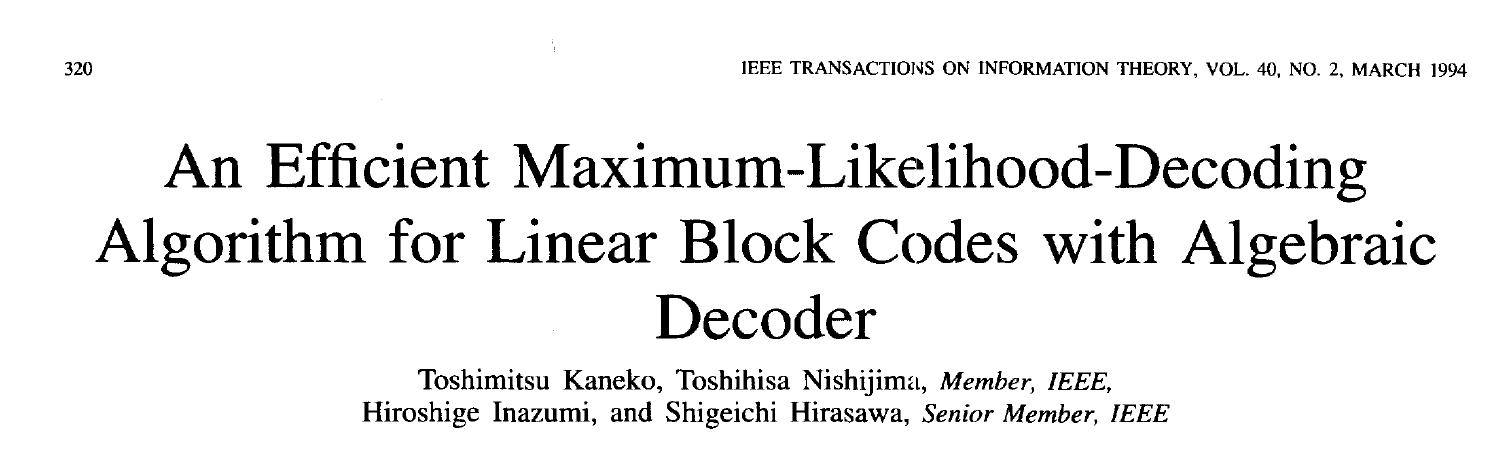
**ML算法描述**

1. **文献1**



1. **文献1逻辑**
2. 先找到码字为最大似然时的充分条件。
3. 找到生成最小的足以达到最大似然的码字集合的方法（找判断准则）。
4. 给出具体的ML算法，并作性能分析。
5. **内容笔记**
6. 分析码字最大似然充分条件时，设置了4个码字：硬判决出的y, 任意两个不同的码字x与x`, 用代数译码器译出来的码字x0，然后比较他们中码字数量，即势函数（这里假设都是二进制，后面再推广至多元）与该码最小汉明距离d的关系。实际分析时，以y为参考系。看待4种码字时，其交集部分的势函数小于d，不相同部分的势函数大于或等于d。

具体可以画出如以下图来做比较（这只是1种case，可能还存在其他case，但不影响结论），四个码字分别是y, x, x`, x0：

y

x

m

h1

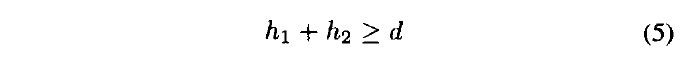
x`

h2

x0

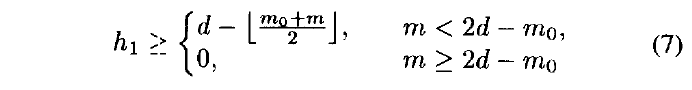
m0

可根据类似上图获得以下关系：

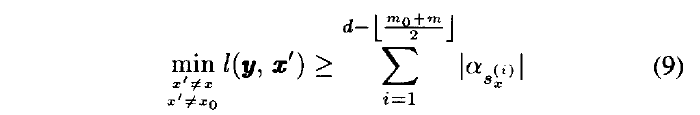


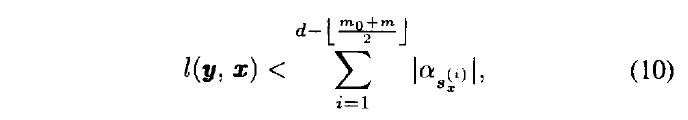


由此可得：



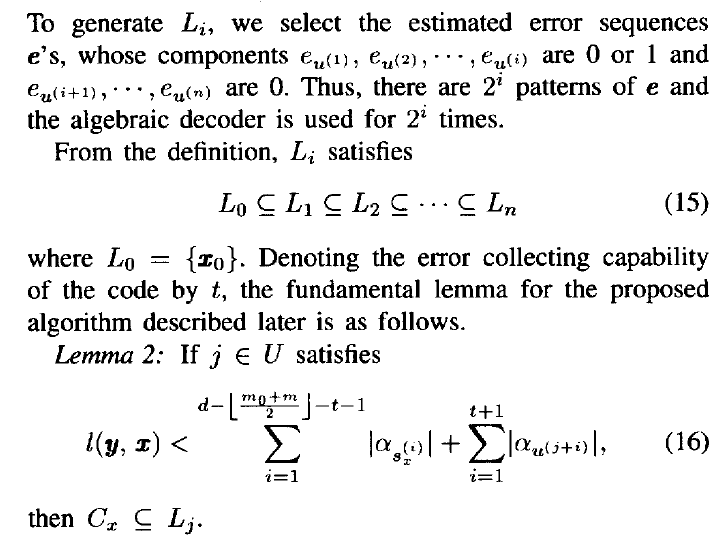
因此，有以下关系，那个累加项即是一个分界：



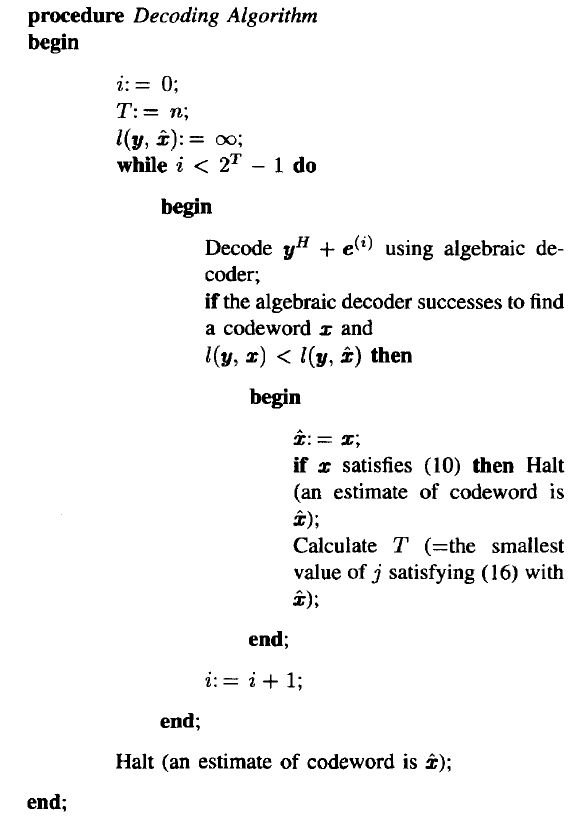


上式(10)即判断码字达到最大似然的的准则，即码字x符合此准则时，不会有其他码字比它的似然测度更低。

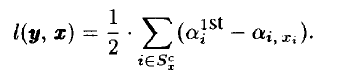
1. 判断最小的足以达到最大似然的码字集合的准则：



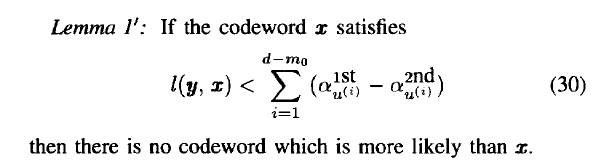
1. 译码算法，实际上更关心ML判决准则多于此算法：



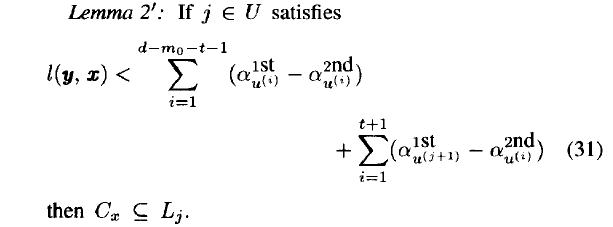
1. 多元码时的推广时，逻辑与二元时是类似的，这时测度函数变为：



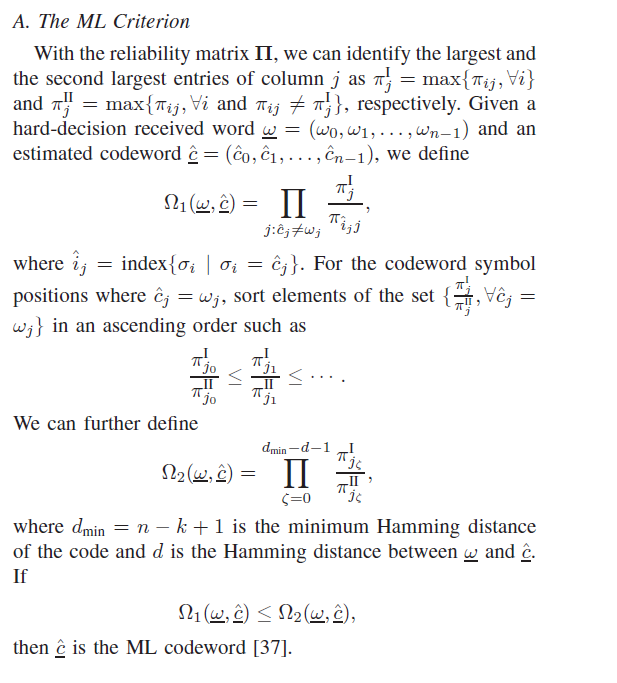
最大似然的充分条件变为：



判决准则变为：

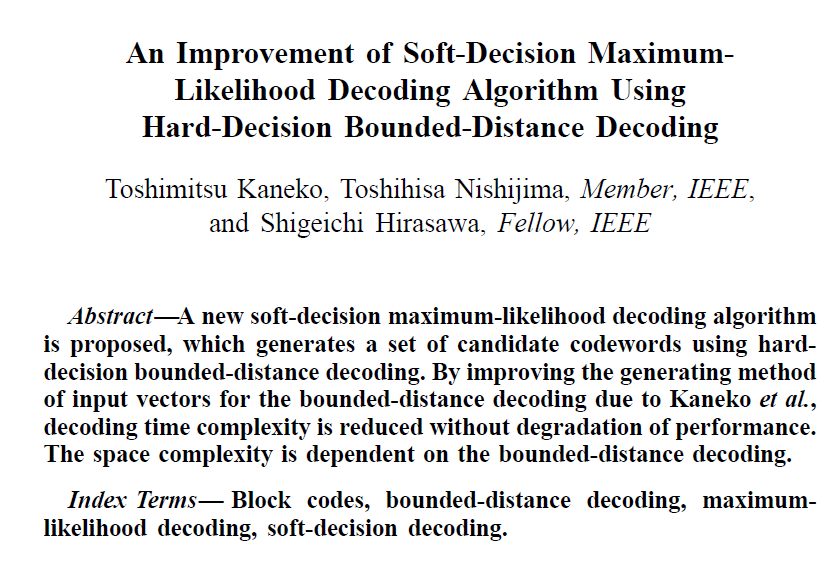


实际应用时，算法描述如下，注意，此时未用对数运算所以减法变除法，判决准则直接利用了上述Lemma1，即对不上部分的测度小于（一些，数量取决于最小码距和汉明距离）对的上部分的测度：



但需要注意的是，使用时用到了全排序，虽然对于一次接收，可以做一些优化只做一次（大）部分排序，但这仍是一件麻烦事。

1. **文献2**



1. **文献2逻辑**

文献2所用到的最大似然判决准则与文献1一致，只是译码逻辑上有所不同，减少了迭代与搜索的复杂度。