Correspondence

On a Class of Linear Maps for Data Compression

SHOVONLAL KUNDU

Abstract—A method for data compression with linear maps has been developed which is found to produce further reduction in overhead storage requirement, compression/decompression time, and clustering overhead as compared to the affine map method in certain cases. Algorithms have been developed for cluster minimization, cluster identification, and compression matrix calculation that may be applied with advantage in both the methods.

Index Terms—Cluster analysis, computer algebra, data compression, dimensionality reduction, linear transformation, overhead storage, redundancy reduction.

I. Introduction

Dimensionality reduction by linear maps and the development of necessary algorithms have earlier been studied in the context of pattern recognition problems [3]. Young and Liu have proposed a multilinear method of data compression at the field level as an alternative to the FLMB scheme to reduce the overhead storage significantly at the cost of a somewhat smaller compression ratio [5]. Data items are divided into K clusters so that the translate of each cluster by its cluster center contains the same number of maximum linearly independent elements m and each cluster is compressed by an affine map so that overhead storage is required only for the K binary matrices and the K cluster centers in place of large C/D tables.

The object of this correspondence is to point out that each cluster may be compressed by a linear map in place of an affine map so that overhead storage for the K cluster centers will not be required and some addition operations can be avoided during the compression and decompression time. It has been shown that for a given group of data items, it is possible to obtain the same amount of compression with the linear map methods as that can be obtained with the affine map method, except in a very special case, where the amount of compression obtained by the latter will be one greater than that obtained by the former, although such a special case seems unlikely to occur in general. Algorithms have been developed here which makes the clusters disjoint, reducing thereby the number of clusters for a given m. This cluster minimization technique may reduce the overhead storage requirement as well as the length of compressed data items simultaneously. The algorithm to find out to which cluster a data item belongs have been simplified. It has also been indicated how some steps in the calculation of compression matrices from decompression matrices can be speeded up. These will actually result in reduction of the compression time. With slight modification, these algorithms for cluster minimization, cluster identification, and calculation of compression matrix from decompression matrix can also be used for the compression method with affine maps. It seems the linear map method will be better for overhead storage reduction purposes when applied with the cluster minimization technique as stated here, although for a practical case, it will be best to decide from the results of the experiments based on the methods of linear map as compared to that of affine map.

Another point of interest is that the linear map method as developed here is a perfectly general method valid for any finite dimensional vector space. The derivation is straightforward and avoids tricky manipulations. In addition to this it gives a clear picture of how the compressed data items are going to look like and indications of how we may vary it.

Manuscript received October 21, 1981.

The author is with the Electronics and Communication Science Unit, Indian Statistical Institute, Calcutta 700035, India.

II. Algebraic Results

The following theorem characterizes the maximum amount of compression possible with the linear map method for a given collection of data items.

Lemma 2.1: Let $\{x_1, \cdots, x_m\}$ and $\{y_1, \cdots, y_m\}$ be two subsets in vector spaces V and V', respectively, over the same field F. Let $x = a_1x_1 + \cdots + a_mx_m$, a_i in F, and $f(x) = y$. If $g: V' \to V$ is any linear map such that $g(y_i) = x_i$, $i = 1, 2, \cdots, m$, then $g(y) = x$.

Proof:

$$y = f(x) = a_1f(x_1) + \cdots + a_mf(x_m)$$

$$= a_1y_1 + \cdots + a_my_m$$

$$g(y) = a_1g(y_1) + \cdots + a_mg(y_m)$$

$$= a_1x_1 + \cdots + a_mx_m = x.$$

Theorem 2.1: Let $S = \{x_1, \cdots, x_m, x_{m+1}, \cdots, x_N\}$ be a subset of N vectors in a vector space V over a field F such that $\{x_1, \cdots, x_m\}$ forms a maximal linearly independent (LI) subset of S. Then the minimum dimension of a vector space V' over F such that there would exist linear maps $f: V \to V'$ and $g: V' \to V$ satisfying

i) all the $f(x_i)$'s are distinct for $i = 1, 2, \cdots, N$

ii) $g(f(x_i)) = x_i$ for $i = 1, 2, \cdots, N$

is m.

Proof: Let V' be any vector space over F of dimension $m' \geq m$ (e.g., $F^{m'}$).

Let $\{y_1, \cdots, y_m\}$ be an LI subset of V'. Extend $\{x_1, \cdots, x_m\}$ to a basis B of the vector space V. Define the linear map $f: V \to V'$ on B by $f(x_i) = y_i$ for $i = 1, \cdots, m$ and for points in $B - \{x_1, \cdots, x_m\}$ define $f(x)$ arbitrarily. Also extend $\{y_1, \cdots, y_m\}$ to a basis B' of the vector space V'. Define the linear map $g: V' \to V$ on B' by $g(y_i) = x_i$ for $i = 1, \cdots, m$ and for points in $B' - \{y_1, \cdots, y_m\}$ define g arbitrarily. Let $a_i, b_i \in F$ for $i = 1, \cdots, m$.

Now $f(a_1x_1 + \cdots + a_mx_m) = f(b_1x_1 + \cdots + b_mx_m)$ implies $a_1y_1 + \cdots + a_my_m = b_1y_1 + \cdots + b_my_m$ which is a contradiction as $\{y_1, \cdots, y_m\}$ is an LI subset of V'. So f satisfies part i). That part ii) is satisfied follows from the previous lemma. Next suppose that V' is a vector space over F of dimension $m' < m$ and linear maps f, g as stated above exists. Let $f(x_i) = y_i$ for $i = 1, \cdots, N$, so that $g(y_i) = x_i$ for $i = 1, \cdots, N$. Now among $y_1, \cdots, y_m$ at most m' are LI as V' has dimension m'. Assume that $\{y_1, \cdots, y_{m'}\}$ is an LI subset and hence a basis of V' so that $y_{m'+1} = b_1y_1 + \cdots + b_{m'}y_{m'}$ for $b_1, \cdots, b_m$ in F, with at least one $b_i \neq 0$, and $m' + 1 \leq m$. Now

0098-5589/82/0900-0530$00.75 © 1982 IEEE

$$g(y_{m'+1}) = b_1 g(y_1) + \cdots + b_{m'} g(y_{m'}) = b_1 x_1 + \cdots + b_{m'} x_{m'} = x_{m'+1}$$

which is a contradiction as $\{x_1, \cdots, x_m\}$ is an LI subset in V.

A. Calculation of Compression/Decompression Matrices

The matrix representations of the maps of f and g with respect to the natural bases will be called the compression and decompression matrices, respectively. We fix notations as follows. For a vector x, x_B denotes the row vector formed with the coordinates of x with respect to the ordered basis B of the underlying vector space. For an ordered set of vectors S, $(S)_B$ denotes the matrix formed by writing the coordinates of the elements of S in the given order as row vectors with respect to the ordered basis B. If B_1, B_2 are two ordered bases of a vector space, then we know that $x_{B_2} = x_{B_1}(B_1)_{B_2}$. If f is a linear map from a vector space V to a vector space W with respective ordered bases B_1 and B_2 then the matrix representation of f with respect to these bases is denoted by $(f)_{B_1,B_2}$ and is the matrix $(S)_{B_2}$ where $S = \{f(x_1), \cdots, f(x_n)\}$ and $B_1 = \{x_1, \cdots, x_n\}$ [2]. Let $V = F^n$ and $V' = F^m$, $m \leqslant n$ with N_1 and N_2 being the natural bases, respectively. The compression matrix $C = (f)_{N_1,N_2}$ and the decompression matrix $D = (g)_{N_2,N_1}$.

Extend the maximal linearly independent subset $\{x_1, \cdots, x_m\}$ to a basis $B_1 = \{x_1, \cdots, x_m, y_1, \cdots, y_{n-m}\}$ of V. $B_2 = \{z_1, \cdots, z_m\}$ be a basis of V' and $\{a_1, \cdots, a_{n-m}\}$ be an arbitrary set of $n - m$ elements from V'.

Now $C = (B_1)_{N_1}^{-1}(f)_{B_1,B_2}$ and $D = (B_2)_{N_2}^{-1}(g)_{B_2,B_1}$ where

$$(f)_{B_1,B_2} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ a_1 \\ \vdots \\ a_{n-m} \end{bmatrix} \quad \text{and} \quad (g)_{B_2,B_1} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

So if $(f)_{B_1,B_2}$ and $(g)_{B_2,B_1}$ remain stored, C, D can be calculated as required. We make the following choice to make the storage requirements less. Let $a_1 = \cdots = a_{n-m} = 0$ and $\{z_1, \cdots, z_m\}$ be the natural basis B_2 of V'. Then

$$D = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \quad \text{and} \quad C = (B_1)_{N_1}^{-1} \begin{bmatrix} I_m \\ 0_{(n-m)\times m} \end{bmatrix}$$

= the matrix formed by first m columns of $(B_1)_{N_1}^{-1}$.

It is possible to determine C from D, when $\{y_1, \cdots, y_m\} \subseteq N_1$ without explicitly determining the y_i's and inverting the whole of the $n \times n$ matrix $(B_1)_{N_1}$ as follows. Perform elementary row operations on the matrix $[DI_m]$ to reduce it to $[HB]$ where H is the row reduced echelon form of D [1]. Let the first nonzero entry in the ith row of H occur in position n_i for $i = 1, \cdots, m$. Then C is the $n \times m$ matrix whose n_ith row is the ith row of B for $i = 1, \cdots, m$ and whose other rows are zero.

The next theorem shows that in many cases the maximum amount of compression for a given set of data items obtained by the linear map method is the same as that obtained by the affine map method except in a special case where the maximum amount of compression obtained by the linear map method is one less than that obtained by the affine map method.

Theorem 2.2: Let $\{x_1, \cdots, x_m\}$ be a maximal linearly independent subset of the set of vectors $\{x_1, \cdots, x_m, y_1, \cdots, y_n\}$; then $m - 1 \leqslant \text{rank}\{x_2 - x_1, \cdots, x_m - x_1, y_1 - x_1, \cdots, y_n - x_1\} \leqslant m$. Also $\text{rank}\{x_2 - x_1, \cdots, x_m - x_1, y_1 - x_1, \cdots, y_n - x_1\} = m - 1$ iff the coordinate of each y_j with respect to the set $\{x_1, \cdots, x_m\}$ satisfies the equation $a_1 + \cdots + a_m = 1$ in the underlying field for $j = 1, \cdots, n$.

Proof: It is clear that

$$\text{rank}\{x_1, \cdots, x_m, y_1, \cdots, y_n\} = \text{rank}\{x_1, x_2 - x_1, \cdots, x_m - x_1, y_1 - x_1, \cdots, y_n - x_1\}$$

and that $\{x_1, x_2 - x_1, \cdots, x_m - x_1\}$ is a linear independent set as the subtraction of x_1 corresponds to a kind of elementary operation on a finite set of vectors [4]. Hence, $\{x_1, x_2 - x_1, \cdots, x_m - x_1\}$ is a maximal linearly independent subset of $\{x_1, x_2 - x_1, \cdots, x_m - x_1, y_1 - x_1, \cdots, y_n - x_1\}$ so that $\{x_2 - x_1, \cdots, x_m - x_1\}$ is always a linearly independent subset of $\{x_2 - x_1, \cdots, x_m - x_1, y_1 - x_1, \cdots, y_n - x_1\}$ which establishes the bounds on the rank as stated above.

Next we have rank $\{x_2 - x_1, \cdots, x_m - x_1, y_1 - x_1, \cdots, y_n - x_1\} = m - 1$ iff $\{x_2 - x_1, \cdots, x_m - x_1\}$ is a maximal linearly independent subset of $\{x_2 - x_1, \cdots, x_m - x_1, y_1 - x_1, \cdots, y_n - x_1\}$. So for each y_j, $j = 1, \cdots, n$ we can write $y_j - x_1 = b_2(x_2 - x_1) + \cdots + b_m(x_m - x_1)$ and let $(a_1, \cdots, a_m)$ be the coordinate of y_j with respect to $\{x_1, \cdots, x_m\}$ so that

$$y_j = a_1 x_1 + \cdots + a_m x_m = (1 - b_2 - b_3 - \cdots - b_m)x_1 + b_2 x_2 + \cdots + b_m x_m$$

where a_i, b_j are scalars. As the coordinate of y_j is unique it is clear that it satisfies the above equation in the underlying field.

III. Cluster Minimization and Identification

The following describes the basic algorithm which will be used for both minimizing the number of clusters and identifying the cluster to which a given data item belongs.

A capital letter denotes a memory location and the corresponding lowercase letter denotes its content. The algorithm basically tries to reduce a rectangular matrix to its row echelon form while keeping track of the row interchanges with the help of another auxiliary array.

Input: Given an m element set of vectors from F^n, its elements have been arranged as the row vectors of an $m \times n$ array A. B is a one-dimensional array of dimension m which indexes each row of A with initial values $B[i] = i$, $i = 1, \cdots, m$.

Output: If the first r rows of A are the only nonzero rows of A, then rows of A numbered $b_1, b_2, \cdots, b_r$ at the start of the algorithm constitutes a maximal linearly independent subset of the given m element set.

A. Algorithm

1) $Q = 0$.
2) Repeat the following steps i)–v) for $K = 1, 2, \cdots, m - 1$.
   i) Stop if $Q = n$.
   ii) $P = K$; $Q = Q + 1$.
   iii) Search for the first nonzero element in the $(m - k + 1) \times (n - q + 1)$ submatrix beginning with a_{kq} in a columnwise manner. If search is unsuccessful stop, else P, Q contains the row index and column index of the first nonzero element found in the above submatrix search.
   iv) If $P \neq K$ interchange the pth and kth row and also interchange the contents of $B[p]$ and $B[k]$. While interchanging it is sufficient to interchange the last $(n - q + 1)$ elements of the two rows.
   v) Add suitable multiples of the kth row to all the rows after it so that the last $(m - k)$ elements of the qth column becomes zero. It is sufficient to consider the rows after pth row.

While adding rows it is sufficient to add the last $(n - q + 1)$ elements of the two rows.

3) If the mth row is nonzero let its first nonzero element occur in the qth column. Divide the last $(n - q + 1)$ elements of the mth row by a_{mq}.

It may be noted that after the kth execution of the loop in step 2) of Algorithm 3.1 a k'th row $(k' > k)$ contains only zeros iff $B[k']$ is linearly dependent on the vectors $B[1], \cdots, B[k]$ with respect to the initial numbering of the rows of the array A with the help of array B at the start of the algorithm. With this observation the clustering algorithm may be developed so that each cluster contains as many data items as possible.

B. Clustering Algorithm

1) Arrange the data items in a data array D in some order. $J = 0$.

2) Repeat the following steps 3)-7) until D becomes empty.

3) $J = J + 1$.

4) Form the array A by collecting all the remaining items in D. Subject A to Algorithm 3.1 but stop after the mth execution of the loop at step 2) of Algorithm 3.1.

5) Rows numbered $B[1], \cdots, B[m]$ of the array D are the rows of the decompression matrix G_J for the Jth cluster. Delete these rows from D.

6) Repeat the following step until all the zero rows in A have been accounted for.

7) If the kth row in A is zero delete $B[k]$th row from D.

The above algorithm ensures that a data item belongs to exactly one cluster and thereby helps to minimize the number of clusters K for a given value of m—the maximum number of linearly independent vectors in each cluster. Its execution will be faster than a strictly sequential clustering algorithm as it is possible to examine a number of data items at the same time for clustering.

Given a data item x, the cluster to which it belongs may be found out as follows. Form a matrix with $m + 1$ rows whose first m rows are the same as those of the G_J and the last row is x. Subject this matrix to Algorithm 3.1. If the row echelon form at the end contains m nonzero rows, then x belongs to the Jth cluster.

IV. Discussion and Conclusions

If n bit data items are compressed to m bits by dividing the data set into K clusters the overhead storage requirement for the affine map method is $Kn(m + 1)$ bits and that for the linear map method is Knm bits. The total length of the compressed data item is $m + \lceil \log_2 K \rceil$ for both the methods where $\lceil \cdot \rceil$ denotes the smallest integer greater than or equal to its argument. So a reduction in K as obtained with the cluster minimization algorithm stated above will decrease both the overhead storage requirement and the total length of the compressed data item with both the methods. If the number of clusters is same with both the methods with same value of m the overhead storage requirement will be less with the linear map method compared to that of the affine map method. Also the overhead of the clustering algorithm with the linear map method will be less because one does not need to subtract here the value of the cluster center from all other data items to form the cluster. In addition, the compression and decompression times for a data item will be smaller with the linear map method as for compression we may do with less m bit addition (XOR) operations and during decompression we may do with less n bit addition (XOR) operations.

From Theorem 2.2 it follows that the number of clusters will be greater with the linear map method only if there are some clusters so that the coordinate of every data item in this cluster has odd number of 1's with respect to the maximal linearly independent subset of this cluster. But in general it may be expected that data items with coordinates having both odd number of 1's and even number of 1's will be present in a cluster—particularly when we are trying to minimize the number of clusters by packing as many data items as possible in a particular cluster. However, in a particular case it is always best to conclude from the results of the practical experiments with both the methods. So the linear map method forms a complimentary approach to the affine map method for data compression. The algorithms for cluster minimization, cluster identification and for calculation of compression matrix from the decompression matrix as stated here are quite general in approach and can be used with advantages in both the linear and affine map methods.

Acknowledgment

The author wishes to thank Prof. D. Dutta Mazumder, Head of the Electronics and Communication Science Unit, for his kind interest and encouragement during this work.

References

[1] K. Hoffman and R. Kunze, *Linear Algebra*. Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 11-12.

[2] —, *Linear Algebra*. Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 86-94.

[3] S. Kundu, "Optimum coding in pattern recognition and development of its computer realizable algorithms," M.E. thesis, Dep ETCE, Jadavpur Univ., Calcutta, 1977.

[4] S. Maclane and G. Birkhoff, *Algebra*. New York: Macmillan 1979, pp. 212-213.

[5] T. Y. Young and P. S. Liu, "Overhead storage consideration and a multilinear method for data file compression," *IEEE Trans. Software Eng.*, vol. SE-6, pp. 340-347, 1980.