A Method for Automatically Finding Interpretations of Reduced Dimension Representations

Marc Fasnacht *Rich Caruana*

March 2002

CMU-CALD-02-104

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Methods such as FastMap and Multidimensional Scaling often are used to project data to a lower dimensional subspace to make the data easier to understand. One drawback of these methods is that although it is easier to see patterns in the reduced dimension representation, interpreting the new dimensions is difficult. We present an automatic method for finding mappings (associations) between reduced dimension representations and auxiliary features that describe the data. Our approach finds groups of dimensions that taken together preserve local structure in the auxiliary feature space. Unlike previous approaches to this problem, this method works well in the non-linear mappings that often arise with reduced dimension projections. We use the method to assign meaning to dimensions resulting from applying MDS to protein helix pairs.

Keywords: data mining, clustering, multidimensional scaling, feature mapping discovery, k-nearest neighbor, protein structure

Contents

1 Introduction **2**

2 Background **3**

2.1 Multi-Dimensional Scaling (MDS) 3

2.2 FastMap 4

2.3 Interpretation of dimensions 5

3 Approach **5**

3.1 Alternate approaches 6

3.2 Time complexity 7

4 Experiments **7**

4.1 Synthetic Data 7

4.2 Protein Helix Data 10

4.3 Results 11

5 Discussion **21**

6 Conclusion **22**

7 Acknowledgments **22**

1 Introduction

Assume we have the following problem: we are given a set of points and a matrix of pairwise distances between the points with respect to some measure. A first step in better understanding the dataset is to apply multidimensional scaling (MDS) to the distance matrix to find a low-dimension coordinate representation of the distance relations among the set of points. This low-dimension vector representation allows easier visualization of the data, in part because it is a vector representation, and in part because it is low dimension. Unfortunately, the coordinate representation found by MDS does not come with a labeling of the coordinates. It may be easy to find patterns in the MDS coordinates, but interpreting these patterns can be difficult.

Suppose we also have a separate set of descriptive features for each point. These auxiliary features may relate to the distance metric in some fashion, but not necessarily in a simple or known way. Given the auxiliary descriptive features for the data, we do not a priori know if and how these feature are related to any of the MDS coordinates. Finding relationships between the auxiliary features and (some of) the MDS coordinates can help us understand the coordinates.

As a concrete example, suppose the objects are countries, and the pairwise distances are subjective similarities between pairs of countries as might be obtained from a questionaire. The auxiliary features would be attributes of the countries such as their population size, GNP, political system, land mass, length of coastline, etc. The goal of applying MDS to this data would be to find a small set of dimensions that faithfully capture the measured subjective similarities between countries[11][10][14]. Once MDS has reduced the data to a low dimension representation, the problem is to find interpretations of the MDS coordinates. Finding relationships between auxiliary features and the MDS coordinates can help us interpret the coordinates. For example, we might find that the first two MDS coordinates are most strongly related to the auxiliary attributes "political system" and "GNP".

If there is a linear relationship between the MDS coordinates and some of the features, we can find these automatically using linear regression. Unfortunately, as our examples show later, MDS often finds coordinates that do not have strong linear relationships to the auxiliary features. In the absence of an automatic way of finding relationships, users of MDS often resort to tedious manual methods of trying to establish an interpretation of the coordinate space. Manual approaches are impractical if there are many auxiliary features (common), if there are many dimensions in the MDS coordinate representation (less common), or if combinations of the MDS coordinates need to be considered (the usual case).

In this paper we describe a method that aids finding interpretations for MDS coordinates by automatically detecting linear and non-linear relationships between combinations of MDS coordinates and a set of auxiliary features. Notable features of the method are:

1. it depends on non-parametric nearest neighbor models of the data and thus makes few assumptions about the relationships and does not require that a parametric model be specified

2. unlike regression, it works with both numerical and discrete auxiliary attributes

3. it depends only on the low dimension representation, not on the method used to find it. Thus it can be used for reduced dimension representations found with MDS, PCA, non-linear PCA, etc.

The method works by analyzing the variance of k-nearest neighbor prediction from the MDS coordinates to the auxiliary attributes. Large reductions in variance (compared to random prediction) suggest a strong association between the coordinates and the auxiliary feature. The reduction in variance is qualitatively similar to the r^2 coefficient in regression, but applies equally well for linear and non-linear relationships.

The first part of this paper gives an overview of multidimensional scaling and FastMap. We then present our method for detecting relationships between MDS coordinates and other features. The third part presents results with synthetic data that highlight the inadequacy of linear models. The next part describes the results we obtained with the method on two real world protein datasets. On one dataset the method automatically rediscovers in several minutes an association that was laboriously previously discovered manually by an expert in protein data analysis after several days of work[8]. It also discovers several new important relationships that were not discovered manually.

2 Background

2.1 Multi-Dimensional Scaling (MDS)

One of the main purposes of multidimensional scaling (MDS) is to provide a coordinate representation of the similarity or distance relations among a set of objects. Often it also results in a dimensionality reduction of the problem. This allows easier visualization of the data and also allows the application of methods that rely on a coordinate representation (e.g., k-means clustering). The description of MDS given below closely follows the one given by Cox [4].

Suppose we are given a set of N objects for which we do not have a coordinate representation but are instead given a matrix of pairwise distances δ_{ij}. MDS allows us to find a set of N points $\{\mathbf{x_i}\}$, $i = 1, 2, \dots, N$ in a p-dimensional Euclidian space such that the distances between the points in that space d_{ij} obey

$$d_{ij} \approx f(\delta_{ij}). \tag{1}$$

Here f is a monotonic function of the distances. There are several methods of solving this problem. Given a set of coordinates $\{\mathbf{x_i}\}$ for the points, we can define a stress function which measures how well the spatial configuration of the points satisfies equation 1. A commonly used function is

$$S = \frac{\sum_{i \neq j}(d_{ij} - f(\delta_{ij}))^2}{\sum_{i \neq j} d_{ij}^2} \tag{2}$$

Finding a coordinate representation $\{\mathbf{x_i}\}$ of the data with the desired distances then corresponds to minimizing this stress function. Minimization can be done in standard fashion by using gradient descent or annealing techniques.

Another approach is the following: Let us define the matrix A as $A_{ij} = -1/2\delta_{ij}^2$, with δ_{ij} as defined above. We define a second matrix $\mathbf{B}$ as $\mathbf{B} = \mathbf{X}\mathbf{X}^\mathbf{T}$ where $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_p}]^\mathbf{T}$. These are the coordinates of the points we want to determine. It can be shown (c.f.[4]) that

$$\boldsymbol{B} = \boldsymbol{HAH} \tag{3}$$

where H is given by $\mathbf{H} = \mathbf{I} - \mathbf{N^{-1}11^T}$ with the length-N vector $\mathbf{1} = (\mathbf{1}, \mathbf{1}, \dots, \mathbf{1})^\mathbf{T}$. $\mathbf{B}$ is symmetric and positive semi-definite of rank p. So it can be decomposed into

$$\boldsymbol{B} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V^T} \tag{4}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues λ_i and $\mathbf{V} = [\mathbf{v_1}, \dots, \mathbf{v_N}]$ is the matrix of eigenvectors of $\mathbf{B}$. The problem of finding the coordinates x_i therefore reduces to solving the decomposition problem in equation 4, since $\mathbf{X} = \mathbf{V}\Lambda^{\frac{1}{2}}$. This is a standard problem in linear algebra. Generally, some of the eigenvalues are small so that they can be neglected. This simplifies the numerical solution of the problem. The advantage of this approach over direct minimizion of the stress function 2 is that the eigenvectors found are equivalent to those found by a principal component analysis in the projection space. This often simplifies interpretation of the data.

2.2 FastMap

FastMap [6] is an approximate but faster alternative to MDS. Given a set of N points and an $N \times N$ distance matrix, it finds approximate positions for the points in a p-dimensional Euclidian space such the relative distances are conserved. The basic idea of the algorithm is the following: We can pretend that the points are embedded in an unknown n-dimensional space and try to project them onto p-mutually orthogonal directions. In order to do this, we select two 'anchor' points that are far away from each other. The line between these two points in n-d space is our first direction. If we assume the space is Euclidian and that the triangle inequality holds for the points, we can use the cosine law from basic trigonometry to project the other points onto this line. We thus obtain the corresponding coordinates in the first direction. Given these coordinates, we can calculate the distances between all points in the hyperplane orthogonal to the first direction. The procedure is then repeated with these new distances until we have p orthogonal directions. This method is not quite as accurate as MDS, but the big advantage is that the complexity of the algorithm is $O(Np)$ whereas MDS is $O(N^2)$.

2.3 Interpretation of dimensions

Once we have a coordinate representation of the data, we can visualize it more easily. However, we do not usually know what the different MDS coordinates mean. Sometimes it is possible to find an interpretation by visual inspection [3], but this is not very practical even for a modest number of dimensions. For example, even if the MDS coordinates have only 10 dimensions, there are still 90 2-D projections and 720 3-D projections one would need to examine manually and interpret. Usually this is not feasible.

Methods that are often used in place of manual interpretation are multiple regression and cluster analysis [3][5][11][13]. Multiple regression can detect linear relationships between MDS coordinates and auxiliary features, but it does not work very well if the relationship is nonlinear.

In cluster analysis, the data is clustered in the MDS space, and then the clusters are examined to see if they can be interpreted in terms of the auxiliary features. The good news is that it is possible to automate the cluster analysis to see what dimensions yield clusters that associate with some of the auxiliary attributes. The bad news is that often points do not cluster well even when there is a strong relationship between the coordinates and the auxiliary attributes. For example, the existence of a linear relationship between coordinate c_i and auxiliary feature f_j does not imply that the data will cluster on the projection c_i in a way that maps well to a_j – correlation between attributes does not imply clumpiness. See for example the synthetic data in Figure1 which exhibits strong relationships that would not cluster well.

3 Approach

In this section we present an alternate way of detecting relationships between the MDS coordinates and auxiliary features. Let $(c_1, \dots, c_p)$ be the p coordinates from an MDS calculation and let $F = \{f_1, \dots, f_A\}$ be a set of A auxiliary attributes. Assume that we are interested in the relationship between auxiliary feature f_i and the a subset of MDS coordinates $\{c_i, \ldots, c_j\}$.

The intuition behind our approach is the following: Suppose two points are near each other in a subspace defined by coordinates $\{c_i, \ldots, c_j\}$. If this subspace is well characterized by feature f_i, the points should have similar values of f_i.

This property can be measured by looking for the nearest neighbor in the MDS subspace and calculating an average distance with respect to feature f_i over all the data. We then compare this value to the average distance between points in f_i.

If we are interested in a feature f_i, and want to test if it is related to a subset $\{c_i, \ldots, c_j\}$ of the MDS coordinates, we measure

$$v_1 = \frac{\sum_{s=1}^{N} (f_{i,s} - f_{i,nn})^2}{N} \tag{5}$$

The subscript nn refers to the nearest neighbor of point s in terms of the euclidian distance in the space $\{c_i, \ldots, c_j\}$. $f_{i,nn}$ is the value of that point for feature f_i. We also measure

$$v_2 = \frac{\sum_{s,t=1, s \neq t}^{N} (f_{i,s} - f_{i,t})^2}{N(N-1)} \tag{6}$$

which gives a measure of the average distance between pairs.

If the space $\{c_i, \ldots, c_j\}$ is unrelated to f_i, then v_1 and v_2 will be similar. Picking the closest point in that space would be equivalent to picking a random point. If there is a strong relationship, picking a close point in $\{c_i, \ldots, c_j\}$ should correspond to picking a point that is close in terms of f_i too, so v_1 should be much smaller than v_2. We can measure this in terms of

$$r' = 1 - \sqrt{\frac{v_1}{v_2}} \tag{7}$$

This quantity will be close to 1 for strong associations, and around zero for weak association.

The main property of this measure is that it measures how well local structure is preserved between the MDS-coordinate space and the auxiliary feature space. If the data is sufficiently dense so that local neighborhoods of points are connected, the measure will also take into account the preservation of the overall structure of the data.

3.1 Alternate approaches

We can user other topological mappings that preserve neighborhoods for our purposes. The simplest extension would be to use k-nearest neighbors, instead of just nearest neighbors. We can also apply a kernel method, where look at a weighted average of neighboring points rather than nearest neighbors.

3.2 Time complexity

The method considers models from all subsets of coordinates to each of the auxiliary attributes. Although this is a large number of models, in practice it usually is computationally feasible to examine them and we have not found it necessary to develop a more efficient approximate algorithm. There are two reasons why examining all models is feasible. First, usually a reduced dimension representation has relatively few dimensions or it would not be useful for visualization. Second, we are not interested in discovering models that require large numbers of dimensions because these will not be intelligible. Thus the original goal of finding intelligible reduced dimension representations saves us from having to consider combinatorially many models. If there are p dimensions in the representation, A attributes in the auxiliary set, and we are interested in relationships that use no more than l dimensions at a time, then we only have to explore

$$A \sum_{i=1}^{l} \binom{p}{i} \tag{8}$$

models.

For example if we have $A = 10$ attributes and $p = 8$ MDS dimensions and we want models with $l = 3$ or fewer attributes we would have to check only 920 cases. A more significant computational cost than evaluating z models is the cost of performing KNN once with N data points. A naive implementation of KNN is $O(N^2)$. If N is large, this cost quickly becomes the computational bottleneck. When N is large, there are two solutions to this problem. One is to use a better implementation of KNN such as kd-trees [1][2] to reduce the computation to $O(N \log N)$. For the higher dimensional cases more general data structures such as the ones described in [15] [9][12] can achieve similar performance. Even $(N \log n)$ can be expensive when N is large and may require more memory than is available. Another approach is to sample the data. Since we are looking for strong relationships from the dimensions carrying the majority of the variance in the original dataset, sampling is a simple and effective procedure. For this task we are not so concerned about small clusters of atypical points that might be lost by sampling.

4 Experiments

4.1 Synthetic Data

To evaluate how much benefit our approach might yield over linear methods, we tested tested it on a synthetic dataset. We simulated MDS data by generating points from a uniform as well as a normal distribution in one and two dimensions. To model a set of auxiliary features, we used

feature f	σ	r'	r^2
noise1	0.00	0.0031	0.0000
noise2	0.00	0.0689	0.0003
lin	0.00	0.9862	1.0000
lin	0.05	0.9708	0.9992
lin	0.25	0.8723	0.9805
quad	0.00	0.9714	0.0010
quad	0.05	0.9394	0.0010
quad	0.25	0.7414	0.0006
sin	0.00	0.9986	0.6015
sin	0.05	0.9277	0.5990
sin	0.25	0.6593	0.5365
exp	0.00	0.9934	0.8009
exp	0.05	0.9747	0.8003
exp	0.25	0.8790	0.7871
r	0.00	0.9736	0.0001
r	0.05	0.8703	0.0000
r	0.25	0.4568	0.0012
α	0.00	0.9035	0.5679
α	0.05	0.8616	0.5706
α	0.25	0.8536	0.5473

Table 1: Results for 3000 data points. For the first two lines, the feature is completely unrelated to the simulated MDS coordinate (uniform and normal noise). The remaining rows give the performance for functions of the type $f = ax + b + \epsilon$, $f = bx^2 + b + \epsilon$, $f = \sin(x)$ and $f = \exp(x)$, with $\epsilon \approx N(0, \sigma)$. (r, α) correspond to polar coordinates in the 2D case.

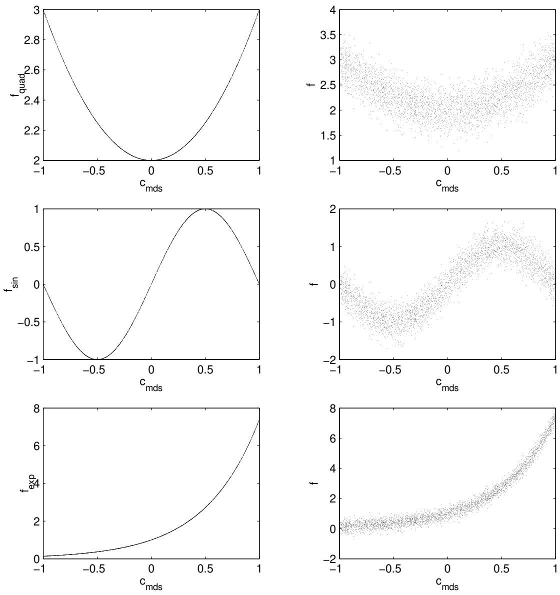


Figure 1: Plots of 1-dimensional artificial data. The horizontal axis corresponds to the 'MDS coordinate', the vertical axis to the 'feature' values. For the plots on the left hand side, there is no noise ($\sigma = 0.00$), for the plots on the right hand side, $\sigma = 0.25$

simple functions of the data, such as linear, quadratic, sinusoidal and exponential functions. Noise also was added to some features, and some features contain only noise. Figure 1 shows a subset of the one dimensional data used. Table 1 shows the results of applying linear regression as well as the new method to a set of 3000 sample points. Both methods perform as expected. When there is no relationship between MDS coordinate and feature (i.e. just noise), the r' and r^2 values both are close to zero. Not surprisingly, regression gives very high r^2 values if the feature is a linear function of the coordinates. For the purely quadratic case, linear regression does not detect any correlation, since the function is symmetric around the origin. For the sinusoidal function, regression detects some correlation, but the value drops from the value shown in the table to much lower values even in the case without noise if the data is spread over more than one period or the phase is shifted. The exponential is a monotonic function, so regression detects some correlation. In the two dimensional case the results are similar.

Table 1 also shows results for feature that are polar coordinates (r, α) of each point. We included polar coordinate features in our synthetic data because we observe similar polar relationships in the real protein data in the next section. The results clearly indicates that r^2 is a bad measure for automatically detecting polar-like relationships. The r' on the other hand shows high local correlations independent of the functional form, as long as there is not too much noise. If the noise has a similar amplitude than the total range of the feature value,

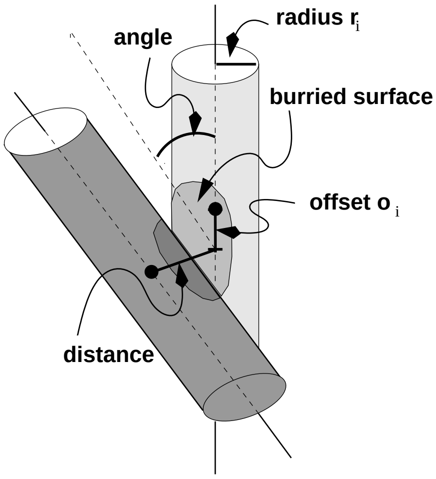


Figure 2: Schematic illustration of a helix pair. Descriptive features include crossing angle α , axial distance, d_{axis} and offset (only shown for one helix) of two helices. The minimal distance d_m (not shown) refers to the smallest distance between two atoms from each helix) The buried surface is the surface where the two helices touch each other

the r' signal starts to decrease, as expected from its definition. This shows that a high r'-value is a good indication that there is a relationship between the simulated MDS coordinates and the corresponding feature values.

The observation that KNN outperforms linear regression for prediction on non-linear datasets is, of course, not surprising. We include results on these synthetic datasets because we observe similarly “shaped” relationships when working with real data, but the patterns sometimes are more difficult to recognize when working with the real data.

4.2 Protein Helix Data

We apply our method to two datasets ultimately derived from the Protein Data Bank [7], a large database containing structural data of proteins and other biological macromolecules. Proteins are polypeptides - linear chains of amino acids, which fold into complex three-dimensional structures. While the overall structure of a protein is relatively complex, locally, the chain forms simple motifs, such as helices, sheets and loops. The relative spacial arrangement of such structural motifs is very interesting to structural biologists, since this information helps in understanding the function of the protein.

The datasets we created consisted of pairs of helices of fixed size. We created these datasets for research we are doing in machine learning for structure prediction. The selection criteria for the helix pairs was that they have to be in contact. We extracted two sets of helix pairs. The first consisted of 7681 helix pairs representatively selected from the set of all known protein families. The second set consisted of 3459 helix pairs taken exclusively from proteins belonging to the globin family[1]. Each helix pair is made up of 72 atoms. The PDB contains the spatial coordinates of these atoms. To measure the similarity between two sets of helix pairs, we calculate their rms distance. Rms distance is the measure most widely used in molecular biology to compare the structure of biological molecules. It is determined as follows: the two pairs of helixes are rotated and translated on top of each other such that the root-mean-square (rms) distance between corresponding atoms on the helices is minimized[2]. This minimal rms distance is used as the distance function.

We calculated the rms distance between all pairs using the *ProFit* program by Dr. Andrew C. R. Martin. This rms-distance was then used to do a MDS calculation. The number of dimensions in the MDS was 10. Besides the rms distance, we have a number of other features that describe a helix pair i, j. Figure 2 gives a schematic view of some of these features. They include crossing angle α, radius r_i, r_j, pitch p_i, p_j and offset o_i, o_j of both helices in a pair, the distance between the two helices (d_m as measured by distance between the two closest atoms, the distance between the axis of the helices, d_{axis}, the buried surface area.

4.3 Results

For both datasets, we applied the nearest neighbor method from section 3 and compared it to a multiple regression analysis.

Tables 2 and 3 show the results of applying linear regression as well as the KNN method to the data set with 7681 helix pairs. When we map one MDS coordinate into one auxiliary feature (Table 2 and the x's in Figure 3), the two methods produce roughly similar results. The strongest correlations are between the minimal distance between two atoms from each helix d_m and the second MDS coordinate c_2. There is also modest association between the angle α and the first MDS coordinate, as well as between the axial distance and MDS coordinate c_2.

The results for pairs of coordinates are shown in table 3. We can see that the similarity between the two methods is no longer very strong. Both methods agree reasonably well when there is a high degree of association. One important exception is the case of the crossing angle α. The KNN method shows that there is a strong connection between the MDS coordinate pair (c_1, c_3) and the crossing angle. The multiple regression r^2 value for this coordinate pair is only slightly larger than the value for α and c_1 alone. Figure 4 explains why this is the case: the auxiliary feature that is the angle α shows up in a polar coordinate-like relationship in the (c_1, c_3) cordinate plane. The data points lie on a clearly visible ring. Points that are close on

[1]The globin family of proteins is the family with the most entries in the PDB. The globin family contains proteins such as hemoglobin and myoglobin, which are responsible for oxygen transport and storage

[2]This corresponds to finding the orientation of maximal overlap. For identical structures it is zero

feature	c_i	r'	r^2
d_m	2	0.7264	0.8127
α	1	0.6215	0.5684
d_{ax}	2	0.5522	0.6937
d_m	1	0.3119	0.0852
d_{ax}	1	0.2926	0.0693
α	3	0.2573	0.1187
α	2	0.1752	0.0196
d_{ax}	3	0.1531	0.0004
o_i	1	0.1529	0.0006
o_j	1	0.1512	0.0000
o_i	7	0.1227	0.0087

Table 2: Association measures r' and r^2 for single MDS coordinates for the 7681 dataset. The first column is the 11 auxiliary descriptive features that are used to help provide interpretations for the MDS coordinates.

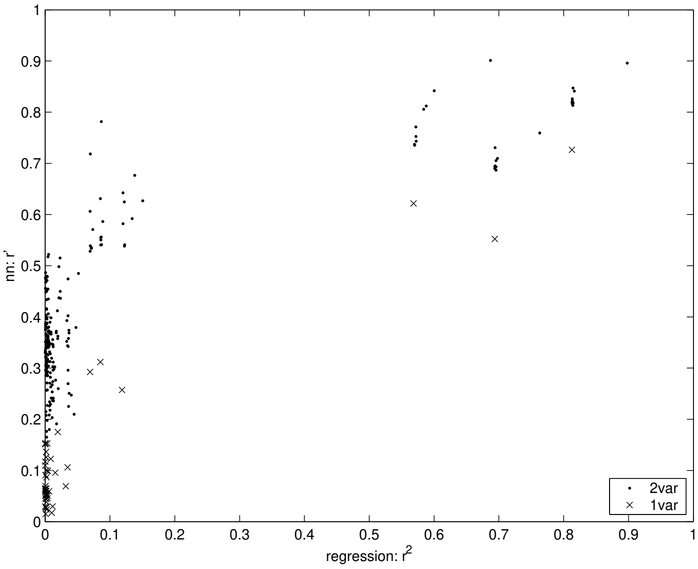
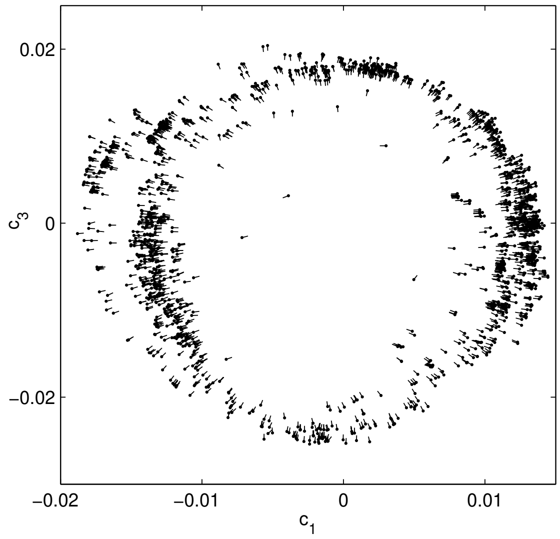


Figure 3: Association measures of the KNN method vs regression r^2 for the original dataset. The crosses show mappings from one coordinate, the dots show mappings from paris of coordinates. In the one coordinate case there is reasonable correlation. However, when there are pairs of coordinates, there are many cases for which the regression has a very low r^2 value whereas the r' for the KNN is quite high. Figure 4 shows one such case.



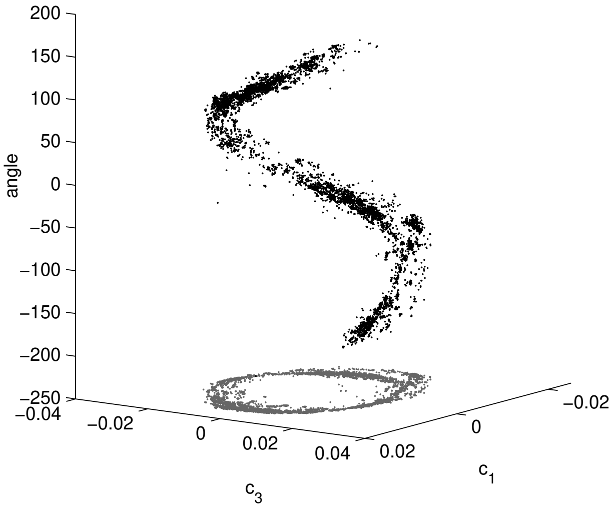


Figure 4: Plots of (c_1, c_3, α). The top figure shows a projection into the (c_1, c_3) plane. The feature "crossing angle" α is indicated by the tilt of the short line segments. The tilt of the lines changes smoothly around the ring of data, which indicates that α is related in almost polar coordinates to the MDS c1, c3 coordinates. The bottom graph shows a 3-dimensional plot of (c_1, c_3, α). A shaodw of the data also is projected into the (c_1, c_3) plane. The angle shows up in almost polar coordinates in the MDS c1, c3 coordinate system, so that the 3-d plot looks like a spiral.

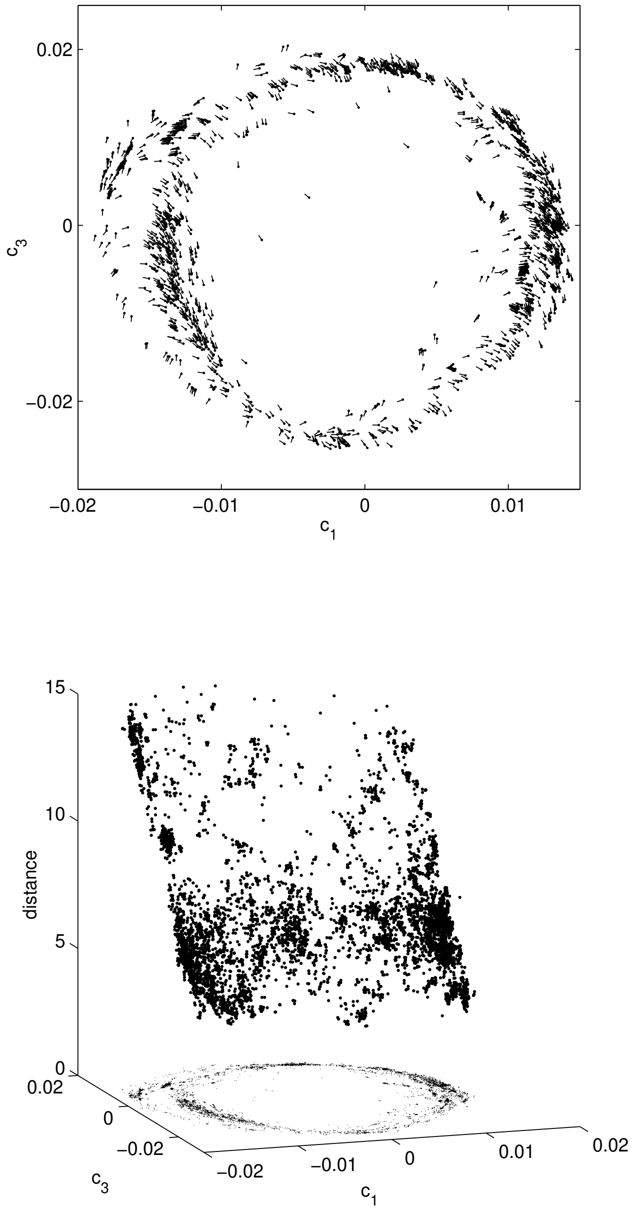


Figure 5: Plots of $(c_1, c_3, distance)$. This is a case for which $r' = 0.7816$ and the regression $r^2 = 0.0868$. The top figure shows a projection into the (c_1, c_3) plane. The distance is indicated by the tilt of the short line (pointing up: helices are close, pointing down, helices are far apart). The second plot shows a three dimensional representation of the data as well as a shadow projection onto the (c_1, c_3) plane on the bottom. The data seems to lie on a cylindrical surface, with the distance c_2 being almost parallel to the axis of the cylinder.

feature	c_i	c_j	r'	r^2
α	1	3	0.9011	0.6871
d_m	1	2	0.8959	0.8979
d_m	2	3	0.8471	0.8143
α	1	6	0.8418	0.6003
α	1	2	0.8120	0.5880
d_m	1	3	0.7816	0.0868
d_{ax}	1	2	0.7594	0.7631
d_{ax}	2	3	0.7307	0.6941
d_{ax}	1	3	0.7183	0.0697
α	2	3	0.6764	0.1383
d_m	1	6	0.6311	0.0853
α	3	6	0.6269	0.1506
α	3	4	0.6247	0.1221
d_m	1	7	0.5504	0.0859
α	3	7	0.5410	0.1228
d_{ax}	1	5	0.5389	0.0697
α	3	9	0.5384	0.1223
d_{ax}	1	9	0.5353	0.0713
d_m	3	4	0.5221	0.0054
α	2	4	0.5152	0.0230

Table 3: Association measures r' and r^2 for pairs of MDS coordinates for the 7681 dataset.

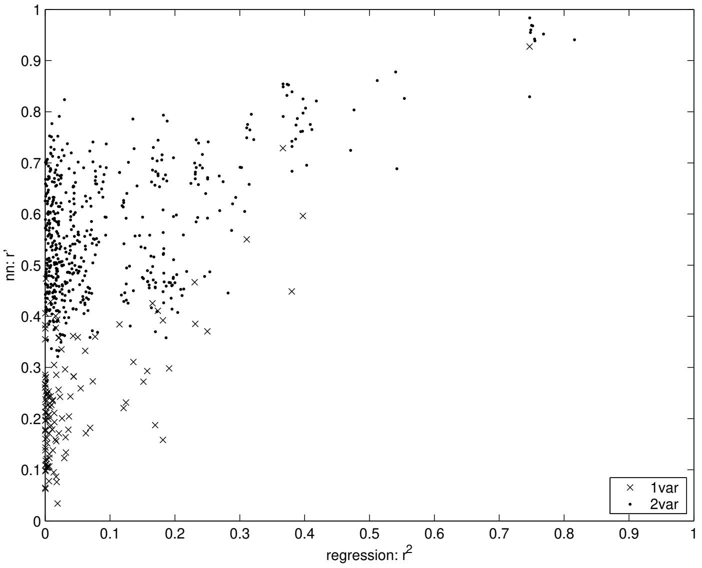


Figure 6: Association measures of the KNN method vs regression r^2 for the globin only dataset. The crosses show the one coordinate case, the dots the pairs of coordinates case. The two methods seem to be only weakly correlated indicating that different properties of the dataset are measured. In particular there are some points for which the regression has a very low r^2 value whereas the r' for the KNN is quite high. Figure 8 shows such a case

the ring have very similar angle α. On the other hand, the angle is not a linar combination of c_1 and c_3, so that the r^2 value for the regression is very low.

Figure 5 shows a case for which the KNN method has a high r'-value of $r' = 0.7816$ whereas the regression calculation gives a very low r^2 of $r^2 = 0.0868$. This is due to the fact that the data in the (c_1, c_2, c_3) coordinate systems appears to lie on a cylinder. The axis is almost parallel to c_2, which is strongly correlated to d_m (see table 2). However, the axis is not quite along the c_2 direction, but slightly tilted along c_1. The projection of the cylinder into the (c_1, c_3) results in a set of shifted rings, so that there is a local correlation of the distance along c_1. This can be seen on figure 5 and is picked up by the KNN method.

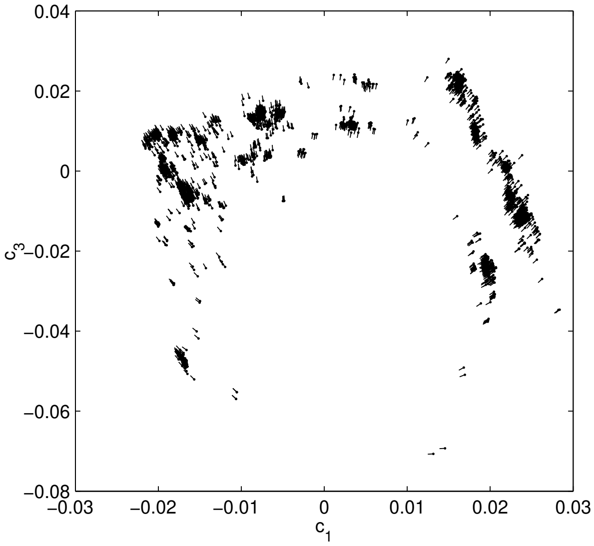
The second set of tables shows the results of applying linear regression as well as the KNN method to a set of 3459 helix pairs from proteins from the globin family. This dataset is qualitatively different from the 7681 dataset, since it contains a lot of very similar proteins. We therefore expect the data contain a lot of large clusters. Tables 4 and 5 compare r' and r^2 for the one, respectively 2 coordinate case. Figure 6 show a scatter plot of r' vs r^2. We can see that if there is a strong correlation between the MDS coordinates and the auxiliary feature, both measures agree reasonably well. Figure 7 shows the case with the highest r' value. We can see that clearly there is strong association between the crossing angle α and the position in

feature	c_i	r'	r^2
α	1	0.9274	0.7467
d_m	1	0.7287	0.3666
A_b	1	0.5964	0.3974
d_{ax}	1	0.5503	0.3107
o_j	1	0.4737	0.0013
r_i	1	0.4667	0.2304
d_m	2	0.4487	0.3801
α	4	0.4289	0.0083
d_{ax}	3	0.4255	0.1654
d_m	3	0.4107	0.1737
d_m	6	0.4061	0.0003
o_i	1	0.4007	0.0162
A_b	3	0.3843	0.1146
o_j	7	0.3706	0.2500

Table 4: Subset association measures for 1 variable for globin dataset. The strongest correlations are for the angle *alpha* and distance related measures

feature	c_i	c_j	r'	r^2
α	1	3	0.9835	0.7469
α	1	10	0.9688	0.7498
d_m	1	3	0.8777	0.5403
A_b	1	3	0.8611	0.5120
d_m	1	7	0.8545	0.3668
d_m	2	3	0.8260	0.5538
A_b	1	5	0.8251	0.3975
α	2	4	0.8237	0.0298
d_{ax}	1	3	0.8036	0.4761
d_{ax}	1	5	0.7951	0.3179
d_m	3	4	0.7934	0.1823
α	2	3	0.7909	0.0217
α	4	5	0.7768	0.0103
o_j	1	3	0.7522	0.0196
o_j	1	7	0.7410	0.2513
o_i	1	4	0.7279	0.1371
o_i	1	3	0.7101	0.1978

Table 5: Association measures r' and r^2 for 2 variables for globin dataset. As for the one variable case, the strongest correlations are for the angle α and distance related measures. The overall values are higher than for the 1D case, which is not surprising.



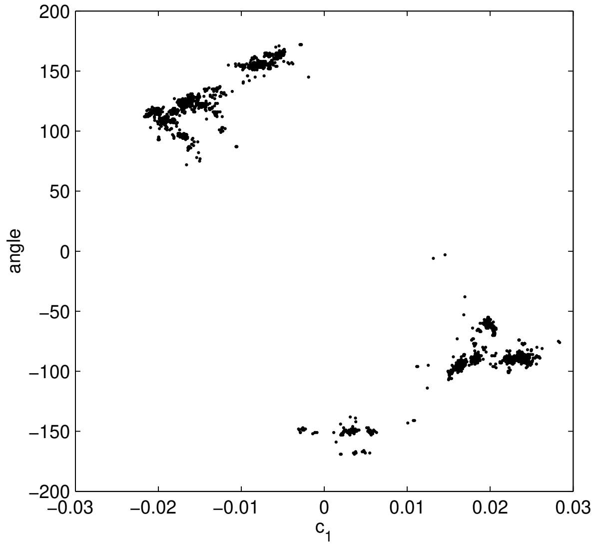


Figure 7: Plots of (c_1, c_3, α) for the globin dataset. This combination showed the highest values for both, r' and r^2. The top figure shows a projection into the (c_1, c_3) plane. The feature 'angle' is indicated by the angle of the short line. The orientation of the line changes smoothly, which indicates strong correlation. However, the relationship is not linear, due to the periodic nature of the angle. The transition from $180°$ to $-180°$ is around $c_1 = 0.0$. This results in a large jump, which explains the relatively low r^2 value. Regression would probably do much better if the origin of the angle to $180°$.

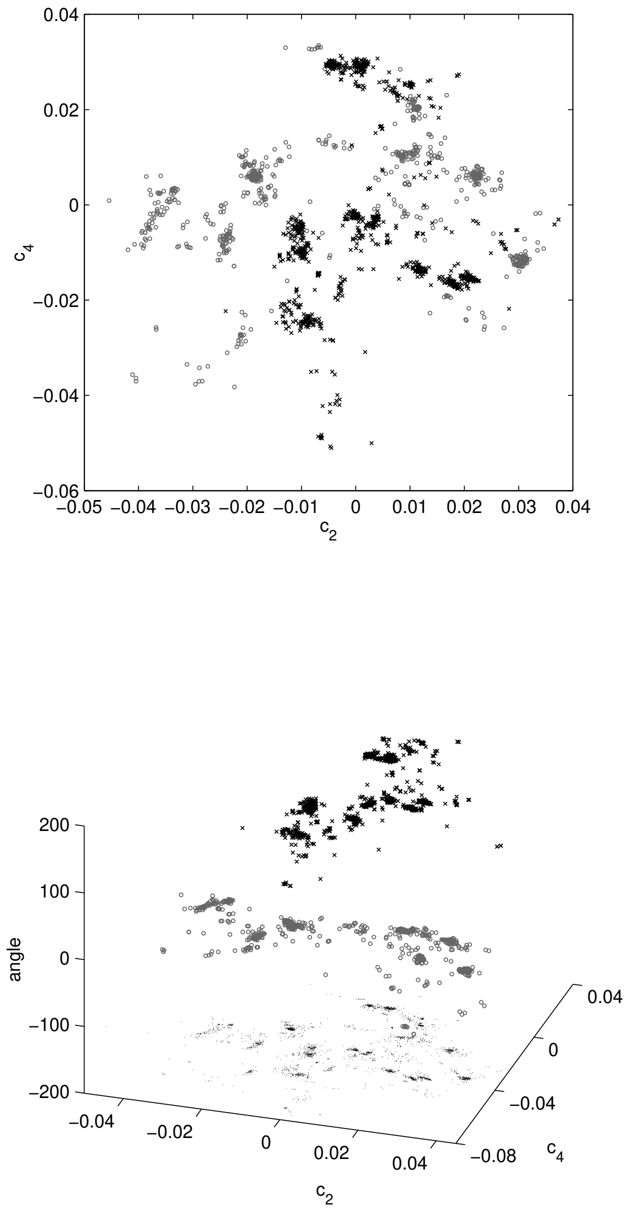


Figure 8: Plots of (c_2, c_4, α) for the globin dataset. This combination had $r' = 0.8237$ and $r^2 = 0.0298$. The top figure shows a projection into the (c_1, c_3) plane. As seen in figure 7, the angle roughly divides into two regions or high and low values. On the figure the high values are represented by crosses, the low value by circles. The bottom graphs shows a three dimensional representation of the same data (the relationship is much easier to see in color graphs). There is a clear grouping of the data into regions with high and low value for the angle, which explains the high r' value. However, there is no linearity at all, thus the low r^2 value.

the (c_1, c_3) space. The r^2 value for the regression is not quite as high as one might expect. This is due to the fact that the angle is a periodic measure, and the change from -180 to 180 occurs in the center of the graph. The origin of the angle could be shifted such that the relationship becomes globally very close to linear. There are a lot of cases for which the KNN method indicate a strong association between the MDS coordinate pair and the auxiliary feature, but the r^2 value for the regression is very low for the same combination. These are cases where we have a local clustering of the data, and the auxiliary feature varies little within the cluster. At the same time, there is no overall linear relationship between the position of the cluster in the MDS coordinate space and the feature value of points in the cluster. Figure 8, which shows the relation between (c_2, c_4, α), is an example of this case: there are large clusters in which the angle is very similar, but the position of the clusters clearly does not relation linearly to the value of the angle. The new method is somewhat biased towards these cases. However, cases like the one shown in figure 8 show a clear structure and might help in discovering patterns in the data.

5 Discussion

Using a non-parametric learning method such as k-nearest neighbors has several advantages. The main advantage, of course, is that it can find non-linear relationships as easily as linear relationships. A related advantage is that it does not require that an appropriate parametric model be specified. KNN is effective for a broad range of relationships. A third advantage is that KNN forms local models. This is important because some of the patterns to be found in reduced-dimensional representations form small, tight clusters such as those in Figure 8 that would be difficult to model globally.

Often the method finds nested sets of relationships between coordinates and descriptive variables. For example, coordinate c_1 may yield an r' of 0.75 to feature f_a, coordinates c_1 and c_2 together might yield r' of 0.90 to feature f_a, and coordinates c_1, c_2,and c_3 together might yield and r' of 0.95. Adding more coordinates to c_1, c_2,and c_3 might reduce r'. c_1, c_2,and c_3 form a nested set. It is up to the user to decide which element of this nested set best describes a useful relationship. The r' values alone are not sufficient. A small set of coordinates with lower r' might be more useful than a larger set of coordinates with higher r' for interpretation. The goal of the r' association measure to efficiently focus the users attention on the associations that are strongest and thus most likely to be interesting.

6 Conclusion

A problem often encountered when applying dimensionality reduction methods such as Multidimensional Scaling or FastMap to a dataset is finding meaningful interpretations of the resulting low dimension coordinate representation. We have introduced an automatic method for finding associations between these coordinates and a set of descriptive auxiliary features. The method is based on the variance of nearest neighbor prediction from subsets of the coordinate representation to each of the auxiliary attributes.

We have applied the method to synthetic data as well as to two real protein structure datasets. Our experiments show that the method works well and is capable of finding nonlinear relationships when standard methods such as linear regression fail. The method has successfully rediscovered an association that required several days to discover manually, and also found several new strong associations that had been missed by the manual search.

7 Acknowledgments

We thank Paul Hodor, John Rosenberg and Bruce Buchanan for their invaluable help in generating the protein datasets and performing the MDS calculations.

References

[1] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[2] J. L. Bentley. Multidimensional binary search trees in data base applications. *IEEE Trans. Softw. Eng.*, SE-5(4):333–340, July 1979.

[3] I. BORG and J. LINGOES. *Multidimensional Similarity Structure Analysis.* Springer-Verlag, New York, 1987.

[4] T. F. Cox and M. A. Cox. *Multidimensional Scaling.* Chapman & Hall, London, 1994.

[5] M. L. Davison. *Multidimensional scaling.* Wiley, New York, 1983.

[6] C. Faloutsos and K.-I. D. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *In Proceedings of ACM SIGMOD*, pages pages 163–174, San Jose, CA, May 1995.

[7] H.M.Berman, J.Westbrook, Z.Feng, G.Gilliland, T.N.Bhat, H.Weissig, I.N.Shindyalov, and P.E.Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[8] P. G. Hodor, R. Caruana, E. Sassaman, B. G. Buchanan, and J. M. Rosenberg. Examination of amino acid sequence rules for alpha helix pairing in proteins. In *International Conference on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, 1999.

[9] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[10] C. King and D. T. (eds.). *Individual differences in perceptions and preferences among nations.* American Merketing Association, Chicago, 1971.

[11] J. B. Kruskal and M. Wish. *Multidimensional scaling.* Sage Publications, Beverly Hills, 1978.

[12] Nene and Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEETPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997.

[13] S. Schiffman, M. Reynolds, and F. Young. *Introduction to Multidimensional Scaling.* Academic press, New York, 1981.

[14] M. Wish. Comparasions among multidimensional structures of nations based on dfferent measures of subjective similarity. *General Systems*, 15:55–65, 1970.

[15] Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1993.