Analysis and practical results of U*C clustering

Alfred Ultsch
Databionics Research Group,
University of Marburg, Germany
ultsch@informatik.uni-marburg.de

Abstract.
U*C is a recently proposed clustering algorithm using Emergent Self-Organizing Maps (ESOM). U*C clustering is superior to standard clustering algorithms such as K-means and hierarchical clustering. This work illumintaes the advantages and limitations of U*C clustering compared to other clustering algorithms. Some insights into the principles of ESOM/U*-maps visualization and clustering techniques are also presented.

1 Introduction

Contrary to common belief, Self-Organizing Maps (SOM) are not well suited for clustering. If each neuron on the grid of output neurons is identified with a cluster, it is easy to show that the SOM algorithm is nothing more than a variant of k-means clustering. SOM in the form of emergent SOM (ESOM) (Ultsch 1999) are projections from the high dimensional data space to the neuronal grid in two dimensions. U-matrix and P-matrix (Ultsch (2003)) display distance and density structures on top of this grid. Recently an automatic clustering algorithm, called U*C, has been introduced which uses ESOM (Ultsch (2005)). The “fundamental clustering problems suite” (FCPS) is a collection of data sets which poses canonical problems to any clustering algorithm. With this suite it has been shown that U*C is superior to other standard clustering algorithms like SingleLinkage, Ward or K-means. (Ultsch 2005). This work reports on some results of U*C on a blind experiment with synthetic data and on real world data.

2 U*C Clustering Algorithm

In the following the formation of a topological correct ESOM and the existence of suitable a U- and P- matrix on top of the ESOM neuron grid is assumed (Ultsch/Herrmann 2005). A topological correct mapping $m:R^n->G$ projects a cluster of data onto a coherent area on the grid G (cluster area). Points within the cluster are mapped to the inside of the cluster area. Data points at the border (surface) of the cluster are projected to the border of the cluster area. Consider a data point x at the surface of a cluster C, with $n_i = bm(x)$. The neighbours N(i) of n_i are either within the cluster C or in a different cluster or interpolate between clusters. If the inter cluster distances are locally larger then the local inner cluster distances, then the U-heights in N(i) will be large in such directions which point away from the cluster C. This means, a gradient descent on the U-Matrix will lead away from cluster borders. A movement from one position n_i to another position n_j with the result that n_j is more within a cluster C than n_i is called immersive. For data points well within C, a gradient descent on a U-Matrix will, however, not necessarily be immersive. The P-heights of the P-Matrix follow the density structure of a cluster. Under the assumption that the core parts of a cluster are those regions with largest density, a gradient ascent on the P-Matrix is immersive. Clusters may also be defined by density alone instead of distance. See, for example, the EngyTime data set of the Fundamental Clustering Problem Suite (FCPS) (http://www.mathematik.uni-marburg.de/~databionics/). This data set

represents situations where the data can be described appropriately by overlapping Gaussian Mixtures.
At the borders of a cluster the measurement of density is, however, critical. At cluster borders the local density of the points should decrease substantially. In most cases the cluster borders are defined either by low point densities or by “empty space” between clusters (= large inter cluster distances). For empirical estimates of the point density a gradient ascent on a P-Matrix may therefore not be immersive for points at cluster borders. A movement on a grid which follows first a gradient descent on a U-Matrix and then a gradient ascent on a P-Matrix is called immersion. Let I denote the end points of immersion starting from every position on a grid. If the density within a cluster is constant, immersion will not converge to a single point for a cluster for all starting points within a cluster. The U*-Matrix is then used to determine which points in I belong to the same cluster. The watersheds of the U*-Matrix are calculated using the algorithm described in Luc/Soille (1991). Points that are separated by a watershed are assigned to different clusters, points within the same basin to a single cluster. The following pseudocode summarizes the U*C clustering algorithm described above.

U*C clustering Algorithm: given U-Matrix, P-Matrix, U*-Matrix, I = {};

Immersion:

For all positions n of the grid:

1) from position n follow a gradient descent on the U-Matrix until a minimum is reached in position u
2) from position u follow a gradient ascent on the P-Matrix until a maximum is reached in position p.
3) $I = I \cup \{p\}$; Immersion(n) = p.

Cluster assignment:

1) calculate the watersheds for the U*-Matrix (e.g. using Luc/Soille (1991)).
2) partition I using these watersheds into clusters $C_1,\ldots C_c$
3) assign a data point x to a cluster C_j if Immersion(bm(x)) $\in C_j$.

3 U*C on synthetic data

The efficiency of the U*C clustering algorithm has been demonstrated on the Fundamental Clustering Problem Suite (FCPS) (FCPS can be downloaded from following website: www.informatik.uni-marburg.de/~databionics. FCPS poses some canonical clustering problems. The following table gives an overview on the data. The data sets of FCPS are selected such important problems of Clustering are addressed. For example, there may be clusters of different inner cluster densities and inner cluster variances. The problem of outliers and linear not separable clusters is addressed. Some data sets stress the limits of cluster separation in situations when the cluster almost touch or are defined by a local minimum in density. Very special is The GolfBall data set. It consists of a 3 dimensional data set where each point has the same distance to its neighbours. This data set tests the reaction of the clustering algorithm to situations where no cluster exists.

Name	n	d	k	Main Problem
Hepta	212	3	7	different densities
Lsun	400	2	3	different variances
Tetra	400	3	4	inner vs inter dist.
Chainlink	1000	3	2	not linear separable
Atom	800	3	2	linear not separable, different densities
EngyTime	4096	2	2	density defined clusters
Target	770	2	6	outliers
TwoDiamonds	800	2	2	touching clusters
WingNut	1070	2	2	densities at borders
GolfBall	4002	3	1	no cluster at all

The results of U*C clustering with comparison to the pre-known clustering on FCPS are given in the following table. Shown is the total accuracy i.e the percentage of data points that are clustered correctly. U*C determines the number of clusters by automatically, for the others the correct number of clusters is an input to the algorithm

Data Set	Single	Ward	K-means	U*C
Hepta	100 %	100 %	100 %	100 %
Lsun	100 %	50 %	50 %	100 %
Tetra	0.01 %	90 %	100 %	100 %
Chainlink	100 %	50 %	50 %	100 %
Atom	100 %	50 %	50 %	100 %
EngyTime	0 %	90 %	90 %	90 %
Target	100 %	25 %	25 %	100 %
TwoDiamonds	0 %	100 %	100 %	100 %
WingNut	0 %	80 %	80 %	100 %
GolfBall	100 %	0 %	0 %	100 %

Performances lower than 80% are emphasized. There is no data set on which U*C performs worse than any of the other clustering algorithms.

4 A blind experiment on synthetic data

In fall 2005 the working group of data analysis and numerical classification of the GfKl (AG DANK) performed a blind experiment. A set of 11 synthetic data sets with a known cluster structure were published without the a priori clustering. This suite was clustered using U*C, Single Linkage, Ward and k-means clustering. The following table summarizes the results. Except U*C all other clustering algorithms needed an estimation of the number of clusters. The results shown are on the basis of the given number of clusters except for U*C which estimates this number by itself.

Data Set	Single	Ward	Kmeans	U*C
dankdata1	83	83	83	100
dankdata2	27	85	85	83
dankdata3	56	49	71	83
dankdata4	81	99	99	99
dankdata5	26	83	82	92
dankdata6	100	100	99	99
dankdata7	78	91	64	78
dankdata8	63	63	63	91
dankdata9	65	81	81	99
dankdata10	100	77	67	100
dankdata11	28	84	91	67

The results are given in percent accuracy. For the data set dankdata11 U*C identified only 4 instead of 5 clusters. In all other cases the correct number of clusters were found.

5 Real world data:Protein Cavities

Many biochemical pathways are catalyzed and regulated via the complementary recognition properties of proteins and their substrates. The ligand accommodates the binding cavity of the protein according to the lock-and-key principle. If two binding cavities have common substructures, it can be assumed that the two active sites are capable to bind similar ligands and thus exhibit related function. The following picture shows a ligand inside a cavity.

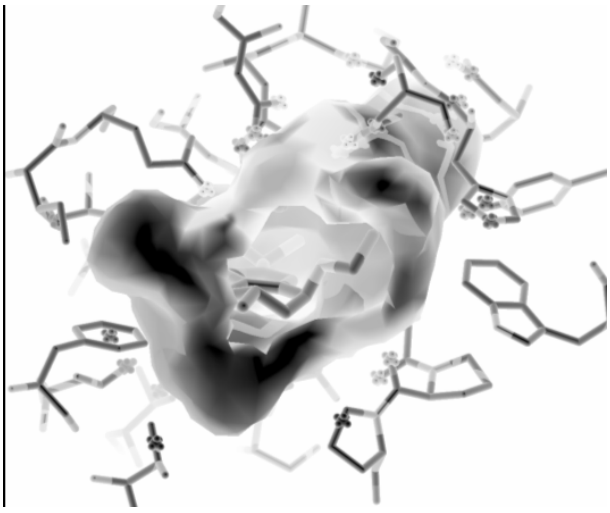

Figure 1: Binding cavity of an enzyme with ligand inside

Enzymes are a particular important class of biochemical agents. Enzymes can be classified with respect to their function and the bound ligand, each enzyme has a specific EC-number. Enzymes having the same EC-number are very similar, those having different EC-numbers have different cavities. The clustering task is to find common substructures within different cavities in order to identify a common functionality. This work is a collaboration with Katrin Kupas . of our institute and the institute of Pharmaceutical Chemistry of the University of Marburg (Kupas et al (2004)). A U*C clustering of 774 binding pockets was performed. For these enzymes the EC class numbers were known. The enzymes belong to 13 different EC classes. The U*C clustering of these enzymes resulted in an accuracy of 98.3 % compared to

the true enzyme classes. Details of the clustering of enzymes are published in (Kupas et al (2006)).

6 Discussion

The definition of a cluster strongly relies on constraints on distances or densities within a particular data set. A clustering algorithm produces meaning ful results, if the underlying model of the clustering algorithm, e.g. spherical for k-means, fits to the data's structure. U*C relies on ESOM for a projection on a surface such that the distance and density structures of a data set are preserved sufficiently. This property may be measured with the methods proposed in (Ultsch/Herrmann (2004)). By using the U- and P- matrix on top of the neuron grid U*C exploits both distance and density information for the stand-alone formation of clusters including an estimation of the number of clusters.

The performance of U*C on FCPS shows, that data sets that are hard to cluster can be coped with. It is astounding that popular cluster algorithm like Ward or K-means show a bad performance on simple clustering tasks such as the LSun, Atom or Target example. The disentangling of intertwined but separate clusters seems to be a unique feature of U*C. This was demonstrated with the ChainLink data set.

In a blind experiment with the AG-DANK synthetic data U*C found in 10 of 11 cases the correct number of clusters. In only one case (dankdata7) another clustering algorithm (Ward) clearly outperformed U*C. In 2 cases U*C surpassed the other algorithms, in 5 cases the results were equal to the best other algorithm. In 3 cases U*C was second best.

In summary: U*C is a useful clustering algorithm which is a sometimes better alternative to conventional clustering algorithms. In particular the stand alone selection of the number of clusters is remarkable. The strength of U*C seems, beside the disentangling of complex cluster structures, the usage of both density and distance information. A first result on a difficult real world problem with known clustering gave an excellent performance. This demonstrated that U*C is ready to use for important practical problems.

7 Conclusion

A new clustering algorithm based on grid projections is proposed. This algorithm uses distance structures (U-Matrix) as well as density structures (P-Matrix) of the data set. No particular geometrical cluster model is imposed on the data by U*C. Other clustering algorithms impose such a model and are performing poor, even on simple synthetic data, which does not follow this model.

The number of clusters is determined automatically in U*C. The correctness and validity of the clusters found can be assessed directly using the U- U*-and P-Matrix visualization (Ultsch (2003)). The U*-Matrix shows a combined picture of distance and density structures of a high dimensional data set.

U*C performs superior to standard clustering algorithms such as K-means and the most popular hierarchical algorithms, see (Jain/Dubes (1998)). This is demonstrated on a group of data sets which represent fundamental clustering problems, like different variances, outliers and other structural difficulties and on a blind experiment with synthetic data. U*C and other tools for ESOM, see (Ultsch/Mörchen (2005)), can be downloaded from our web site.

References

A.ULTSCH,A., MÖRCHEN,F. (2005): ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM, Dept. of Computer Science University of Marburg, Research Report. 46.

JAIN, A.K., DUBES, R.C. (1998): Algorithms for Clustering Data, New York, Wiley.

KASKI ET AL (1999): Analysis and Visualisation of Gene Expression Data using Self Organizing Maps, Proc NSIP.

Kupas, K. et al (2004) : An algorithm for finding similarities in protein active sites, In Matthew He, Giri Narasimhan, Sergei Petoukhov (Eds), Advances in Bioinformatics and its Applications, Proceedings of the International Conference, Nova Southeastern University, Fort Lauderdale, Florida, USA, World Scientific, pp. 373-380

Kupas, K. et al (2006) Classifcation of substructures in protein binding cavities using wavelets, to appear.

LUC,V, SOILLE, P. (1991): Watersheds in Digital Spaces: An Efficient Algorithm Based on Im-mersion Simulations, IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol. 13(6), 583-598.

ULTSCH, A., VETTER, C. (1994): Selforganizing Feature Maps vs. statistical clustering, Dept. of Computer Science University of Marburg, Research Report 9.

ULTSCH,A. (2003): Maps for the Visualization of high-dimensional Data Spaces, In Proc. WSOM, Kyushu, Japan, 225-230.

ULTSCH,A., (1999): Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series, E. Oja and S. Kaski (eds), Kohonen Maps, 33-46.

ULTSCH,A., (2003): U*-Matrix: A Tool to visualize Clusters in high dimensional Data, Dept. of Computer Science University of Marburg, Research Report 36.

ULTSCH,A., HERRMANN, L. (2005), The architecture of Emergent Self-Organizing Maps to reduce projection errors, Proc ESANN, Brugges 2005.

ULTSCH,A., SIEMON, H.P. (1990): Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis, In Proc. Intern. Neural Networks, Kluwer Academic Press, Paris, 305-308.