

A Search Approach to General Stabbing Problems

Carl Ponder

Computing Research Group, L-306
Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, CA. 94550

March 29, 1991

Abstract

The general "stabbing" or "transversal" question in n -space asks whether or not a flat of dimension k can simultaneously intersect each of m figures. A search procedure finds whether or not an intersect exists in $O((d\sqrt{mn}/\epsilon + 2)^{mn}(p(mn) + Q))$ steps, where ϵ is the distance from the "best solution" to the figures in question, d is the diameter of the largest figure, Q is the time to test membership inside every figure, and p is a small polynomial. Though not very efficient, particularly for the understood cases of $k = 0, 1, n-1$, or n , no other general method is known in the literature. A variation of this procedure, implemented on a Cray I, was used to solve a related statistics problem to within reasonably tight bounds.

1 Introduction

Given a set F of figures f_1, \dots, f_m in R^n , a k -transversal or k -stabber of F is a flat of dimension k which intersects every element of F . Finding a k -transversal of F , or that none exists, is a significant problem in computational geometry. A related statistics problem is finding the closest k -flat to a set of points, under a cumulative distance measure. A related linear algebra problem is finding the closest matrix of rank $\leq k$ under some metric.

Previous results concentrate on the cases $k = 1$ or $k = (n-1)$ [1] [2] [4] [6] [9], using constructions familiar to the theory of computational geometry, such as convex-hulls, locuses, linear-programming, or Helly-type theorems [3], and do not appear to extend to intermediate values of k . This work is concerned with arbitrary k -transversals, and draws instead from the theory of linear algebra: the problem is reduced to *principle component analysis* [7], which uses the *singular-value decomposition* [5] to find low-rank matrices within a neighborhood.

A naive approach is to reduce each f_i into a finite set of gridpoints g_i , pick a gridpoint from each g_i , then see if the set of chosen points defines a k -flat. If repeating this process does not eventually produce a transversal, either none exists or no transversal intersects all m grids. The fundamental question is *what information does a non-solution provide?* Sections 2 and 3 show an important structure to the problem, adapting the naive approach to make use of non-solutions in an exhaustive search procedure described in section 4. An implementation is described in section 6.

2 Figures in R^n and $R^{n \times m}$

The space of $n \times m$ real matrices is Euclidean, partially ordered under

$$A \leq B \quad \text{iff} \quad A_{ij} \leq B_{ij} \quad \forall i, j$$

and distanced under the Frobenius metric

$$L_F(A, B) = \sqrt{\sum_{i,j} (A_{ij} - B_{ij})^2}$$

Given a set F of figures in R^n , let \overline{F} be the set of $n \times m$ matrices M such that M_j corresponds to a point in f_j . \overline{F} defines a figure in $R^{n \times m}$.

If a matrix M has $\text{rank} \leq k$, the columns of M span a flat of dimension $\leq k$ in R^n which also intersects the origin. Therefore, if \overline{F} contains a matrix M of $\text{rank} \leq k$, M_j is a point in f_j and these points all lie in a common k -flat. This common k -flat is a k -transversal of F through the origin.

Sections 3 & 4 explain how to exhaustively search \overline{F} for matrices of $\text{rank} \leq k$, in such a way that we eliminate neighborhoods devoid of $\text{rank} \leq k$ matrices. Section 5 lifts the restriction that the k -flat intersect the origin, to find general transversals.

3 Principle component analysis

A statistical method called *principle component analysis* allows us to find the closest $\text{rank} \leq k$ matrix under L_F to a given matrix A . For positive diagonal matrices we have a simple solution:

Lemma:

Given a positive diagonal matrix Λ , let $\hat{\Lambda}$ be a matrix of $\text{rank} \leq k$ minimizing $L_F(\Lambda, \hat{\Lambda})$. $\hat{\Lambda}_{ii} = \Lambda_{ii}$ for the k elements of largest magnitude in Λ , and $\hat{\Lambda}_{ij} = 0$ everywhere else. (There may be several such $\hat{\Lambda}$ if elements of Λ are repeated).

Proof:

Start with some $\text{rank} \leq k$ matrix $\hat{\Lambda}$ minimizing $L_F(\Lambda, \hat{\Lambda})$. This matrix $\hat{\Lambda}$ can be factored as $\hat{\Lambda} = L_{n \times k} R_{k \times m}$, where the columns of L are orthonormal. Under column operations L generates a space containing $\hat{\Lambda}$; the closest matrix in this space to Λ is the projection of Λ onto this space, so

$$\hat{\Lambda} = LL^T \Lambda$$

and the squared distance between them can be expressed as

$$\begin{aligned} \sum_{i,j} (\Lambda_{ij} - \hat{\Lambda}_{ij})^2 &= \sum_{i,j} (\Lambda_{ij} - LL^T \Lambda_{ij})^2 = \text{tr}((\Lambda - LL^T \Lambda)^T (\Lambda - LL^T \Lambda)) \\ &= \text{tr}(\Lambda^T \Lambda - 2\Lambda^T LL^T \Lambda + \Lambda^T LL^T LL^T \Lambda) = \text{tr}(\Lambda^T \Lambda) - \text{tr}(\Lambda^T LL^T \Lambda) \end{aligned}$$

To minimize this we need only maximize $\text{tr}(\Lambda^T LL^T \Lambda)$. We will argue by monotonicity that L will not want to spread the mass of Λ off of the diagonal. Now

$$\text{tr}(\Lambda^T LL^T \Lambda) = \sum_j \sum_i L_{ij}^2 \Lambda_{ii}^2 = \sum_i \Lambda_{ii}^2 \sum_j L_{ij}^2 = \sum_i \Lambda_{ii}^2 h_i$$

where $h_i = \sum_j L_{ij}^2$. The h_i are the diagonal elements of LL^T . Now

$$\sum_i h_i = \text{tr}(LL^T) = \text{tr}(L^T L) = \text{tr}(I_{k \times k}) = k$$

since the columns of L are orthonormal. $h_i \geq 0$ by definition. If any diagonal element in LL^T were greater than one, it would have to increase in $(LL^T)(LL^T)$ since it is squared and summed with other squares. But $L(L^T L)L^T = LL^T$, so $h_i \leq 1$.

Now $0 \leq h_i \leq 1$, $\sum h_i = k$, and $\Lambda_{ii} \geq 0$; the monotonicity property

$$\Lambda_{ii}^2 > \Lambda_{jj}^2 \implies (h_k + h_l)\Lambda_{ii}^2 > h_k \Lambda_{ii}^2 + h_l \Lambda_{jj}^2$$

implies that the sum

$$\sum_i \Lambda_{ii}^2 h_i = \text{tr}(\Lambda^T LL^T \Lambda)$$

must maximize with $h_i = 1$ for the k largest Λ_{ii}^2 , and $h_i = 0$ otherwise. The solution is for LL^T to contain 1's in all diagonal positions corresponding to the k largest Λ_{ii}^2 , and 0's everywhere else; the product $LL^T \Lambda = \hat{\Lambda}$ is the matrix Λ with the $(n - k)$ elements of Λ_i of smallest magnitude set to zero.

□

This generalizes to arbitrary real matrices as follows:

PCA Theorem:

Given a matrix A , the expression

$$\min_{\hat{A}} L_F(A, \hat{A}) \quad \hat{A} \text{ has rank } \leq k$$

has value $(\lambda_{k+1}^2 + \dots + \lambda_n^2)^{\frac{1}{2}}$ where λ are the singular-values of A in order $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$.

Proof:

Under the *singular-value decomposition* theorem [5]

$$\Lambda = U^T A V$$

where U and V are unitary matrices, and Λ is a diagonal matrix with elements $\Lambda_{ii} = \lambda_i \geq 0$ ordered nonincreasing in i . Also

$$A A^T = (U \Lambda V^T)(U \Lambda V^T)^T = U \Lambda^2 U^T$$

so U^T is the matrix of orthonormal eigenvectors of $A A^T$ and $\Lambda \Lambda^T$ is the diagonal matrix of the corresponding eigenvalues.

The Frobenius metric L_F is preserved under unitary transformations, so

$$L_F(A, \hat{A}) = L_F(U^T A V, U^T \hat{A} V) = L_F(\Lambda, \hat{\Lambda})$$

where $\hat{\Lambda}$ is a closest rank $\leq k$ matrix to Λ under L_F . But by the Lemma, a closest rank $\leq k$ matrix $\hat{\Lambda}$ is found by zeroing the $(n - k)$ smallest diagonal elements from Λ , so

$$L_F(\Lambda, \hat{\Lambda})^2 = \lambda_{k+1}^2 + \dots + \lambda_n^2$$

□

Corollary:

Some rank $\leq k$ matrix \hat{A} minimizing the expression

$$\min_{\hat{A}} L_F(A, \hat{A})$$

has the form $\hat{U} \hat{U}^T A$, where the columns of \hat{U} are the eigenvectors corresponding to the k largest eigenvalues of $A A^T$.

Proof:

Continuing the proof of the PCA Theorem, $\hat{A} = U \hat{\Lambda} V^T$ is a closest rank $\leq k$ matrix to A . Substituting,

$$\hat{A} = U \hat{\Lambda} V^T = \hat{U} \Lambda V^T$$

where \hat{U} is produced by zeroing the last $(n - k)$ columns of U . Now

$$\hat{U} \Lambda V^T = \hat{U} U^T A V V^T = \hat{U} U^T A$$

and $\hat{U} U^T = \hat{U} \hat{U}^T$.

□

The critical observation: a given matrix A is the center of a ball of radius $\min_{\hat{A}} L_F(A, \hat{A})$; a rank $\leq k$ matrix of the form $\hat{A} = \hat{U} \hat{U}^T A$ lies on the surface of this ball, and *no rank $\leq k$ matrix lies in its interior*. (In fact, since rank is nonincreasing as we scale a matrix, no rank $\leq k$ matrix lies inside the cone extending the ball thru the origin.) The corollary gives us a way of finding rank $\leq k$ matrices within a given neighborhood, and the theorem gives us a way of discarding neighborhoods as containing no rank $\leq k$ matrices. The next section devises a way to systematically search the volume of a figure \bar{F} for rank $\leq k$ matrices.

4 Searching for a transversal

To solve the transversal problem for F , then, we must either find a rank $\leq k$ matrix inside \overline{F} or show none exists by completely covering the volume of \overline{F} with "empty" balls. The running time can be bounded in terms of ϵ , defined as follows:

- For rank $\leq k$ matrix $M \in \overline{F}$, let ϵ be the minimal distance from M to the surface of \overline{F} , maximized across all such M .
- If no rank $\leq k$ matrix lies within \overline{F} , let ϵ be the minimal distance from the surface of \overline{F} to any rank $\leq k$ matrix $M \in R^{n \times m}$.

The transversal of F must intersect points *inside* the figures, rather than on the surface. Otherwise $\epsilon = 0$ and the procedure may not terminate. Our procedure is useless for figures of 0-volume in R^n .

A quadrature algorithm operates as follows:

1. Start with $\hat{\epsilon} = d$, the diameter of the largest figure.
2. An axis-parallel square grid contains \overline{F} ; $\hat{\epsilon}$ is the diameter of each square, ϵ/\sqrt{mn} is the edge-length of each square, and there are $(d\sqrt{mn}/\hat{\epsilon} + 1)^{mn}$ gridpoints.
3. For each gridpoint x , ignore it unless $x \in \overline{F}$, in which case proceed to step 4.
4. Find a closest rank $\leq k$ matrix \hat{x} to x . If $\hat{x} \in \overline{F}$, this is a solution and we are done.
5. If $L_F(x, \hat{x}) < \hat{\epsilon}$, halve $\hat{\epsilon}$ and return to step 2.

This completes by the time $\hat{\epsilon} \leq \epsilon$:

- If rank $\leq k$ matrices lie within \overline{F} , they must lie closer to some interior gridpoint than does any rank $\leq k$ matrix outside \overline{F} , and hence be identified in step 4.
- Otherwise, the closest rank $\leq k$ matrix to \overline{F} is farther away from any gridpoint of \overline{F} than is the surface of \overline{F} , and is therefore known to lie outside of \overline{F} .

It requires at most $\log_2(d/\epsilon)$ major iterations. Steps 3 & 4 take $\leq (d\sqrt{mn}/\hat{\epsilon} + 1)^{mn}$ singular-value decompositions ($p(mn) \in O(\max(m, n)^3)$ operations) and membership tests (call this time Q). The total number of steps is then $O((d\sqrt{mn}/\epsilon + 2)^{mn}(p(mn) + Q))$. Section 6 gives some refinements which make the procedure more practical.

5 Transversals ignoring the origin

Here we reduce the problem of general transversal to transversal-thru-origin, by adjusting the figures F and dimension n . Adding a new coordinate $x_{n+1} = 1$ to each point x embeds R^n as an axis-parallel hyperplane in R^{n+1} . Call \overline{F} the set of $(n+1) \times m$ matrices corresponding to F in R^{n+1} ; each matrix $M' \in \overline{F}$ has the form

$$\begin{bmatrix} M \\ [1]_{1 \times m} \end{bmatrix}$$

where $M \in \overline{F}$.

Let X be a $(k+1)$ -flat thru the origin which intersects every figure in F in R^{n+1} , and X' be a k -flat in R^n which intersects every figure in F in R^n , not necessarily thru the origin. In R^{n+1} they are equivalent:

- X' is the intersection of X and the hyperplane $x_{n+1} = 1$.
- X is the closure of X' under addition and scaling of points.

Although the problems of general transversal and transversal-thru-origin now appear equivalent, $\epsilon = 0$ for the embedded figures in R^{n+1} if a solution exists. Further transformation is necessary: for each $Y' \in \overline{F}'$, construct $Y''_{n \times (m-1)}$ such that

$$Y''_{ij} = Y'_{i(j+1)} - Y'_{i1}$$

This transforms \overline{F}' into $\overline{F}'' \subset R^{n \times (m-1)}$. Subtracting the first column of Y' from the remaining columns renders Y'_1 linearly independent of the remainder. The task is now to determine whether \overline{F}'' contains a rank $\leq k$ matrix M'' , and

$$\left[Y'_1 \left| \frac{M''}{[0]_{1 \times (m-1)}} \right. \right]$$

is the rank $\leq (k+1)$ matrix defining X . The whole transformation is equivalent to multiplying $R^{n \times m}$ on the left by an $m \times (m-1)$ matrix, so the general shapes of the figures F are preserved.

6 An implementation

The actual complexity of the problem has much to do with the nature and representation of the figures F . If the figures are readily decomposable, the "membership tests" are irrelevant; rather, the interior points are simply selected and processed. This is complicated by the transformation in section 5, which may not be easy to perform under certain representations.

The motivation for solving this problem was to solve principle component analysis under the L_∞ metric [10] [11]. Given A we want the rank $\leq k$ matrix X minimizing $\max_{ij} |A_{ij} - X_{ij}|$, equivalent to finding the minimal L_∞ -distance d from any k -flat through the origin to a set of points in R^n . Geometrically it translates to finding a rank $\leq k$ matrix X such that

$$A - [d]_{n \times m} \leq X \leq A + [d]_{n \times m}$$

for minimal d . The columns of A represent the points, and X is the "best" rank $\leq k$ approximation of A under L_∞ . For fixed d , the above expression defines \overline{F} to be an axis-parallel square hyperrectangle centered at A .

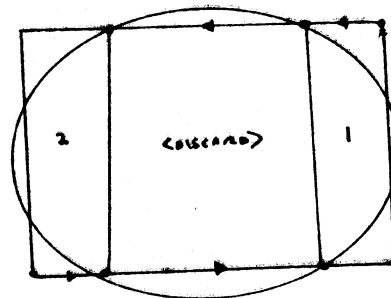
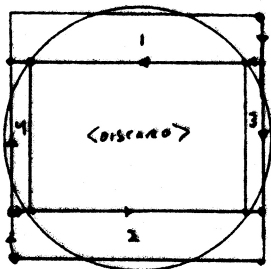
To minimize d , a variation of our search procedure is used: fill the region around A with "empty" spheres, to find the largest "empty" \overline{F} . Dividing \overline{F} into a 2^{mn} -element grid is computationally infeasible; the procedure in section 4 must be traded for something practical. A feasible search has to eliminate as much "empty" volume as possible from \overline{F} in each step, while compactly representing the search space. Three heuristics are used:

- Upperbounding d limits the size of the \overline{F} to be searched. An initial upper bound comes from finding the closest rank $\leq k$ matrix to A under L_F ; this can be improved with a Monte-Carlo procedure: randomly select a matrix W inside \overline{F} , and find its closest rank $\leq k$ matrix \hat{W} . \hat{W} factors into $Y_{n \times k} Z_{k \times m}$, and applying linear programming to Y and Z in alternation brings YZ closer to A under L_∞ .
- The balls of section 3 cover a limited volume, since their radii are determined by the rank $\leq k$ matrices within the neighborhood. Scaling the space converts these balls into ellipsoids, which can be contorted to cover more volume.

For a given box B , let c be its center and l be the length of its diagonal. Let r be the radius of the largest "empty" ball centered at c . For each of the $(n+m)$ dimensions, a scaling factor is chosen to maximize r/l . This constructs an ellipsoid which absorbs more of B 's volume. If $2r > l$ the ellipsoid contains B , so B can be discarded as containing no solutions.

- If B cannot be discarded, it is partitioned to eliminate much of the volume while generating few new

boxes. The box is hierarchically decomposed, as shown in the following sketches:



One subproblem is produced at each face where the outer and inner boxes do not meet. The inner box must be contained within the ellipsoid, but there are many ways it can be drawn. The inner box is made to meet the outer box along each dimension where the outer box is short in proportion to the length of the ellipsoid. This keeps the eliminated volume of the inner box large, while causing the boxes to meet along most dimensions.

The search procedure begins with an upper bound on d , defining the first box $B = [A - [d]_{n \times m}, A + [d]_{n \times m}]$. An ellipsoid is drawn inside B ; if B contains a solution the upper bound d is reduced, and the box is reprocessed. Otherwise B is partitioned into subproblems. The box closest to A is processed each time; its minimal L_∞ distance to A is a lowerbound on d . The upper and lower bounds on d converge as the search proceeds.

The subproblems, always axis-parallel hyperrectangles, are represented as pairs of upper and lower corners. Rather than using a 20×43 matrix to represent each corner point, each point is represented as an "edit" to another point, along one dimension. An upper- or lower-corner of a box is an edit to the upper- or lower-corner of the box it was cut from. The order of the edits is shown by the arrows in the above sketches. The space required is $2mn$ words to represent the first box, plus 4 words to represent each subsequent box that is not discarded.

Figures 1 & 2 give upper and lower bounds on the principle component analysis problem, for a 10×45 matrix and a 20×43 matrix. The matrices and the purpose of the analysis are described in [10], and details of the procedures are given in [11]. The bounds appear reasonably tight on this scale; each took several hours to compute on a Cray-1 using Eispack routines to perform the decompositions.

7 Areas for future research

Poljak & Rohn have shown that the problem of finding a singular matrix within a box is \mathcal{NP} -complete, equivalent to finding a hyperplane transversal for hyperrectangles [8]. Rohn's solution [12] to this problem has similar complexity to the linear-programming approach [4], and does not obviously extend to arbitrary k .

Better exponential algorithms should come from an understanding of the structure of $R^{n \times m}$. The subset of rank $\leq k$ matrices has interesting properties, of which only the most rudimentary have been used. Such an algorithm may be useful for transversal problems under restricted n . For principle component analysis problems n will tend to be large, but producing bounds which converge at a reasonable rate may be sufficient.

8 Acknowledgments

Special thanks to Barney Krebs and Terry Speed for introducing me to principle component analysis; Steve Ashby for the explaining the singular-value decomposition; Jim Demmel for familiarizing me with Jiri Rohn's work; and Cecilia Aragon, Anamaria Amenta, Jim Ruppert, Marco Pellegrini, Rephael Wenger, and Raimund Seidel for their discussions on the transversal problem.

This work was supported (in part) by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48. **DISCLAIMER:** This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

References

- [1] D. Avis, M. Doskas. *Algorithms for High Dimensional Stabbing Problems*. School of Computer Science, McGill University Tech. Report No. SOCS-87.2 (Jan. 1987).
- [2] S.E. Cappell, J.E. Goodman, J. Pach, R. Pollack, M. Sharir, R. Wenger. "The Combinatorial Complexity of Hyperplane Transversals." *Proc. 6th Ann. Symp. on Computational Geometry* (1990), 83-91.
- [3] L. Danzer, B. Grunbaum, V. Klee. "Helly's Theorem and its Relatives." *Convexity, Proc. Symp. Pure Math 7*, Amer. Math. Soc. (1963), 100-181.
- [4] H. Edelsbrunner. "Finding Transversals for Sets of Simple Geometric Figures." *Theoretical Computer Science 35* (1985), 55-69.
- [5] G.H. Golub, C.F. van Loan. *Matrix Computations*. Johns Hopkins University Press (1983), 16.
- [6] J.W. Jaromczyk, M. Kowaluk. "Skewed Projections with an Application to Line Stabbing in R^3 ." *Proc. 4th Ann. Symp. on Computational Geometry* (1988), 362-370.
- [7] K.V. Mardia, J.T. Kent, J.M. Bibby. *Multivariate Analysis*. Academic Press (1979), 213-250.
- [8] S. Poljak, J. Rohn. "Radius of Nonsingularity." To appear in *Mathematics of Control, Systems and Signals*.
- [9] R. Pollack, R. Wenger. "Necessary and Sufficient Conditions for Hyperplane Transversals." *Proc. 5th Ann. Symp. on Computational Geometry* (1989), 152-155.
- [10] C. Ponder. *An Analytical Look at Linear Performance Models*. Lawrence Livermore National Laboratory Tech. Report No. UCRL-JC-106105 (Sept. 1990).
- [11] C. Ponder. *Principle Component Analysis Under Alternative Error-Metrics*. In preparation.
- [12] J. Rohn. "Systems of linear interval equations." *Lin. Alg. Appl. 126* (1989), 39-78.

Figure 1: principle component analysis of 10x45 matrix

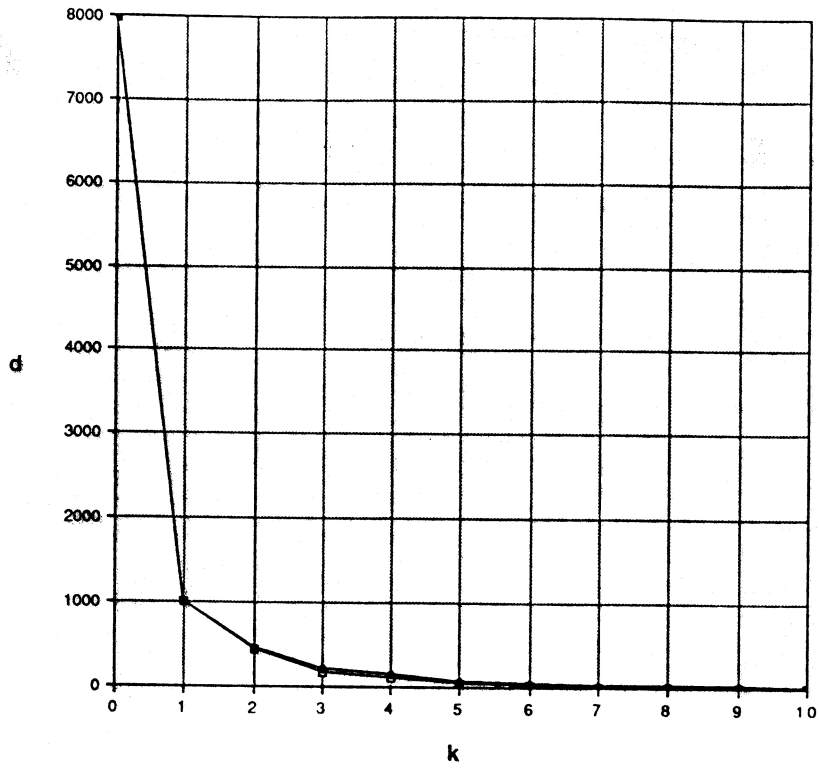


Figure 2: principle component analysis of 20x43 matrix

