Exploratory Analysis of Point Proximity in Subspaces

Tin Kam Ho
Bell Labs, Lucent Technologies
700 Mountain Avenue, 2C425, Murray Hill, NJ 07974, USA
tkh@research.bell-labs.com

Abstract

We consider clustering as computation of a structure of proximity relationships within a data set in a feature space or its subspaces. We propose a data structure to represent such relationships, and show that, despite unavoidable arbitrariness in the clustering algorithms, constructive uses of their results can be made by studying correlations between multiple proximity structures computed from the same data. We describe a software tool that facilitates such explorations and example applications.

1 Introduction

In unsupervised learning many methods have been proposed to find clusters in a data set [5][10][12], but there has been little emphasis on systematic use of the resulting clusters beyond the discovery of their existence and confirmation of their validity. Methods for connecting a clustering study to the analysis of other characteristics of the data are relatively undeveloped. Motivated by several applications in science and engineering, we explored ways to extend the use of clustering that reveal new opportunities for pattern recognition methodologies. In this paper we discuss some of these uses and describe a software tool that facilitates such explorations.

The study began with the observation that, in many real world applications, objects under study can be described by many features that are not necessarily measured on a single scale. In addition to numerical features on incomparable scales, objects may be described by ordinal or categorical features. Most algorithms in statistical pattern recognition do not apply directly to measurement spaces of mixed scales, and there are no easy ways to define and interpret a global similarity measure that is a function of all such measurements.

Nevertheless, natural metrics may exist for groups of related measurements. In the subspaces spanned by those measurements, the traditional algorithms are applicable. Thus the challenge is how to use clustering results from those subspaces in a larger context of study involving all the relevant measurements.

The necessity of such explorations is highlighted by the following question asked recently by a practitioner, after spending much effort on estimating a finite mixture model: “Now that we have all these Gaussians, what should we do with them?”

2 Clustering as Proximity Analysis

Clustering analysis is plagued with arbitrariness, in the choice of the validation criterion, or in the number or shape of clusters one is willing to accept or impose on the data. A goal of our study is to make constructive uses of clustering results despite such arbitrariness. Towards this we consider a view of clustering that emphasizes its function of computing a *proximity structure* from the data.

This view is motivated by the fact that many clustering methods depend on a similarity metric. Even in popular probabilistic model-based methods that estimate mixtures of Gaussian distributions, location of the mean of each component distribution and the dispersion of the data around the mean are of primary concern.

Therefore a starting point to relate clustering analysis to other data characteristics is to consider clustering as a procedure to construct a data structure that highlights the proximity relationship among the data points. From this perspective, clustering algorithms differ just by the particular structure they compute using different metrics, models, or different scales of resolution. We then investigate the commonalities of such structures and develop measures, algorithms, and tools to relate alternative structures computed from the same data.

We define a *proximity structure* P of a d-dimensional data set $D = \{x | x = (x_1, ..., x_d)\}$ to be a pair $P = (S, G)$, where S is a set of subsets in D, G is a weighted graph (S, E), and the weights of edges in E are values of a function of proximity between two elements of S. We limit our discussions to two types of structures as follows:

1. partitional structures, where S is a partition of D, i.e., S consists of nonintersecting subsets of D whose union equals D, and G is a graph where the nodes are elements of S;

2. hierarchical structures, where G is a tree that splits the root D into a set of leaves that partition D, and S is the set of all the nodes in G between and including the root and the leaves.

Note that the partitional subsets s in S can be singletons containing only one point in D. The length (weight) of an edge in E is the value of a proximity function p that is a notion of distance between two subsets, and may involve scatter within the subsets. For singletons it is the distance between two points. G need not be fully connected if p is undefined between certain elements of S.

An example of a partitional structure $P^p = (S^p, G^p)$ has S^p being the clusters computed from a k-means procedure using Euclidean distance, and G^p being a minimum spanning tree connecting the centroids of those clusters (see Figure 2). An example of a hierarchical structure $P^h = (S^h, G^h)$ is the tree resulting from a complete-linkage agglomerative procedure applied to elements of S^p in P^p, with G^h representing the tree and S^h containing all the nodes in the tree. The edge lengths in G^h are distances between the centroids of parent subsets and those of their children.

Naturally this view of clustering is helpful in cases where meaningful metric exists only in certain subspaces, so that in each subspace a different proximity structure can be constructed. In addition, for each numerical or ordinal feature, there exists a trivial partitional structure where S is the set of all singletons in D, and G is the linear ordering defined by the value of the feature. If the range of values is divided into regular intervals, points in each interval can be taken as a cluster, and representative values of each interval can be ordered to create a relationship graph. For a categorical feature, a degenerate structure exists where S contains the sets of points in each category and the edge set E of G is empty.

Subspaces defined in other ways can be converted to coordinate subspaces if the feature space is augmented to include variables defined by specific projections and transformations of the raw features. Functional dependences can be represented in a similar way. External criteria for cluster validation can be included by augmenting the feature space to include variables defining such a criterion. To represent soft partitions computed by some algorithms, one can first harden the cluster memberships and/or extend the partitional structure to allow intersecting subsets. Finally, while the (S, G) representation is motivated by clustering, it can describe similar structures resulting from other processes, such as classification and regression trees [1] where the structure is related to that of a categorical feature or response variable by construction.

3 Exploration in Proximity Correlations

Given our broad definition of proximity structures, it is obvious that many such structures can be constructed from an arbitrary data set. Our task is to study the relationships among multiple structures computed from the same data. In the literature this has been addressed as interpretation, comparison [4], and combination [11] of a multiplicity of clustering results. One approach defines a global similarity metric as a simple function of the subspace metrics, and uses it to obtain clusters in the full space. Such combination functions need to be carefully justified. Methods have also been proposed for measuring agreement of partitions [4] or intersection of matched groups [5].

Nevertheless, it is realized that a summarizing measure of agreement is less relevant than an investigation of what is causing the disagreement [5]. Therefore we believe that methods and tools for detailed analyses of the correlation of different proximity structures are of critical importance.

Clusters compress the data for more efficient handling and multi-resolution study. By introducing the relationship graph G, the data can be ordered in a meaningful way by methods of traversal in G. This is especially important for multidimensional subspaces that do not have a unique natural ordering. A particular method of traversal selects paths in G along which one can define measures to quantify characteristics of a proximity correlation, and answer questions such as the followings:

- (Continuity) Do small changes in one subspace induce small changes in another subspace?
- (Monotonicity) Do changes in one subspace always induce changes in the same direction in another subspace?
- (Linearity) Do changes in one subspace always induce changes by the same proportion in another subspace?
- (Connectedness) How far can a cluster expand in one subspace if its members are to stay in the same cluster in another subspace?
- (Intrinsic dimensionality) How many different directions of changes are significant in a particular subspace?

4 A Tool for Interactive Analysis

Many studies in clustering emphasize the importance of visualization and interactive graphical analysis. As a first step to obtain insights for a systematic study of proximity correlations, we adopted this approach and constructed a software tool, **Mirage**, to support such analyses (Figure 1). **Mirage** is written in Java 1.3 with a heavy use of the Swing library, and can be run on both Unix and Windows platforms that have an implementation of the Java Virtual Machine.

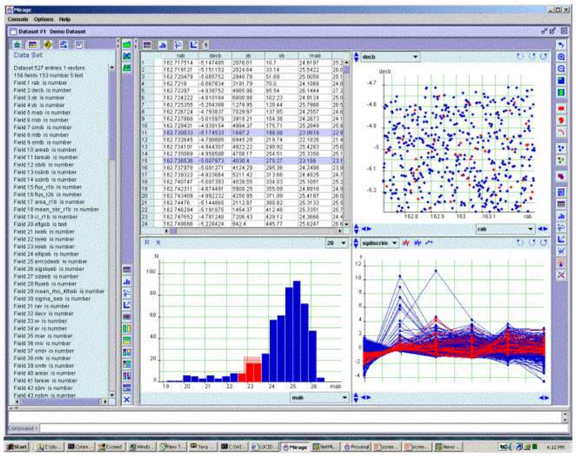


Figure 1. A screen shot with a highlighted subset in the four raw data displays. The subset is selected in the histogram display and broadcasted to other views.

Mirage contains displays of data points projected to one, two, and multidimensional subspaces, as well as clusters in such subspaces along with their relationship graphs. Most importantly, connections between all the displays are maintained, in a way that subsets selected in one view can be broadcasted and tracked in all other views. This is achieved by an object-oriented organization of the data set and common interfaces to all the displays. The interface specifies that every display supports manual selection of a cluster, automatic selection of each cluster in turn along a pre-defined path in a relationship graph, broadcasting a selected cluster, a way to highlight a broadcasted cluster, and a way to show membership of every point in a set of partitional clusters.

4.1 Displays of raw data

We assume that a data set is a matrix where points are represented by rows and the features are represented by columns. Several views of the data sets are offered (Figure 1), and intuitive means to interact with the displays are provided:

- A table view of the data matrix, with a color tag attached to each row that shows its membership in partitional clusters. Points can be selected by a mouse dragged to cover the corresponding rows, and highlighted by changing the background color of the rows.

- A histogram plot that can be reconfigured to show a one-dimensional projection of the data set to any single feature with frequencies in a choosable number of bins. The bins form a partition structure that can be traversed in simple left-right paths. Clusters (bins) can be selected with a mouse drawing an interval. Broadcasted selections or partitions are shown by highlighting or coloring each bar in corresponding heights.

- A scatter plot that displays a two-dimensional projection of the data, where the X and Y axes can be chosen to be any of the feature dimensions. Regions in the projection plane can be selected by drawing boxes or irregular regions with a mouse. The selected points are highlighted and can be tracked when the plot is reconfigured to show a different pair of features, or broadcasted to other plots.

- A feature vector plot that is also known as a plot of *profiles*[2], *distribution maps*[7], or *parallel coordinates*[9]. This plot shows the projection of the data to a multi-dimensional subspace by plotting the value of every feature against the index of that feature in the subspace. That is, a point projected to a subspace of m dimensions as $(z_1, ..., z_m)$ is shown as a polygonal line with nodes marked at (i, z_i) for each $i \ \ (1 \leq i \leq m)$. This plot is a natural display for vectors such as a spectrum or a time series. Vectors of measurements on incomparable scales need to be first standardized so that each component has mean 0 and standard deviation 1. Data can be selected and broadcasted from this plot by drawing intervals in each feature dimension and composing unions or intersections of such intervals. Highlights and partitions are shown by coloring the lines. The plot can be reconfigured to show vectors in different subspaces with the selections preserved.

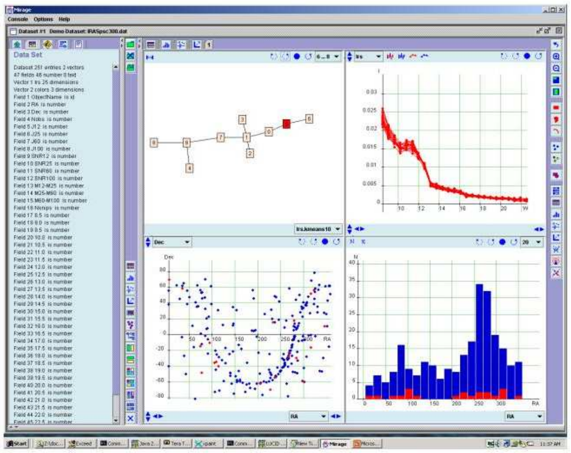
4.2 Displays of nontrivial proximity structures

- A cluster can be shown in the context of the entire data set by highlights in the plots. Members of a cluster can also be shown in isolation in any of the four raw data displays.

- Partitional clusters are shown by colors in each display.

- Hierarchical clusters are shown in a tree panel resembling popular displays of file trees. Nodes can be selectively expanded or closed to show more or less details. Clusters corresponding to selected nodes can be broadcasted, and nodes can be tagged with colors to show broadcasted selections or partitions.

- Relationship graphs are shown in two dimensions using a spring-model layout algorithm that positions the nodes to best preserve the edge lengths. Clusters represented by the nodes can be selected and broadcasted to other plots, or painted in color according to broadcasted selections or partitions (Figure 2).



Mirage

Dataset #1 Demo Dataset: IRASpsc300.dat

Data Set

Dataset 251 entries 2 vectors
47 fields 46 number 0 text
Vector 1 lrs 25 dimensions
Vector 2 colors 3 dimensions
Field 1 ObjectName is id
Field 2 RA is number
Field 3 Dec is number
Field 4 Nobs is number
Field 5 J12 is number
Field 6 J25 is number
Field 7 J60 is number
Field 8 J100 is number
Field 9 SNR12 is number
Field 10 SNR25 is number
Field 11 SNR60 is number
Field 12 SNR100 is number
Field 13 M12-M25 is number
Field 14 M25-M60 is number
Field 15 M60-M100 is number
Field 16 Nmps is number
Field 17 8.5 is number
Field 18 9.0 is number
Field 19 9.5 is number
Field 20 10.0 is number
Field 21 10.5 is number
Field 22 11.0 is number
Field 23 11.5 is number
Field 24 12.0 is number
Field 25 12.5 is number
Field 26 13.0 is number
Field 27 13.5 is number
Field 28 14.0 is number
Field 29 14.5 is number
Field 30 15.0 is number
Field 31 15.5 is number
Field 32 16.0 is number
Field 33 16.5 is number
Field 34 17.0 is number
Field 35 17.5 is number
Field 36 18.0 is number
Field 37 18.5 is number
Field 38 19.0 is number
Field 39 19.5 is number
Field 40 20.0 is number
Field 41 20.5 is number
Field 42 21.0 is number
Field 43 21.5 is number
Field 44 22.0 is number
Field 45 22.5 is number

Figure 2. A screen shot showing a set of partitional clusters linked by an MST, and a step on a leaf-to-leaf path in the MST being broadcasted to other views.

4.3 Exploration algorithms

With this infrastructure many exploration tools can be implemented. Supported operations include manually selecting a subset and tracking it in other views, automatically stepping through each subset along a predefined path and tracking the movement in other views, and manually or automatically reconfiguring a specific plot to show other subspaces and track the location of a selected subset. These operations give visual answers to the questions posed in Section 3. From these one can easily detect regularities or anomalies in the data or in the correlations of structures.

Besides these basic operations, one can construct facilities to define new subspaces and import new features or data points, match a selected subset with comparison data such as samples from a known distribution or points predicted by a theoretical model, select or track the data in the context of a background image such as a map or a photograph, compare the data to a geometrical object projected to the same subspace such as a regression line or a hyperplane, and execute a command script containing a prespecified sequence of operations. While some of these techniques have been attempted before (e.g. slicing data by intervals in subspaces as in the Trellis graphics system [3]), it is obvious that our setup opens up many interesting new possibilities in data analysis.

5 Applications

We have used **Mirage** and its predecessors to study observation or simulation data from many areas of science and engineering. It has enabled perturbation studies of several complex models of physics in computational photonics that yielded new designs of optical fibers and identified practical architectures of Raman amplifiers. It revealed relationships between failure rates and usage patterns of a wireless system, and served to monitor IP traffic in an MPLS-based network management system. It helped in a diagnosis of problems in the data pipeline of an ongoing astronomical survey. Other uses were found in traditional areas including biometrics, image and speech processing, and document analysis. In supervised learning, **Mirage** has been used to study the geometry of various data sets to suggest classifiers, and to relate data characteristics to classifier performance [6][8].

6 Conclusions

We investigated methods for using cluster analysis in the context of other measurements and structures in a data set. We proposed a representation of clustering results that enables interesting analysis of the regularities and anomalies in multiple proximity structures arising from different choices of features, metrics, scale types, shape models, algorithms, and resolution. The proposed analysis framework is implemented in a software tool that has found many interesting uses with data from observations and simulations in several areas of science and engineering.

Acknowledgements

Motivations for the project and the early prototypes were developed in the course of a joint work with Lawrence Cowsar, who contributed many feature suggestions. Discussions with Anil Jain, George Nagy, and visualization suggestions from Bill Cleveland and Michael Andrews are gratefully acknowledged.

References

[1] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*, Chapman & Hall, 1984.

[2] J.M. Chambers, W.S. Cleveland, B. Kleiner, P.A. Turkey, *Graphical Methods for Data Analysis*, Duxbury Press, 1983.

[3] W.S. Cleveland, *Visualizing Data*, Hobart Press, Summit, NJ, 1995.

[4] L.A. Goodman, W.H. Kruskal, Measures of association for cross classification, *JASA*, **49**, 1954, 732-764.

[5] A.D. Gordon, *Classification: Methods for the exploratory analysis of multivariate data*, Chapman and Hall, 1981.

[6] T.K. Ho, A data complexity analysis of comparative advantages of decision forest constructors, *Pattern Analysis and Applications*, to appear.

[7] T.K. Ho, H.S. Baird, Pattern Classification with Compact Distribution Maps, *Computer Vision and Image Understanding*, **70**, 1, April 1998, 101-110.

[8] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, *IEEE Trans. on PAMI*, **24**, 3, March 2002, 289-300.

[9] A. Inselberg, B. Dimsdale, Multidimensional lines I: Representation, Multidimensional lines II: Proximity and Applications, *SIAM J. of Applied Mathematics*, **54**, 2, April 1994, 559-577, 578-596.

[10] A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.

[11] Y. Qian, C.Y. Suen, Clustering combination method, *Proc. of 15th ICPR*, Barcelona, Spain, Sep. 3-7, 2000, 736-739.

[12] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.