BCH码

维基百科，自由的百科全书

BCH码（BCH codes、Bose–Chaudhuri–Hocquenghem codes）为取自Bose、Ray-Chaudhuri与Hocquenghem的缩写，是编码理论尤其是纠错码中研究得比较多的一种编码方法。用术语来说，BCH码是用于校正多个随机错误模式的多级、循环、错误校正、变长数字编码。BCH码也可以用于质数级或者质数的幂级的多级相移键控。11级的BCH码已经用于表示10进制数外加一个符号位。

目录

构建

BCH 码使用有限域上的域论与多项式。为了检测错误可以构建一个检测多项式，这样接收端就可以检测是否有错误发生。

要构建一个能够检测、校正两个错误的 BCH 码，我们要使用有限域 GF(16) 或者 $\mathbf{Z}_2[x] / \langle x^4 + x + 1 \rangle$。如果 α 是 $m_1(x) = x^4 + x + 1$ 的一个根，那么 m_1 就是 α 的极小多项式，这是因为

$$m_1(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4)(x - \alpha^8) = x^4 + x + 1。$$

如果要构建一个能够纠正一个错误的 BCH 码，那么就使用 $m_1(x)$，这个代码就是所有满足

$C(x) \equiv 0 \pmod{m_1(x)}$ 且根为 $\alpha, \alpha^2, \alpha^4, \alpha^8$ 的多项式 $C(x)$。

编码

构建码字为

$$(c_{14}, c_{13}, \ldots, c_8)$$

这样多项式为

$$c_{14} + c_{13} + \ldots + c_8$$

我们将它称为 C_I。

然后就要找出 C_R 满足 $C_R=C_I \pmod{m_{1,3}(x)}=c_7+c_6+...+c_0$

这样就得到待发的码字 $C(x) = C_I+C_R \pmod{m_{1,3}(x)} = 0$

例如，如果我们要对 (1,1,0,0,1,1,0) 进行编码

$$C_I=x^{14}+x^{13}+x^{10}+x^9$$

然后用 $m_{1,3}(x)$ 除以（这里的除法是多项式除法）C_I，得到结果为 $C_R(x)$，在$\mathbf{Z}_2$域中，我们可以算出 C_R为

$$x^3+1$$

这样，待发的码字为

(1,1,0,0,1,1,0, 0,0,0,0,1,0,0,1)

解码

BCH 的解码过程可以分为以下四步

1. 计算接收到的向量 R 的 2t 伴随矩阵
2. 计算错误定位多项式
3. 解多项式，得到错误位置
4. 如果不是二进制 BCH 码，就计算错误位置的误差值

假设我们收到一个码字向量 **r**，即多项式 $R(x)$）。

如果没有错误，那么 $R(\alpha)=R(\alpha^3)=0$

如果有一个错误，例如 $\mathbf{r}=\mathbf{c}+\mathbf{e}_i$，其中 $\mathbf{e}_i$ 表示 $\mathbf{R}^{14}$ 的第 i个基向量 于是

$$S_1=R(\alpha)=C(\alpha)+\alpha^i=\alpha^i$$
$$S_3=R(\alpha^3)=C(\alpha^3)+(\alpha^3)^i$$
$$=(\alpha^i)^3=S_1^3$$

这样就可以纠正错误。α 的指数显示的数据位变化可以帮助我们校正错误。

如果有两个错误

$$\mathbf{r}=\mathbf{c}+\mathbf{e}_i+\mathbf{e}_j$$

那么

$$S_1=R(\alpha)=C(\alpha)+\alpha^i+\alpha^j$$
$$S_3=R(\alpha^3)=C(\alpha^3)+(\alpha^3)^i+(\alpha^3)^j$$
$$=(\alpha^3)^i+(\alpha^3)^j$$

这与 S_1^3 不同，所以我们认为有两个错误。更进一步的代数方法可以帮助校正着两个错误。

开头两段内容出自 Federal Standard 1037C

上面的文字摘自：https://web.archive.org/web/20070213013106/http://bch-code.foosquare.com/

BCH 解码算法

流行的解码算法有，

1. Peterson Gorenstein Zierler算法
2. Berlekamp-Massey算法

Peterson Gorenstein Zierler 算法

假设

Peterson 算法是普通 BCH 解码过程的第二步，在这里使用 Peterson 算法计算多项式 $\Lambda(x) = 1 + \lambda_1 X + \lambda_2 X^2 + \ldots + \lambda_{2t} X^{2t}$ 的错误定位多项式系数 $\lambda_1, \lambda_2 \ldots \lambda_{2t}$

这样给定 BCH 码 (n, k, d_{min})，可以校正 $[t = \frac{d_{min} - 1}{2}]$ 个错误的 Peterson Gorenstein Zierler 算法就是

算法

- 首先生成 $2t$ 伴随矩阵
- 然后生成元素为

$$S_{t\times t} = \begin{bmatrix} s_1 & s_2 & s_3 & \ldots & s_t \\ s_2 & s_3 & s_4 \ldots & \ldots & s_{t+1} \\ s_3 & s_4 & s_5 & \ldots & s_{t+2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ s_t & s_{t+1} & s_{t+2} & \ldots & s_{2t-1} \end{bmatrix}$$ 的矩阵 S_{txt}

- 生成元素为

$$C_{t\times 1} = \begin{bmatrix} s_{t+1} \\ s_{t+2} \\ \ldots \\ \ldots \\ s_{2t} \end{bmatrix}$$ 的矩阵 c_{tx1}

- 让 Λ 表示未知的多项式系数，用

$$\Lambda_{t\times 1} = \begin{bmatrix} \lambda_t \\ \lambda_{t-1} \\ \ldots \\ \lambda_3 \\ \lambda_2 \\ \lambda_1 \end{bmatrix}$$ 表示

- 这样就得到矩阵方程

$$S_{t\times t}\Lambda_{t\times 1} = C_{t\times 1}$$

- 如果矩阵 $S_{t\times t}$ 存在行列式，那么我们就可以找到这个矩阵的逆，然后就可以得到 Λ 的值
- 如果 $det(S_{t\times t}) = 0$，那么按照

```
if t = 0
then
      declare an empty error locator polynomial
      stop Peterson procedure.
end
set t ← t − 1
continue from the beginning of Peterson's decoding
```

- 在 Λ 的值确定之后，自然就得到错误定位多项式
- 结束 Peterson procedure。

错误定位多项式的因式分解

在得到 $\Lambda(x)$ 多项式之后，用 *Chiens search* 算法就可以得到它的解 $\Lambda(x) = (\alpha^i X + 1)(\alpha^j X + 1)\dots(\alpha^k X + 1)$ 。根据素元 α 的指数幂就能得到接收到的码字中错误的位置，这也就是误差定位多项式名称的由来。

错误校正

对于二进制的BCH码，可以直接根据错误定位多项式因数素元指数的位置校正接收到的向量。最后，对这些位置接收到的数值取反，就可以得到正确的BCH解码码字。

另外也可以使用Berlekamp-Massey 算法确定错误定位多项式，从而解决BCH解码的问题。

参考文献

主要参考

- Hocquenghem, A., Codes correcteurs d'erreurs, Chiffres (Paris), September 1959, **2**: 147–156 **(法语)**
- Bose, R. C.; Ray-Chaudhuri, D. K., On A Class of Error Correcting Binary Group Codes, Information and Control, March 1960, **3** (1): 68–79, ISSN 0890-5401, doi:10.1016/s0019-9958(60)90287-4

次要参考

- Gill, John, EE387 Notes #7, Handout #28 (PDF), Stanford University: 42–45, n.d. [April 21, 2010], （原始内容 (PDF)存档于2014年6月30日） Course notes are apparently being redone for 2012: http://www.stanford.edu/class/ee387/（页面存档备份 (https://web.archive.org/web/20130605170343/http://www.stanford.edu/class/ee387/)，存于互联网档案馆）
- Gorenstein, Daniel; Peterson, W. Wesley; Zierler, Neal, Two-Error Correcting Bose-Chaudhuri Codes are Quasi-Perfect, Information and Control, 1960, **3** (3): 291–294, doi:10.1016/s0019-9958(60)90877-9
- Lidl, Rudolf; Pilz, Günter, Applied Abstract Algebra 2nd, John Wiley, 1999
- Reed, Irving S.; Chen, Xuemin, Error-Control Coding for Data Networks, Boston, MA: Kluwer Academic Publishers, 1999, ISBN 0-7923-8528-4

延伸阅读

- Blahut, Richard E., Algebraic Codes for Data Transmission 2nd, Cambridge University Press, 2003, ISBN 0-521-55374-1
- Gilbert, W. J.; Nicholson, W. K., Modern Algebra with Applications 2nd, John Wiley, 2004
- Lin, S.; Costello, D., Error Control Coding: Fundamentals and Applications, Englewood Cliffs, NJ: Prentice-Hall, 2004
- MacWilliams, F. J.; Sloane, N. J. A., The Theory of Error-Correcting Codes, New York, NY: North-Holland Publishing Company, 1977
- Rudra, Atri, CSE 545, Error Correcting Codes: Combinatorics, Algorithms and Applications, University at Buffalo, [April 21, 2010], （原始内容存档于2010-07-02）

外部连接参考文献

- S. Lin and D. Costello. Error Control Coding: Fundamentals and Applications. Prentice-Hall, Englewood Cliffs, NJ, 2004.
- Step by step decoding instructions (https://web.archive.org/web/20070107163035/http://ietfec.oxfordjournals.org/cgi/reprint/E88-A/8/2236.pdf) (pdf file).
- **Federal Standard 1037c:** https://web.archive.org/web/20060820191527/http://federal-standard-1037c.foosquare.com/
- Galois Field Calculator: https://web.archive.org/web/20060212215733/http://www.geocities.com/myopic_stargazer/gf_calc.zip
- Decoding Algorithms for BCH codes: http://students.uta.edu/mx/mxa6471/research/ecc-report.pdf
- Source code for BCH channel simulation: http://students.uta.edu/mx/mxa6471/research/ecc-code.tgz