathrm{wt}(\mathbf{a} - \mathbf{r}) \le \rho N\}$1: 置 $L = \emptyset$2: 调用 RS-list-decoder(ρ, l)，输入为 $\mathbf{r}$。令 L' 为其输出。3: **对于** $\mathbf{c} \in L' \cap \mathbb{F}_q^N$，**执行**4: **如果** $\mathrm{DetermineCodeword}(\mathbf{c})$**成立，则**5: $L = L \cup \{\mathbf{c}\}$6: **结束 if**7: **结束 for**8: **返回** L

同样，步骤 2 将需要 $O(\mathrm{M}_q(m) \cdot \frac{1}{\epsilon^{\omega+1}} N \log^2 N \log\log N)$ 次操作，其中 $\mathrm{M}_q(m)$ 表示在 $\mathbb{F}_q$ 上将两个次数小于 m 的一元多项式相乘所需的操作次数。根据定理 2.4，步骤 4 需要 $O(N \log N)$ 次操作，错误概率不超过 $1/\mathrm{poly}(N)$。因此，输出列表 L 包含非 Reed-Muller 码字的概率小于 $1 - (1 - 1/\mathrm{poly(N)})^l \sim l/\mathrm{poly}(n)$。

3.2 与 RS 码专用算法的比较

在本小节中，我们将我们的算法与其他确定性伴随式计算算法进行比较。这些算法专为 RS 码设计，主要基于快速傅里叶变换技术 [LTC07] [WYL12] [BFT11] [TH22]。对于 RS(255，223) 和 RS(1023，895)，我们分别使用 RS(255，32) 和 RS(1023，128) 对伴随式进行编码，因此设计概率分别为 $31/255$ 和 $127/1023$。进行 7 轮独立试验后，错误概率小于 4×10^{-7} 和4.6×10^{-7}。可以看出，在这些设置下，对于 RS 码字验证问题，我们的算法与专用算法的表现相当。

表1：与专用伴随式计算算法的复杂度比较

	RS(255，223)		RS(1023，895)	
	Mul.	Add.	Mul.	Add.
方法 in [LTC07]	3,060	4,998	33,620	73,185
方法 [WYL12]	252	3,064	2,868	19,339
[BFT11]中的方法	149	2,931	824	36,981
[TH22]中的方法	752	1,696	4,160	9,088
我们的方法	1,785	1,778	7,161	7,154

参考文献

[AKK+05]N. Alon、T. Kaufman、M. Krivelevich、S. Litsyn 和 D. Ron。Reed–Muller码测试。*IEEE*《信息论汇刊》，51(11):4032–4039，2005。[BFT11] S. Bellini、M. Ferrari 和 A. Tomasoni。分圆傅里叶变换及其在 Reed–Solomon码中的应用。*IEEE*《通信汇刊》，59(8):2110–2118，2011。[CH15] Henry Cohn 和 Nadia Heninger。Coppersmith定理和 Guruswami–Sudan列表译码的理想形式。《通信数学进展》，9(3):311–339，2015。[Che09] H. Chen。代数几何码测试。《中国科学：数学》，52(10):2171–2176，2009。[FR93] G.-L. Feng 和 T.R.N. Rao。译码代数几何码至设计最小距离。*IEEE*《信息论汇刊》，39(1):37–45，1993。[GM10] Shuhong Gao 和 Todd Mateer。有限域上的加法快速傅里叶变换。*IEEE*《信息论汇刊》，56(12):6265– 6272，2010。[GS95] Arnaldo Garcia 和 Henning Stichtenoth。一个达到 Drinfeld–Vladut 界的函数域 Artin–Schreier扩张塔。《数学新进展》，121(1):211–222，1995年12月。[GS06] Oded Goldreich 和 Madhu Sudan。局部可测试码和近线性长度的概率可检验证明。《*ACM*学报》，53(4):558–655，2006年7月。[KL05] T. Kaufman 和 S. Litsyn。近乎正交线性码是局部可测试的。载于第*46*届*IEEE*计算机科学基础年会（*FOCS*），第317–326页，2005年。

[KLL+25] Yi Kuang、Songsong Li、Shu Liu、Liming Ma 和 Chaoping Xing。具有准线性复杂度的 Reed–Muller 码编码与译码。*IEEE*《信息论汇刊》，71(12):9330– 9339，2025。[KV10] Tali Kaufman 和 Michael Viderman。局部可测试码与局部可解码码。在 Maria Serna、Ronen Shaltiel、Klaus J ansen 和 Jos´e Rolim 编，近似、随机化与组合优化：算法与技术，第 670–682 页，柏林、海德堡，2010。Springer Berlin Heidelberg。[LTC07] Tsung-Ching Lin、T. K. Truong 和 P. D. Chen。用于 Reed–Solomon 码代数译码中伴随式计算的快速算法。*IEEE*《通信汇刊》，55(12):2240–2244，2007。[LX24] Songsong Li 和 Chaoping Xing。通过有理函数域自同构群实现快速傅里叶变换。在*2024*年*ACM-SIAM*离散算法年会（*SODA*）论文集，第 3836–3859 页，2024。[Mas69] J. Massey。移位寄存器综合与 BCH 译码。*IEEE*《信息论汇刊》，15(1):122–127，1969。[Pol71] J. M. Pollard。有限域上的快速傅里叶变换。计算数学，25(114):365–374，1971。[PW04] R. Pellikaan 和 Xin-Wen Wu。q 元 Reed–Muller 码的列表译码。*IEEE*《信息论汇刊》，50(4):679–682，2004。[Sti09] Henning Stichtenoth。代数函数域与编码，第 254 卷，数学研究生教材。Springer，柏林、海德堡，第 2 版，2009。[TH22] Nianqi Tang 和 Yunghsiang S. Han。基于快速傅里叶变换和模方法的 Reed–Solomon 码新译码方法。*IEEE*《通信汇刊》，70(12):7790–7801，2022。[WYL12] Xuebin Wu、Zhiyuan Yan 和 Jun Lin。基于复合分圆傅里叶变换的长 Reed–Solomon 码低复杂度译码器。*IEEE* 信号处理汇刊，60(7):3920–3925，2012。