Learning to Visualise High-Dimensional Data

Khurshid Ahmad, Bogdan Vrusias
Dept. of Computing, University of Surrey, Guildford, Surrey, UK.
{k.ahmad@surrey.ac.uk, b.vrusias@surrey.ac.uk}

Abstract

Visualisation techniques focus on reducing high dimensional data to a low dimensional surface or a cube. Similar dimensional reduction is attempted in the so-called 'self-organising maps'. A number of techniques have been developed to visualise categories learnt by these maps through and exemplified by the term sequential clustering. An evaluation of the techniques is presented using the learning capability of the self-organising maps as a baseline for building systems that learn to visualise complex data.

1. Introduction

A self-organising feature map ('SOFM') is a neural computing program, and its output, used typically for categorising complex multi-dimensional data. An SOFM not only computes the category of objects in a dataset but also learns to compute the same. Typically the system is given an *n*-dimensional input, representing a set of objects in the dataset, which an SOFM, through a process of regression, *maps* onto a two-dimensional surface. Each of the *n* dimensions refers to a property of the objects that either relates to features it may share with other objects in a given category together with features that may distinguish it. The SOFM is a neural network usually with two layers: the *n*-dimensional input layer and the two-dimensional output layer. The objects in the datasets are 'won over' by nodes in the output layer in a *winner-takes-all* manner, and one node may be associated with more than one object in the training dataset, and some nodes may not be able to win over any of the objects. If categorially similar objects occupy a neighbourhood of the nodes, then the SOFM folk call it a 'cluster' – and a number of examples show this to be a demonstrable fact [1].

Once the map is created, and presented with an object, particularly one it wasn't trained on, the system then assigns a category to the novel or unknown object. One of the key advantages of using the SOFM is that it preserves the topology of the input data despite reducing it onto a surface. This topology preserving property of the maps is worth noting. The category information is not directly available, in that, like many other neural computing systems, the output is not as discernible as may be the case for other learning systems like *k-means* or *hierarchical clustering*. This is one significant drawback in this otherwise versatile and novel categorisation system.

More recently, one sees projects that use the output of an SOFM, or related techniques, as an input to a traditional clustering system. Authors variously use the terms 'road map of the data space' when they describe a classification produced using an SOFM [1][5], and the 'floor space layout of the original data set'. The visually indiscernible categories produced by an SOFM have been examined by looking at the variance between and across neighbourhoods of proximate nodes [6].

We begin by reviewing the work of Ultsch who has been working on the so-called *U-matrix* for producing visually discernible categories of complex data (section 2). This is followed by some of our work on using complex organisations of neural networks – the so-called multi-net systems for classifying data which, in its natural form, may exist in more than more modality of communication, for example image and text (section 3). We briefly evaluate the efficacy of sequential clustering, or using the output of a trained neural network as 'floor space layout of a complex data set' (section 4). Section 5 concludes this paper.

2. Visualising clusters

Ultsch and colleagues have been working on "the local distance structures of [a complex] […] dataset" [5], which they claim has the advantage of "a non-linear disentanglement of complex cluster structures". Ultsch takes a complex dataset, for example the Iris dataset, and some recently created genomic data, categorises it using a self-organised map. This map has one major disadvantage in that the nodes on the borders of the two-dimensional output map have different cartographic properties than, say, the nodes in the centre. This is an artefact of the reductionist strategy used in order to create the map and has been discussed in some detail by Kohonen himself. The U-matrix is an attempt to create a borderless manifold which produces a map with the cartographic qualities.

The U-matrix is derived from a map and essentially manipulates the distance between two neurons and the map

Proceedings of the Eighth International Conference on Information Visualisation (IV'04)
1093-9547/04 $ 20.00 IEEE

and the weight associated with each node in the output map. There are claims that using U-matrix as a second sequential clustering technique enhances the effectiveness of using SOFMs considerably. In the discussion of the Iris dataset, the claim is that "an U-matrix clustering of the Iris data coincides with prior classification at least as much as a clustering with the WARD hierarchical clustering algorithm". The U-matrix has been used to visualise datasets as diverse as sea-level prediction to customer relation management and from genomic analysis to stock portfolio selection.

Another visualization technique that displays SOM's clusters is ViSOM [2]. ViSOM was developed to overcome the U-matrix shortcomings. The algorithm is similar to the Sammon mapping which produces graded mesh that preserves the distances between the data points on the map as well as the topology. ViSOM however is much simpler in computational complexity than Sammon mapping.

The U-matrix shows the local density structures on a topology preserving transformation of high dimensional dataspace onto a two-dimensional map. A variant of U-matrix is U*-matrix which combines distance and density information. We have compared the performance of SOFM/U-matrix classification with SOFM/k-means classification and have found that the latter has a better performance especially when we deal with images or texts.

3. Unimodal and multimodal classifiers

3.1. A brief note on the architecture of unsupervised classifiers

Self-organising feature maps have been used extensively for classifying text despite the fact that the creator of these maps has argued that "it may sound surprising that vector space methods such as SOFM can handle structured, discrete, linguistic data" [1].

Kohonen's key project, WEBSOM, is an unsupervised text classifier that has been described variously as a schema, content-addressable memory, method and architecture. A collection of texts that is to be categorised is analysed for important keywords and a *vector* is created to represent texts in the collection. There are many ways of constructing this vector, suffice it to say that a group of between 10 to 100 component vector is selected and each component relates to the presence and absence of 10 to 100 keywords respectively. The vectors are presented to the input node of an SOFM and the neurons in the output node win over the individual texts to themselves. Hence a category is created by virtue of the fact that a group of proximate nodes may represent documents belonging to one category and not others.

It is not as common to classify images using self-organising feature maps as, for example, the classification of texts. There are many reasons for this and we feel the principal reason may be that the contents of an image really can only be described at the physical level – the distribution of colours, various shapes, different textures, changing levels of illumination and brightness, to name but a few physically perceptible features. It is difficult to describe a football as opposed to a cricket ball merely on the basis of physically perceptible features unless the context is explained or the viewer has experience of the two different kinds of balls. Such context is typically added by text or is the preserve of the viewer.

Notwithstanding the fact that images are quite difficult to describe using physically perceptible features alone, we have attempted to see to what extent an unsupervised classifier can classify individual images. Our intention was twofold: first, to see whether the use of the SOFMs as the floor space layout of the image dataset will help a sequential classifier to classify the images better. Second, we believe that it is a mixture of modalities that allow human beings to act efficiently and intelligently in the world, the physical features may fail to provide clues about the objects in the image, but a description just might succeed. The use of alternative modalities, images and text, for understanding an image (or text) suggests a systems' architecture for a computer-based system for recognising, retrieving and archiving such images: consider a system which is being trained to classify a special set of images which always have a collateral linguistic description for every image in the collection. Furthermore, two classifiers are used, one to classify images and the other to classify texts. Add to these two classifiers a third classifier that learns the relationship between textual description and the physical features of an image. Once this three-tier system, or a multi-net, is trained, then if we were to bring an image without its collateral text and provided the image is recognised by our system, the system can then generate an approximate description of the image – a system that can automatically produce candidate annotations for this unknown image. This system will help to recall images or to illustrate a text or a set of words by an image. Once these keywords are presented to the system it will retrieve a set of related images. This process can be referred to as auto-illustration.

The reason we are discussing our image and text classifiers, both unimodal and multi-net classifiers, is that we have seen that the use of SOFM as a first-pass classifier followed by automatic classification methods like k-means or U-matrix, appears to improve the performance of our system when compared to the case of unimodal classifiers only or that of automatic classifiers only.

3.2. Training data

We have used a large picture collection, the Hemera set – c.50,000 images, each image accompanied by a set of keywords. We have randomly selected over 1,000 images divided into 10 categories by Hemera human classifiers. We created text and image vectors based on the internal physical features of the image and frequently used keywords in the textual descriptions, but did not use the category information.

EXEMPLAR IMAGE	KEYWORDS	CATEGORY
	billiard ball thirteen orange pool parlor striped Boston stripes sports	***BALLS***
	anea marthesia red butterfly Peru insects black wings	***BUTTERFLIES & MOTHS***
	1967 Plymouth fury police car automobile transportation law officer vehicle drop shadow	***CARS***
	beer brown glass bar drinking foam head suds booze liquor restaurant ale lager	***DRINKS***
	azalea green home house potted plant pink housewarming blooms blooming	***FLOWERS***
	green grape cluster green food vineyard wine	***FRUIT***
	two francs silver change piece cash coins France 2	***MONEY***
	antique sofa comfort couch home sitting burgundy white furniture	***SEATING***
	jet airplane military air force airport white aviation flying army jet pilot fighter transportation	***TRAINS & PLANES***
	bullet gold ammunition war metal military arms ammunition cartridge shell	***WEAPONS***

Table 1 Exemplar images and related keywords of the "Hemera PhotoObjects I" data set.

Ten different categories were selected randomly from the collection, and for each category there was an average of 115 randomly selected images, giving a total of 1151 images. The categories include: *balls*, *butterflies & moths*, *cars*, *drinks*, *flowers*, *fruit*, *money*, *seating*, *trains & planes*, and *weapons*. The terms attached to the images totalled 10,018, and there is an average of 8.7 terms associated with each image. The characteristics of each category in relation to the terminology given by the experts are shown in Table 2.

The two techniques used for generating the feature vectors are *binary* and *logarithmic partitioning* [3]. The first vector type is a binary vector based on keywords extracted from the most highly weighted TF*IDF terms. By eliminating duplicated terms we were left with 195 unique terms, which are the basis for constructing the feature vector (i.e. binary vector). The other three types of vectors are based on partitioning all the terms from each category into logarithmic sets. For each category the total number of terms in that category was divided into sets of terms, starting with a small number in the first set, and exponentially increasing the number of terms in the remaining sets.

CATEGORY	AVER. No TERMS	TOTAL No TERMS	No ITEMS
BALLS	9	915	97
BUTTERFLIES & MOTHS	8	993	129
CARS	10	1217	118
DRINKS	10	664	65
FLOWERS	9	1099	117
FRUIT	4	561	131
MONEY	8	909	120
SEATING	8	862	107
TRAINS & PLANES	11	1542	139
WEAPONS	10	1256	128
AVERAGE	8.7	1,001.8	115.1

Table 2 Characteristics of the selected training and testing "Hemera PhotoObjects I" data set.

Image data vectors have been also based on their characteristics. Four basic characteristics of an image that have been considered are: 21 colour features, 19 edge features, 7 shape features, and 20 texture features. The final vector model contains all the features, 67 in total.

3.2. SOFM clustering and activation

The keywords with the 10 categories, about 1000 keywords per class, help to produce a well clustered text-based feature map for all the clusters except perhaps for class 10 (designated as "weapons"), which appears to be distributed in two clusters (Figure 1).

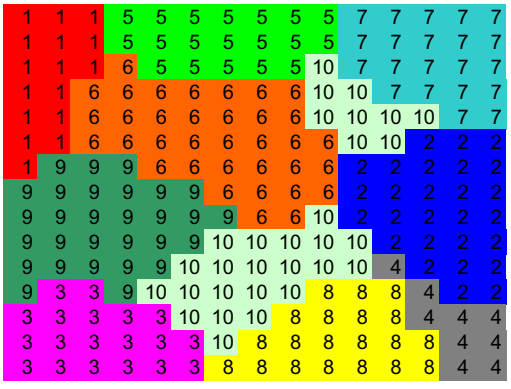

Figure 1 Visualising the clusters formed by the text-based SOFM.

The image features provide a good cluster for only two classes (7 money and 10 weapons), otherwise the feature map has two separated clusters for each class (Figure 2).

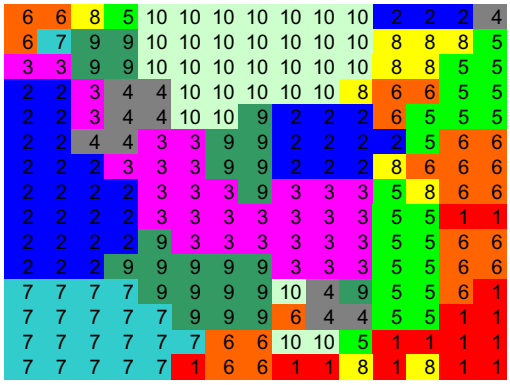

Figure 2 Visualising the clusters formed by the image-based SOFM.

Each node may represent more than one class, but with different activation values. For instance the bottom left hand side node in Figure 2 contains several input vectors that produce a different activation, and some of the nodes represent a different than type "1"class (50% of them). Whereas in the text map (Figure 1), the top left hand side node only contains input vectors from the same class type. That phenomenon can also be seen from the activation given by each individual input vector; the image input vectors have much lower activations when compared to the text input vectors. The differences between the clustering of the two maps can be shown more clearly, but with added information, by looking at a test input vector and the way it activates the map.

3.3. Testing: Topology of the Single-net

Testing was carried out to evaluate the clustering effectiveness of the trained networks. Recall that the entire sample was used for training (1151 images and the collateral keywords). Four different evaluation methods for measuring the classification were used: precision, recall, JAC, and FM; where JAC=TP/(TP+FP+FN), FM=TP$\sqrt{}$((TP+FP)x(TP+FN)), *TP* is true positive, *FP* false positive, and *FN* false negative. The evaluation results are based on 5 different repetitions of the same experiments, and the final values have been averaged.

We tested the topology by looking at the clustering ability of the text SOFM using the U-Matrix method, where the 15x15 map has the best results on all four types of evaluation. The image SOFM's precision is not as good as the text SOFM, but the recall is almost as good. The JAC and FM levels though show how the image SOFM is outperformed by the text SOFM, therefore the high recall of the image SOFM is probably due to the fact that the map has clustered the largest proportion of image vectors into a single cluster.

TEXT U-MATRIX CLASSIFICATION					
Topology	**F0.5**	**Precision**	**Recall**	**JAC**	**FM**
10x10	0.61	0.47	0.87	0.46	0.63
15x15	0.69	0.53	0.99	0.53	0.73
50x50	0.63	0.48	0.91	0.46	0.66
IMAGE U-MATRIX CLASSIFICATION					
10x10	0.18	0.10	0.84	0.10	0.30
15x15	0.19	0.11	0.75	0.11	0.29
50x50	0.18	0.10	0.97	0.10	0.32

Table 3 Clustering ability of different size text SOFMs, trained on 50-dimensional text data or 67-dimensional image data and clustered with the U-Matrix method.

In order to understand the clustering efficiency we can also use K-Means analysis on the SOFM's output. Both U-Matrix and K-Means cluster the SOFM based on the weight vectors that correspond to each node in the output map.

TEXT K-MEANS CLASSIFICATION					
Topology	**F0.5**	**Precision**	**Recall**	**JAC**	**FM**
10x10	0.70	0.60	0.83	0.51	0.72
15x15	0.72	0.63	0.84	0.57	0.73
50x50	0.80	0.70	0.92	0.66	0.80
IMAGE K-MEANS CLASSIFICATION					
10x10	0.25	0.24	0.27	0.15	0.26
15x15	0.26	0.25	0.27	0.15	0.26
50x50	0.29	0.28	0.29	0.17	0.28

Table 4 Clustering ability of different size text SOFMs, trained on 50-dimensional text data or 67-dimensional image data and clustered with the K-Means method.

The first experiment uses the 30-dimensional binary text vectors constructed from the keywords associated to each image description in the data collection. In this task each classifier has performed almost as well, given that each text vector is limited to only 30 dimensions (features). The best average clustering ability was performed by the SOFM clustered with the K-Means method, while the U-Matrix method follows second.

Method	**F0.5**	**Precision**	**Recall**	**JAC**	**FM**
SOFM K-Means	0.80	0.70	0.92	0.66	0.80
SOFM U-Matrix	0.44	0.29	0.93	0.29	0.52
K-Means	0.67	0.59	0.79	0.51	0.68
Fuzzy C-Means	0.63	0.56	0.74	0.47	0.64
Hier. Centroid	0.52	0.40	0.75	0.35	0.54
Hier. Complete	0.56	0.44	0.79	0.40	0.59
Random	0.10	0.10	0.10	0.05	0.10

Table 5 Performance of several classification methods when classifying the 30-dimensional binary text vectors.

When analysing the 50-dimensional binary text vectors, the overall performance of all classifiers seems to

have improved. The best average clustering ability was again performed by the SOFM clustered with the K-Means method. The best statistical clustering is performed by the Hierarchical Centroid method, which outperforms even the SOFM clustered with the U-Matrix method.

Method	F0.5	Precision	Recall	JAC	FM
SOFM K-Means	0.72	0.63	0.84	0.57	0.73
SOFM U-Matrix	0.69	0.53	0.99	0.53	0.73
K-Means	0.56	0.43	0.79	0.39	0.58
Fuzzy C-Means	0.67	0.57	0.81	0.50	0.68
Hier. Centroid	0.70	0.61	0.84	0.54	0.71
Hier. Complete	0.56	0.43	0.80	0.39	0.59
Random	0.10	0.10	0.10	0.05	0.10

Table 6 Performance of several classification methods when classifying the 50-dimensional binary text vectors.

Following the same procedure for the 100-dimensional text vectors, once again the SOFM with the K-Means clustering performed better than any other method, followed by the SOFM with the U-Matrix clustering. The other methods seem to struggle with such high dimensional vectors, because their results have dropped significantly. It looks like the more complicated the data gets, the more the statistical classifiers struggle, whereas the SOFM is not affected.

Method	F0.5	Precision	Recall	JAC	FM
SOFM K-Means	0.81	0.73	0.90	0.67	0.81
SOFM U-Matrix	0.62	0.48	0.86	0.45	0.65
K-Means	0.49	0.35	0.79	0.32	0.53
Fuzzy C-Means	0.45	0.31	0.81	0.29	0.50
Hier. Centroid	0.28	0.17	0.72	0.16	0.35
Hier. Complete	0.42	0.29	0.78	0.27	0.48
Random	0.10	0.10	0.10	0.05	0.10

Table 7 Performance of several classification methods when classifying the 100-dimensional binary text vectors.

Looking at the image vectors it is difficult to judge which method is better than the other due to the fact that the results are poor. During the different runs none of the methods showed any significant difference to the other. Also it is worth mentioning that K-Means clustering applied to SOFM's weight vectors gives better results when compared to applying K-Means to the data vectors directly.

3.4. Sequential information bottleneck algorithm

The problem of clustering large corpora of texts or images has exercised the efforts of the unsupervised document clustering community. The main problem in document clustering is that specially for open-ended document streams, for example news wires, scientific texts, it is not possible to give detailed categorial information about the incoming documents; the point about an open-ended document stream is that we don't know in detail what the category of the document is. Supervised classification methods are not much help because such methods shoe-horn the categorisation process. The supervised method requires the pre-knowledge of the categories and, once defined, these categories are fixed. Thus it is not possible to incorporate new categories unless the training is performed all over again with the new category. Unsupervised classification, on the other hand, offers this facility of incorporating newer categories to areas of proximate older categories.

4. Evaluation

We have indicated above that the classification produced by an SOFM is difficult to surmise except by visual identification or through the use of various clustering techniques (K-Means, U-Matrix, Hierarchical clustering, etc.). An application of K-Means clustering on the output of an SOFM shows how the SOFM has found data in the proximate classes given by the Hemera experts. We have used this sequential clustering method (SOFM followed by K-Means) to examine the clusters of keywords and clusters of visual features: Table 8 shows the clustering of the 1151 images, by a 50x50 SOFM trained for over 1,000 cycles, using keywords to represent the images. Recall that each of the 1151 images has, on average, 8.7 keywords.

	Clusters defined by K-Means on the text SOFM									
CAT.	A	B	C	D	E	F	G	H	I	G
1	44				4				49	
2								129		
3										118
4					65					
5					117					
6			85		46					
7							120			
8						107				
9										139
10		64		64						

Table 8 Distribution of the 1151 text items from the Hemera collection in 10 clusters, based on the K-Means algorithm applied on the text SOFM.

The images that were associated with Hemera classes (*butterflies* and *moths*, *money*, *seating*), and represented through their keywords have all been separately clustered in a single class each (H, G, F). *Cars* and *trains* and *planes* have been put together into one single class (G). Two-thirds of fruits have been assigned to a unique class (C). *Weapons* have been equally divided into two classes (B and D), as is the case with the class balls (A and I).

CAT.	Clusters defined by K-Means on the image SOFM									
	A	B	C	D	E	F	G	H	I	G
1	10	2	17	11	6	7	9		1	34
2	13	6			1	4	21	39	44	1
3	13	12	3	7		1	5	57	18	2
4	6	8	7	5	1	9	10	5	12	2
5	11	11	4	6	2	26	14	10	9	24
6	16	3	12	5	3	35	10	13	15	19
7	14		3		102			1		
8	11	12	5	14	7	9	19	10	15	5
9	16	18	1	5	1	3	5	58	31	1
10	4	77	7	5	1	2	9	6	16	1

Table 9 Distribution of the 1151 image items from the Hemera collection in 10 clusters, based on the K-Means algorithm applied on the image SOFM.

The keywords used with butterflies result in matching with keywords only related to butterflies, and the same is true of a cluster of chairs.

CAT.	F0.5 measure of the clusters defined by K-Means on the text SOFM									
	A	B	C	D	E	F	G	H	I	G
1	62	0	0	0	2	0	0	0	67	0
2	0	0	0	0	0	0	0	100	0	0
3	0	0	0	0	0	0	0	0	0	63
4	0	0	0	0	44	0	0	0	0	0
5	0	0	0	0	67	0	0	0	0	0
6	0	0	79	0	25	0	0	0	0	0
7	0	0	0	0	0	0	100	0	0	0
8	0	0	0	0	0	100	0	0	0	0
9	0	0	0	0	0	0	0	0	0	70
10	0	67	0	67	0	0	0	0	0	0

Table 10 Effectiveness measure for each of the 10 clusters identified by the K-Means algorithm when applied on the text SOFM.

The classification based on visual features of the images has not proven to be clear-cut using this sequential clustering method mentioned above. Table 11 shows the distribution of images. The visual features can only be used to cluster images in the money class in any meaningful fashion in that over 95% were assigned to the class E', although a small number of balls and seating have also been assigned to that class. Half of the cars and trains and planes have been clustered together (class H) but this class also contains one-third of the butterflies and moths and some seating. One-third of the balls, about a quarter of flowers and fruits have been allocated to one class. The rest of the class assignments appear to be almost random. Note that the classes produced by the sequential clustering method (A to G, and A' to G') are arbitrary.

CAT.	F0.5 measure of the clusters defined by K-Means on the image SOFM									
	A	B	C	D	E	F	G	H	I	G
1	10	2	22	14	5	7	9	0	1	37
2	11	4	0	0	1	4	18	24	30	1
3	11	9	3	8	0	1	5	36	13	2
4	7	7	11	8	1	11	12	4	11	3
5	10	8	5	7	2	24	13	6	6	23
6	13	2	13	5	2	31	9	8	10	17
7	12	0	3	0	84	0	0	1	0	0
8	10	9	6	17	6	9	18	7	11	5
9	13	13	1	5	1	3	4	34	21	1
10	3	56	7	5	1	2	8	4	11	1

Table 11 Effectiveness measure for each of the 10 clusters identified by the K-Means algorithm when applied on the image SOFM.

Conclusions

Being able to visualise complex multi-dimensional data is a challenging task. Learning how to visualise such data is even more challenging. We have shown how SOFMs reduce the multi-dimensional data onto a simple plane and how we can automatically visualise the formed clusters of similar data points. We evaluated two methods for visualising SOFMs, k-means and U-matrix, and concluded that k-means can cluster the map better than U-matrix in most of the cases.

References

[1] T. Kohonen, *Self-organising maps*, Berlin, New York: Springer-Verlag, 1997.

[2] H. Yin, "ViSOM—A Novel Method for Multivariate Data Projection and Structure Visualization", *IEEE Transactions on Neural Networks*, Vol. 13, No. 1, Jan 2002.

[3] B. Vrusias, *Combining Unsupervised Classifiers: A Multimodal Case Study*, Unpublished PhD thesis, University of Surrey, Guildford, Surrey, UK. 2004.

[4] N. Slonim, N. Friedman, and N. Tishby, "Unsupervised document classification using sequential information maximization". *Proc SIGIR'02, 25th ACM international Conference on Research and Development of Information Retrieval*, Tampere, Finland, ACM Press, New York, USA, 2002.

[5] A. Ultsch, "Maps for the Visualization of high-dimensional Data Spaces", *Proc Workshop on Self organizing Maps WSOM03*, Kyushu, Japan, 2003, pp 225 - 230.

[6] K. Ahmad, B. Vrusias, A. Ledford, "Choosing Feature Sets for Training and Testing Self-Organising Maps: A Case Study", *Neural Computing and Applications*, Vol. 10, 2001, pp 56-66.