第7卷 第6期 电子科学学刊 Vol. 7, No. 6
1985 年 11 月 JOURNAL OF ELECTRONICS Nov., 1985

用查表法提高分类器速度*

郑勤奋 吴健康 王文涛
(中国科学技术大学)

1. 引言

在使用地球资源卫星进行农、林、矿等各种资源调查中，需要处理极其大量的数据.对数据分类是各种处理中最费机时的操作. 提高分类器的速度将可缩短处理周期，降低处理费用. 因此寻找快速算法一直是人们努力解决的课题之一.

利用微型机处理数据时，运行时间长的问题尤其严重，这大大限制了遥感技术的推广和普及.作者在遥感应用软件的研究中，利用 MSS 数据是整型，且动态范围较窄的特点，采用查表技术，并通过增加一些计算机完全容许增加的内存容量，大幅度地提高了分类速度. 以 ML 分类器为例，实验表明: (1)在进行二维特征分类时，使用查表法可以把分类时间减少到同磁盘数据的读写时间同一个数量级，即仅为传统方法的百分之几; (2)在高维特征情况下，使用增量查表技术可以节约分类时间一半以上. 本文通过对 ML 分类器的简单介绍引出了查表分类器，并进一步引出了增量查表分类器，然后介绍了算法的数学原理，软件的实现流程和实验的结果. 全部实验是在 CROMEMCO 3 微型机上进行的. 该机内存为 64kB，没有专用乘法器. 最后讨论了将查表技术用于其它几种统计分类器的设计方法和达到的性能.

2. ML 分类器简介

对于N类M维数据的分类问题，根据最大似然 (Maximum Likelihood, 以下简称 ML) 分类准则，假定数据满足M维正态分布:

$$P(X/\omega_i)=\frac{1}{(2\pi)^{M/2}|\Sigma_i|^{1/2}}\exp\left[-\frac{1}{2}(X-M_i)^T\Sigma_i^{-1}(X-M_i)\right], \tag{1}$$

式中: $X\in\omega_i$ 为一M维矢量，表示一个样本; ω_i 为一特定的地类; Σ_i 为第 i 类的协方差矩阵，$\Sigma_i=(\sigma_{ikj})_{M\times M}$; M_i 为第 i 类的均值矢量，$M_i=(m_{ik})_M$.

分类的目的是以最小差错率(最大可能性)判定输入样本属于哪一类. 运用概率论中最大后验概率的计算方法，代入 Bayes 公式，经数学推导可得 ML 分类方法的最终判决准则为

$$X\to\{\omega_k\},\ \text{当且仅当}\ f_k(X)=\max_{1\le j\le N} f_j(X); \tag{2}$$

式中 $f_j(X)=C_j-(X-M_j)^TA_j(X-M_j)$ 为第 j 类的似然函数，$A_j=\Sigma_j^{-1}$，可以事先求出; $C_j=2\ln P(\omega_j)-\ln|\Sigma_j|$，为常数项，$P(\omega_j)$ 为第 j 类的先验概率.

将上述分类准则用软件实现就得到 ML 分类器.

* 1984 年 10 月 17 日收到，1985 年 2 月 26 日修改定稿.

3. 查表 ML 分类器

从上述介绍可知，传统的 ML 分类器对每一样本都要逐类计算似然函数值，其中有大量重复运算，若能省去这些重复计算就可以提高分类速度．实际上 MSS 数据的分布较窄，在一幅图象中完全一样的象元重复出现的频率很高，只要让计算机具备“学习”功能，对重复出现的样本按“经验”判决就可以大大提高分类器的速度．具体做法是，采用查表技术，在程序中建立一个临时的“知识库”；分类时，对新输入的样本回忆一下以前是否已处理过相同的样本，若处理过，则不必重新计算似然函数值，直接按照以前的结论分类；反之则对新样本运用 ML 准则进行分类，并将结果充实进“知识库”．抽象地看，分类过程就是将输入样本映射到结果集合中的过程．传统的 ML 分类器是通过函数运算完成映射的，而查表算法则是通过查分类表完成映射的，显然后者的速度快得多．

下面是在森林分类研究中，应用查表分类器的两个例子．

实验 1　*对茅山林区的分类时间比较*　实验数据为 Landsat-II，南京幅茅山地区，图幅尺寸为 128 × 64；使用 5，7 两个波段数据进行分类，要求分为 12 类；不计训练区统计的时间，用传统 ML 分类器需时间 1519.8 秒，而采用查表 ML 分类器仅需时间 98.3 秒，节省时间 93.5%．

实验 2　*对南平林区的分类时间比较*　实验数据取自 Landsat-III，福建幅南平地区，图幅尺寸为 256 × 256；使用 5、7 两个波段数据，分为 37 类；采用传统 ML 分类器需时间 32900.7 秒，而采用查表分类器仅需时间 1407.5 秒，节省时间 95.7%；其中用于磁盘数据的读写时间占 319.0 秒，约为分类时间的四分之一．

应指出的是，查表 ML 分类器在所需的表容量和运行时间两方面都与要求的分类类数无关，这是这种方法的独特优点．同时，图象的尺寸越大，分类的类数越多，节省的分类时间就越多．

这里介绍的查表分类器是通过边分类边学习来建立分类表的．这样，每分类一个样本都需核实一下该样本是否已出现过，这一操作要占用不少时间．在应用中，如果要处理的图幅较大时，应先采用一次学习法建立分类表；在开始实际分类前，先计算好分类表，分类时只需查表即可；这样，可以大大节约运行时间．

4. 增量查表 ML 分类器

现在让我们来考察上述查表 ML 分类器对分类表的容量要求．设数据的维数为 M，数据在各维特征的变化范围为 L_i，$i = 1, 2, \cdots, M$；则分类表所占的内存容量为

$$B_s = B_t \cdot \left(\prod_{i=1}^{M} L_i \right), \tag{3}$$

B_t 为一个单元所需的字节数．由于应用上要求分的类数总少于 128，故取 $B_t = 1$ 就够了．为了方便，取 $L = \max\limits_{1 \leqslant i \leqslant M} L_i$，让分类表在各维上范围相同，则有 $B_s = L^M$．对于 Landsat MSS 数据，可以取 $L = 100$．由于 B_s 按特征维数 M 的指数规律增长，取 $M = 3$ 时，B_s 已接近 1000kB，所以这种直接查表法只适用于 $M \leqslant 2$ 的情形；对于高维特征的分类问题，受内存容量的限制，直接查表法已不适用，必须另找办法．从似然函数的计算式 $f_i(X) = C_i - (X - M_i)^T A_i (X - M_i)$ 可以算出，每求一次似然函数值需进行 $M(M + 1)$次乘法，$M(M + 2)$次加法．由此可见加法次数与乘法次数相当，若能通过查表减

少乘法次数，则可望提高分类器的速度．另一方面，由 B_s 的计算式知，若能设法降低样本数值的变化范围，就可以使表容量的增长减慢，而这可以利用数据的空间相关性来解决．由于 Landsat MSS 数据是地物光谱特性的反映，相邻点的灰度变化往往很小，只要经过变换，使查表按相邻点间的增量来进行，就可以使查表法适用于设计较高维特征的分类器．这就是增量查表法的基本出发点，下面给出其数学推导．

设对 X_1 有 $f_i(X_1)=C_i-(X_1-M_i)^TA_i(X_1-M_i)$，$i=1,2,\cdots,N$；则对 $X_2=X_1+\Delta X$ 有

$$\begin{aligned}f_i(X_2)&=f_i(X_1+\Delta X)=C_i-(X_1+\Delta X-M_i)^TA_i(X_1+\Delta X-M_i)\\&=C_i-[(X_1-M_i)+\Delta X]^TA_i[(X_1-M_i)+\Delta X]\\&=C_i-(X_1-M_i)^TA_i(X_1-M_i)-\Delta X^TA_i(X_1-M_i)\\&\quad-(X_1-M_i)^TA_i\Delta X-\Delta X^TA_i\Delta X\\&=f_i(X_1)-\Delta X^TA_i(X_1+X_2-2M_i),\quad i=1,2,\cdots,N.\end{aligned}\tag{4}$$

记 $\Delta X^T=\{a_1,a_2,\cdots,a_M\}=Z_1^T+\cdots+Z_M^T$，$Z_K^T=\{a_i\delta_{ik}\}_M$，$k=1,2,\cdots,M$，$\delta_{ik}=\begin{cases}1, & i=k;\\0, & i\neq k;\end{cases}$ 由(4)式有

$$\begin{aligned}f_i(X_1+\Delta X)&=f_i(X_1)-\left(\sum_{k=1}^{M}Z_k^T\right)A_i(X_1+X_2-2M_i)\\&=f_i(X_1)-\left[\sum_{k=1}^{M}(Z_k^TA_i)\right](X_1+X_2-2M_i).\end{aligned}\tag{5}$$

记 $A_i=\{\sigma_{ijk}\}_{\substack{1\leqslant j\leqslant M\\1\leqslant k\leqslant M}}$，则 $Z_k^TA_i=a_k(\sigma_{ik1},\cdots,\sigma_{ikM})$．分类时用公式(5)计算似然函数值，$Z_k^TA_i$ 则是通过查表得到．这样，不计查表所需的操作，分类一个样本的计算量为 $N\cdot M$ 次乘法，$N\cdot M(M+2)+2M$ 次加法．考虑到对于 CPU 字长为 8 位的微型机，用软件实现一次乘法所需的时间约为加法的 8 倍，可估计出用式(5)进行增量查表 ML 分类较传统 ML 分类器节省计算量的理论值为

$$\begin{aligned}\eta&=1-T_{\mathrm{Tab}}/T_{\mathrm{ML}}=1-\frac{8NM+NM(M+2)+2M}{8NM(M+1)+NM(M+2)}\\&=\frac{8NM-2}{9NM+10N}\doteq\frac{8M}{9M+10}\quad(\text{因 }NM\gg1).\end{aligned}\tag{6}$$

图 1 给出了增量查表 ML 分类器的实现流程图．对该分类器存贮表的容量要求为

$$B_s=B_tNM\left[\sum_{i=1}^{M}(2\Delta L_i+1)\right],\tag{7}$$

式中 B_t 为一个单元的字节数，这里是实型数，所以 $B_t=4$；ΔL_i 为数据在第 i 维上的增量变化幅度，在实现中，可根据容量的限制进行调整．

注意，在计算 B_s 的表达式(式(7))中连乘项已变成求和形式，使得算法能用于高维特征分类器．同时 ΔL_i 的值是程序根据可用的内存容量自行调整的，这使算法有较好的通用性．

在增量查表 ML 分类器的实现中，由于涉及大量的查表操作，所以在表的快速检索上要有一些处理技巧．这里我们采用一维表，将公式(5)中的 M 维表按下述方式转换

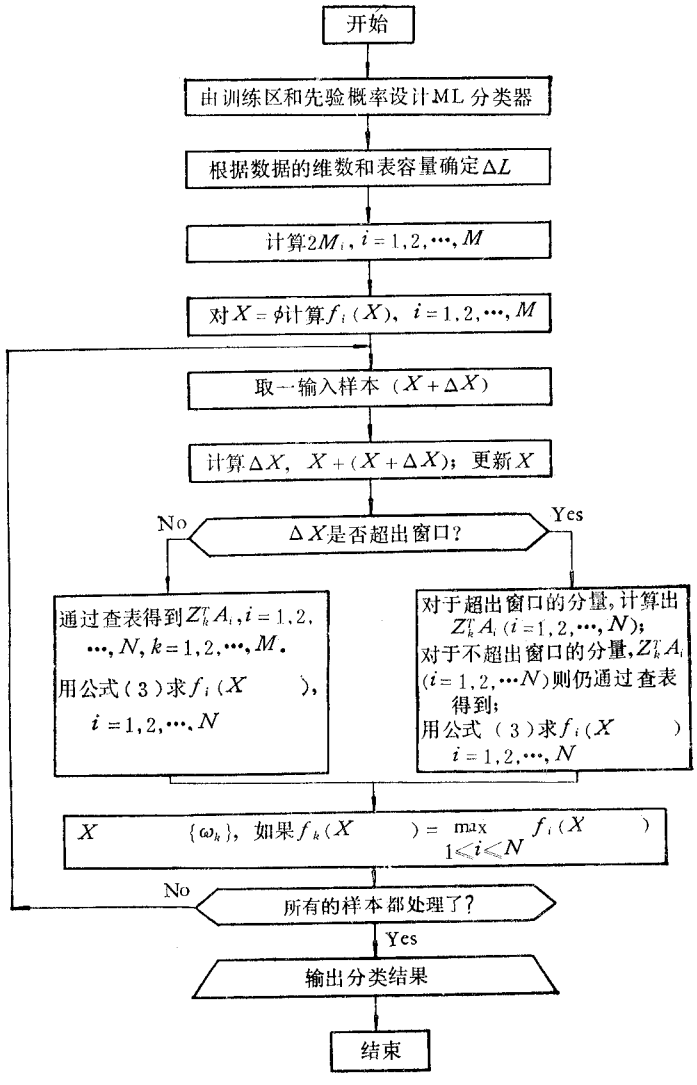

图 1　增量查表 ML 分类器流程图

成一维.

设有M维数组 TABA$(N_1, N_2, \cdots, N_M)$ 和一维数组 TABB(N_0)，则可以用公式

$$k_0 = (\cdots((k_M - 1) * N_{M-1} + k_{M-1} - 1) * N_{M-2} + k_{M-2} - 1) * N_{M-3} + k_{M-3} - 1) * N_{M-4} + \cdots + k_3 - 1) * N_2 + k_2 - 1) * N_1 + k_1 \quad (8)$$

建立起 TABA 与 TABB 的一一对应的关系. 这里 $N_0 = N_1 N_2 \cdots N_M$，$1 \leqslant k_i \leqslant N_i$，$i = 1, 2, \cdots, M$；TABA$(k_1, k_2, \cdots, k_M) \sim$ TABB(k_0).

实验表明，采用一维数组存贮技术后，较采用三下标数组存贮方法进一步节约运行时间 31.7%. 下面给出几组实验结果.

实验 3　相邻象素灰度变化的分布统计　为了了解式(7)中的 ΔL_i 取多大合适，我们对实验 1 中采用的数据，128×256 图幅统计了相邻象元灰度增量的分布规律.结果表明，

增量的平均变化范围比原始数据的平均变化范围小了许多，从而证明以增量代替原始变量的方法是可行的．表 1 给出了在不同窗口宽度下，常用特征组合可以用增量查表法计算的象元百分数．此时取 $\Delta L_i = \Delta L$，$i = 1, 2, \cdots, M$．

实验 4 采用一维查表法与三维查表法的速度比较 为了考察采用一维表存贮技术可提高速度多少，我们对实验 1 中介绍的数据，128 × 64 图幅，用不同的方法进行了四维

表 1 不同窗口宽度下可用增量查表法计算的象元百分数 (%)

特征 / ΔL	4	5	6	7	5,7	4,5,7	4,5,6,7
5	99.02	88.67	84.48	97.38	86.83	86.70	77.16
10	99.97	99.01	97.61	99.86	98.90	98.90	96.89
15	99.99	99.92	99.47	99.98	99.91	99.90	99.41
20	99.99	99.99	99.89	99.99	99.98	99.98	99.88
24	99.99	99.99	99.99	100.0	99.99	99.99	99.95

特征 12 类的分类实验，统计了各种情况下的运行时间，结果见表 2．采用一维表检索方案后，使分类所需时间减少了 31.7%．这充分说明了采用快速表检索技术的优越性．

表 2 采用一维表和三维表的运行时间比较

算 法	运行时间 (s)	较传统方法节约数(%)	较三维方法节约数(%)
传统 ML 方法	4672.6	—	—
三维查表方法	2626.8	43.8	—
一维查表方法	1795.3	61.6	31.7

实验 5 增量查表法在不同特征维数下应用的效果检验 为检验增量查表分类器的实际性能，我们用实验 1 中相同的数据，128×64 图幅，在 12 类的情况下统计了对不同特征维数下的分类时间，结果见表 3．表中作为比较还同时给出了相同情况下 ML 分类器的

表 3 增量查表 ML 分类器的性能检验

维 数	传统 ML 法时间 (s)	增量查表法时间 (s)	实测节约量(%)	理论节约量(%)
1	570.5	338.4	40.7	42.1
2	1519.8	734.3	51.7	57.1
3	2896.8	1221.9	57.8	64.9
4	4672.6	1795.3	61.6	69.6

运行时间，以及由式(6)求得的节约计算量的理论值．从表 3 可知，如式(6)所表明的，增量查表 ML 分类器节省时间的效果随使用的特征数的增加而变好．实测节约量略低于理论值，这是由于推导式(6)时忽略了磁盘数据读写时间、查表时间以及少数超出窗口的样本的处理时间．

在实现增量查表 ML 分类器时，在实际分类开始前先建立起增量表．实验表明，这一步骤所花的时间同整个分类时间相比可忽略不计．如对于四维特征 12 类的分类情况，建立增量表仅需几秒钟.需指出的另一点是：在用增量查表法进行分类时，由于进行计算的那一点的似然函数值与前一点的似然函数值有关，在理论上，因字长有限可能产生积累误差，但实际上，由于数据的增量是以正负等概率出现的，这种积累误差有相互抵消的趋势，所以造成分类错误的可能性很小．我们对四维 12 类 128 × 64 图幅的分类结果进行了检验，表明与采用传统 ML 分类器得到的结果完全一致．

5. 查表法用于其它统计分类器

上面介绍的增量查表算法的实质是利用 Landsat MSS 数据动态范围窄，可用查表法减少乘法运算次数来提高分类速度．这种技巧对其它的统计分类器也适用．下面就对用于其它几种常用统计分类器的实现方法进行讨论．

(1) 线性 Bayes 分类器　　线性 Bayes 分类器的判决函数为

$$f_i(X) = C_i + \sum_{j=1}^{M} a_{ij}x_j,\ i = 1, 2, \cdots, N; \tag{9}$$

其中 $X = (x_1, x_2, \cdots, x_M)^T$，$C_i, a_{i1}, \cdots, a_{iM}$ 是由第 i 类分布特性所决定的常数和系数．

对于式(9)中的乘法项 $a_{ij}x_j$ 如可通过查表得到，则可节省的计算量的理论值为 $\eta = 1 - \dfrac{M}{8M + M} = 8/9 \doteq 88.9\%$．对存贮表的容量要求为 $B_s = B_t N \sum_{i=1}^{M} L_i$，其中 $B_t = 4$．若取 $L = \max\limits_{1 \leqslant i \leqslant M} L_i$，则 $B_s = 4NML$．

(2) 最小欧氏距离分类器　　最小欧氏距离分类器的判决函数为

$$f_i(X) = (X - M_i)^T(X - M_i) = \sum_{j=1}^{M} (X_j - m_{ij})^2,\ i = 1, 2, \cdots, N. \tag{10}$$

实现时，$(x_j - m_{ij})^2$ 可通过查表得到，节省的计算量的理论值为 $\eta = 1 - \dfrac{M}{M + 8M + M} = 9/10 = 90\%$．这时，对贮存表的容量要求与线性 Bayes 分类器相同，为 $B_s = 4NML$．

(3) 加权最小欧氏距离分类器　　加权最小欧氏距离分类器的判决函数为

$$f_i(X) = \sum_{j=1}^{M} [a_{ij}(x_j - m_{ij})]^2,\ i = 1, 2, \cdots, N. \tag{11}$$

实现时，$[a_{ij}(x_j - m_{ij})]^2$ 可通过查表得到，节省的计算量的理论值为 $\eta = 1 - \dfrac{M}{M + 8M + 8M + M} = \dfrac{17}{18} \doteq 94.4\%$．对存贮表的容量要求与线性 Bayes 分类器相同，为 $B_s = 4NML$．

(4) 加权城市块距离分类器　　加权最小城市块距离分类器的判决函数为

$$f_i(X) = \sum_{j=1}^{M} a_{ij}|x_j - m_{ij}|,\ i = 1, 2, \cdots, N. \tag{12}$$

实现时，$a_{ij}|x_j - m_{ij}|$ 可通过查表得到，可节省的计算量的理论值为 $\eta = 1 -$

$\frac{M}{M+8M+M}=90\%$. 算法所要求的表容量与线性 Bayes 分类器相同，为$B_s=4NML$.

综上所述，查表法用于统计分类器可以大幅度提高分类速度. 该方法简便易行. 分类判决函数的计算式越复杂可节约的计算量越大. 对于最后讨论的四种统计分类器，用查表法后，分类的运算操作完全一样，需占用的存贮表的容量也相同. 所以分类时都可采用性能较好的加权最小欧氏距离分类器.

参 考 文 献

[1] K. Fukunaga 著，陶笃纯译，统计图形识别导论，科学出版社，1978 年 12 月.

FAST CLASSIFICATION ALGORITHM USING TABLE-LOOK-UP METHOD

Zheng Qinfen, Wu Jiangkang, Wang Wentao
(China University of Science and Technology)

In this paper a fast classification algorithm which uses table-look-up method is proposed. Both theoretical and experimental results have shown that (1) in two dimensional case it can reduce the computer time to the same order needed for the data I/O; (2) in the case of more than two dimensions it can save more than half of computer time. The classification accuracy is not affected by the use of table-look-up method. The algorithm is most suitable for computers without floating-point processor while there are many remote-sensed data to be processed.