7th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES (AIKED'08),
University of Cambridge, UK, Feb 20-22, 2008

ISSN: 1790-5109 ISBN: 978-960-6766-41-1

Clustering with alternative similarity functions

Wesam Barbakh
University of Paisley
School of Computing
Paisley, PA1 2BE, Scotland
UK

Colin Fyfe
University of Paisley
School of Computing
Paisley, PA1 2BE, Scotland
UK

Abstract: We [6, 7] have recently investigated several families of clustering algorithms. In this paper, we show how a novel similarity function can be integrated into one of our algorithms as a method of performing clustering and show that the resulting method is superior to existing methods in that it canbe shown to reliably find a globally optimal clustering rather than local optima which other methods often find. We also extend the method to perform topology preserving mappings and show the results of such mappings on artificial and real data.

Key–Words: Clustering, Similarity function, Visualisation, K-means, Topographic mapping.

1 Introduction

One of the major tasks falling to a data analyst is to find groups of data all of which share some underlying feature which is not shared by other samples in the dataset. This is known as clustering the data if it is done in an unsupervised manner i.e. if we do not give the method any class information on which to work. The K-means algorithm [16, 20, 22] is one of the most frequently used investigatory algorithms in data analysis. The algorithm attempts to locate K prototypes or means throughout a data set in such a way that the K prototypes in some way best represents the data. It is an iterative algorithm in which K means are spread throughout the data and the data samples are allocated to the mean which is closest (often in Euclidean norm) to the sample. Then the K means are repositioned as the average of data points allocated to each mean. This continues until stable convergence is reached. The K-means algorithm is one of the first which a data analyst will use to investigate a new data set because it is algorithmically simple, relatively robust and gives 'good enough' answers over a wide variety of data sets: it will often not be the single best algorithm on any individual data set but it may be close to the optimal over a wide range of data sets. However the algorithm is known to suffer from the defect that the means or prototypes found depend on the initial values given to them at the start of the simulation: a typical program will converge to a local optimum. There are a number of heuristics in the literature which attempt to address this issue but, at heart, the fault lies in the performance function on which K-means is based.

[19] proposed a global K-means algorithm, an incremental approach to clustering that adds one cluster prototype at a time through a deterministic global search consisting of N (the data size) executions of the K-means; this algorithm can obtain equivalent or better results than the standard K-means, but it suffers from high computation cost and at the same time gives no guarantee to find the optimum.

Arthur and Vassilvitskii [4] improved the K-means algorithm by substituting the random allocation of the prototypes with a seeding technique. They give experimental results that show the advantage of this algorithm in time and accuracy.

A variation on K-means is the so-called soft K-means [21] in which prototypes are allocated according to

$$\mathbf{m}_k = \frac{\sum_n r_{kn}\mathbf{x}_n}{\sum_{j,n} r_{jn}} \quad (1)$$

$$\text{where e.g. } r_{kn} = \frac{\exp(-\beta d(\mathbf{x}_n, \mathbf{m}_k))}{\sum_j \exp(-\beta d(\mathbf{x}_n, \mathbf{m}_j))} \quad (2)$$

and $d(a, b)$ is the Euclidean distance between a and b. β is a "stiffness" parameter. Note that the standard K-means algorithm is a special case of the soft K-means algorithm in which the responsibilities, $r_{kn} = 1$ when $\mathbf{m}_k$ is the closest prototype to $\mathbf{x}_n$ and 0 otherwise. However the soft K-means does increase the non-localness of the interaction since the responsibilities are typically never exactly equal to 0 for any data point-prototype combination.

However there are still problems with soft K-means. We find that with soft K-means it is important to choose a good value for β; if we choose a poor

ISSN: 1790-5109 ISBN: 978-960-6766-41-1

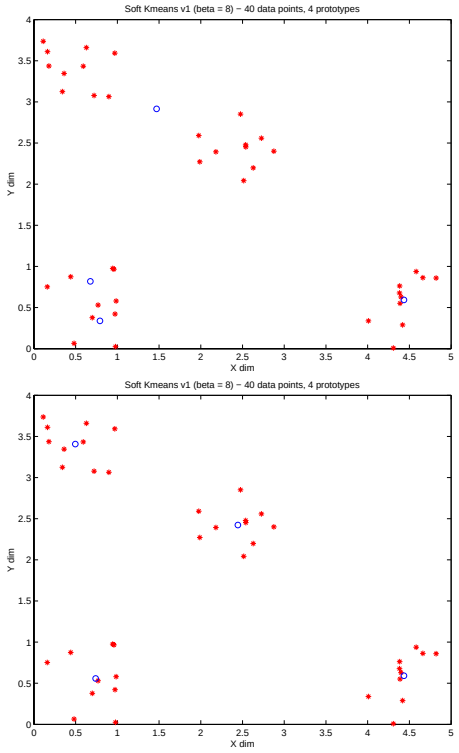

Figure 1: The vagaries of simulations: top, the soft K-means failed to identify the four clusters but on the bottom is successful with the same data set.

value we may have poor results in finding the clusters. However, even with a good value, we often still find that soft K-means has the problem of sensitivity to the prototypes' initialization. As shown in Fig. 1, while soft K-means succeeds in identifying the clusters (bottom diagram), sometimes it failed for the same data sample when we used a different initialization of the prototypes (top diagram).

In [5, 6, 24] we derive a family of new clustering algorithms that solve the problem of sensitivity to initial conditions in the K-means algorithm. In this paper, we show how to extend one of our algorithms, IWC [8, 9], to do clustering based on similarity functions. We take as an example the exponential function to measure the similarity between data points and prototypes; we can use other similarity functions. We compare the new algorithm with the soft K-means algorithm as it is also based on the exponential function. We show in this paper that the new algorithm gives better results than both K-means and soft K-means. Also, this new algorithm solves the problem of excessive computation in [19].

2 Clustering with similarity functions

The inverse weighted clustering algorithm (IWC) has the following rule:

$$\mathbf{m}_k = \frac{\sum_{i=1}^{N} b_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} b_{ik}} \quad (3)$$

where

$$b_{ik} = \frac{\| \mathbf{x}_i - \mathbf{m}_{k*} \|^{P+2}}{\| \mathbf{x}_i - \mathbf{m}_k \|^{P+2}} \quad (4)$$

(3) introduces the clustering algorithm (IWC). We have given extensive simulations in [8] showing that this algorithm is insensitive to the prototypes' initialization. The question now, is how can we use this algorithm for clustering if we have similarity functions that measure the similarity between objects (or data points)? How can we work with the similarity function and at the same time get the benefit of (3)?

2.1 Exponential function as similarity function

Let the exponential function be used, as an example, to measure the similarities between points as $\text{similarity}(\mathbf{x}, \mathbf{m}) = \exp(- \| \mathbf{x} - \mathbf{m} \|)$.

To use this similarity function for clustering while taking the benefits of (3) we need to go through the following steps:

1. Measure the similarities between all data points and prototypes.

2. For alternative similarity function, if the similarity measurements are outwith [0,1], normalize, so that 1 corresponds to the highest similarity and 0 corresponds to no similarity.

3. Map the similarity measurements to distances: $\text{Distance}(\mathbf{x}, \mathbf{m}) = 1 - \exp(- \| \mathbf{x} - \mathbf{m} \|)$.

4. Compute (3) for all prototypes using the new distance measurements generated from the similarity function.

$$b_{ik} = \frac{distance(\mathbf{x}_i - \mathbf{m}_{k*})^{P+2}}{distance(\mathbf{x}_i - \mathbf{m}_k)^{P+2}}. \quad (5)$$

We will call this Inverse Weighted Clustering with Similarity Function algorithm (IWCwSF).

ISSN: 1790-5109 ISBN: 978-960-6766-41-1

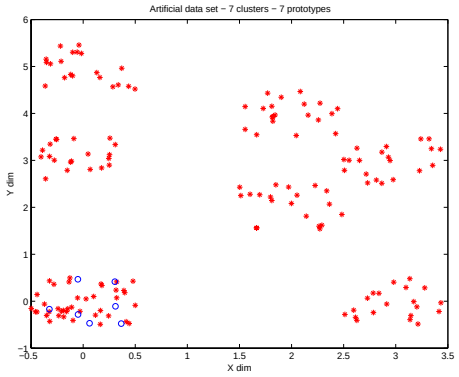

Figure 2: Artificial data set: 7 clusters of red '*'s, and 7 prototypes of blue 'o's.

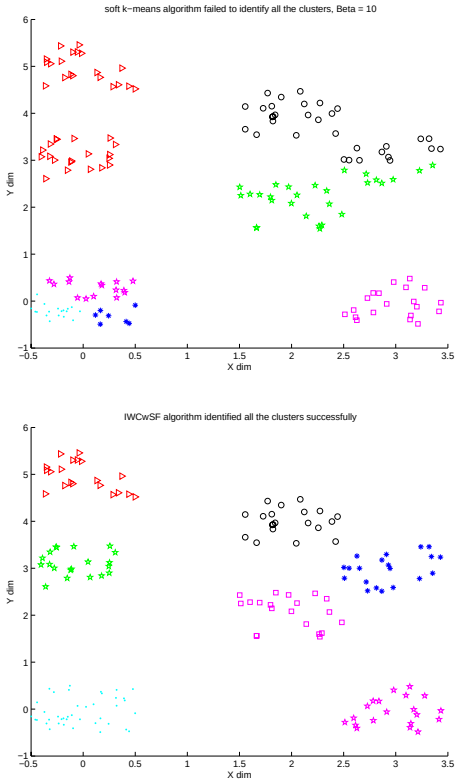

Figure 3: Top: result after applying soft K-means algorithm. Bottom: result after applying inverse weighted clustering with similarity function (IWCwSF).

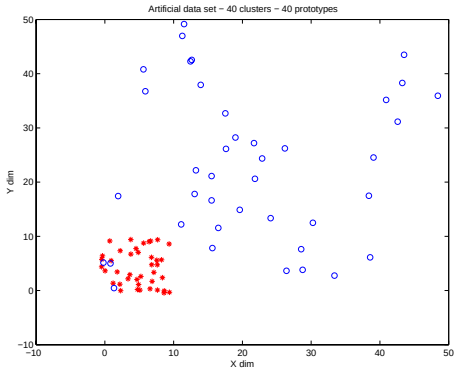

Figure 4: Artificial data set: 40 clusters of red '*'s, and 40 prototypes of blue 'o's.

2.2 Simulations

Fig. 2 shows an artificial data set consisting of 7 clusters of red '*'s, and all the prototypes are initialized to lie within one cluster and are shown as blue 'o's. Fig. 3 shows the results after applying soft K-means and inverse weighted clustering with similarity function (IWCwSF) to this artificial data set shown in Fig. 2. In Fig. 3, top, we see the first cluster, bottom left, is divided into three sub clusters. Also, the upper two clusters, left, are grouped together as one cluster. This poor result from the soft K-means appeared due to its convergence to a local optimum. As shown in Figure 3, while the soft K-means failed to identify the clusters, top, the IWCwSF algorithm identified all of them successfully, bottom.

To show how the new algorithm behaves with dead prototypes, we have in Fig. 4 another artificial data set consisting of 40 data points. Each data point represents a cluster, so we have 40 clusters, and 40 prototypes are initialized randomly and very far from data to represent some dead prototypes. Fig. 5 shows the result after applying soft K-means and IWCwSF to the artificial data set shown in Fig. 4. Again, the new algorithm IWCwSF identified all the clusters successfully, bottom, while soft K-means algorithm didn't, top.

3 A Topology Preserving Mapping

In this section we show how it is possible to extend the IWCSF algorithm to provide a new algorithm for visualization and topology-preserving mappings.

ISSN: 1790-5109 ISBN: 978-960-6766-41-1

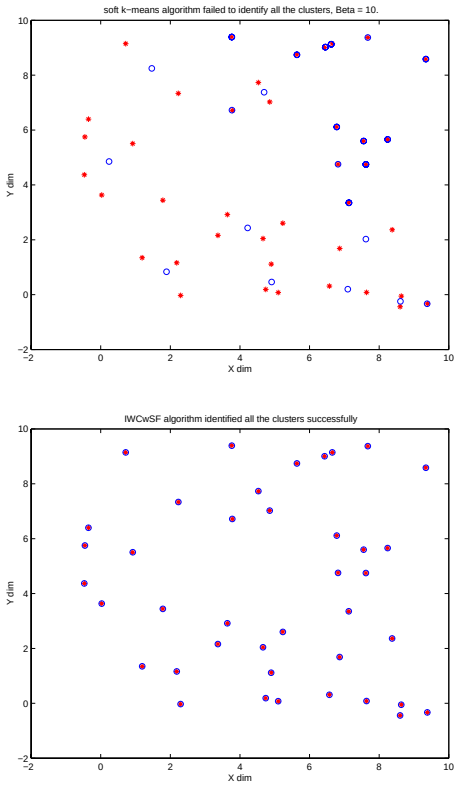

Figure 5: Top: result after applying soft K-means algorithm to the data of Fig. 4. Bottom: result after applying inverse weighted clustering with similarity function (IWCwSF).

3.1 Inverse Weighted Clustering with Similarity Function Topology Preserving Mapping(IWCSFToM)

A topographic mapping (or topology preserving mapping) is a transformation which captures some structure in the data so that points which are mapped close to one another share some common feature while points which are mapped far from one another do not share this feature. The Self-organizing Map (SOM) was introduced as a data quantisation method but has found at least as much use as a visualisation tool.

Topology-preserving mappings such as the Self-organizing Map (SOM) [17, 18] and the Generative Topographic Mapping(GTM) [10, 11, 12, 26, 27] have been very popular for data visualization: we project the data onto the map which is usually two dimensional and look for structure in the projected map by eye. We have recently investigated a family of topology preserving mappings [15] which are based on the same underlying structure as the GTM.

The basis of our model is K latent points, $t_1, t_2, \cdots, t_K$, which are going to generate the K prototypes, $\mathbf{m}_k$. To allow local and non-linear modeling, we map those latent points through a set of M basis functions, $f_1(), f_2(), \cdots, f_M()$. This gives us a matrix Φ where $\phi_{kj} = f_j(t_k)$. Thus each row of Φ is the response of the basis functions to one latent point, or alternatively we may state that each column of Φ is the response of one of the basis functions to the set of latent points. One of the functions, $f_j()$, acts as a bias term and is set to one for every input. Typically the others are gaussians centered in the latent space. The output of these functions are then mapped by a set of weights, W, into data space. W is $M \times D$, where D is the dimensionality of the data space, and is the sole parameter which we change during training. We will use $\mathbf{w}_i$ to represent the i^{th} column of W and Φ_j to represent the row vector of the mapping of the j^{th} latent point. Thus each basis point is mapped to a point in data space, $\mathbf{m}_j = (\Phi_j W)^T$.

We may update W either in batch mode or with online learning: with the Topographic Product of Experts [15], we used a weighted mean squared error; with the Harmonic Topographic Mapping [15, 23, 25], we used Harmonic K-means [28, 29]. We now apply the IWCwSF algorithm to the same underlying structure to create a new topology preserving algorithm.

Each data point is visualized as residing at the prototype on the map which would win the competition for that data point. However we can do rather better by defining the responsibility that the j^{th} prototype has for the i^{th} data point as

$$r_{ji} = \frac{\exp(-\gamma \parallel \mathbf{x}_i - \mathbf{w}_j \parallel^2)}{\sum_k \exp(-\gamma \parallel \mathbf{x}_i - \mathbf{w}_k \parallel^2)} \qquad (6)$$

ISSN: 1790-5109 ISBN: 978-960-6766-41-1

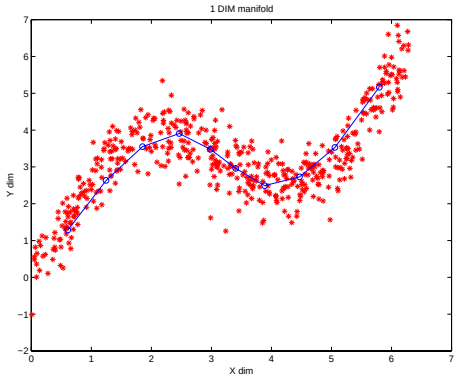

Figure 6: The resulting prototypes' positions after applying IWCSFToM. The prototypes are shown as blue 'o's.

We then project points taking into account these responsiblities: let y_{ij} be the projection of the i^{th} data point onto the j^{th} dimension of the latent space; then

$$y_{ij} = \sum_k t_{kj} r_{ki} \tag{7}$$

where t_{kj} is the j^{th} coordinate of the k^{th} latent point. When we use these algorithms for visualisation purposes, it is these y-values (which are typically two dimensional coordinates) which we use. Note that this method represents each data point $\mathbf{x}_i$ by a value $\mathbf{y}_i$ where $\mathbf{y}_i$ is a weighted sum of the coordinates of the original latent points. An alternative (which is typically used the SOM) is to find the latent point with greatest responsibility for the data point and allocate its $\mathbf{y}_i$ value at this latent point.

3.2 Simulations

- Artificial data set
  We create a simulation with 10 latent points deemed to be equally spaced in a one dimensional latent space, passed through 5 Gaussian basis functions and then mapped to the data space by the linear mapping W which is the only parameter we adjust. We generated 500 two dimensional data points, (x_1, x_2), from the function $x_2 = x_1 + 1.25\sin(x_1) + \mu$ where μ is noise from a uniform distribution in [0,1]. Final result from the IWCSFToM is shown in Fig. 6 in which the projections of consecutive latent points are joined. We see that nearby latent points take responsibility for nearby data points.

- Real data sets

  1. The iris data set is a data set with 150 random samples of flowers from the iris species setosa, versicolor, and virginica collected by Anderson in 1935 [3]. There are 50 observations from each species for sepal length, sepal width, petal length and petal width in cm. This data set was used by Fisher(1936) [13] in his initiation of linear discriminant-function technique. The iris data set is available at [1].
  2. In the algae data set we have 72 samples classified into 9 types. Each sample is recorded as an 18 dimensional vector representing the magnitudes of various pigments. This data set is available at [1].
  3. The Bank data set appeared in [14]. This data set has 200 observations. It contains measurements on 100 forged and 100 genuine bank notes. Each data record contains the six measurments:
     LENGTH: length of bill.
     LEFT: width of bill, measured on the left.
     RIGHT: width of bill, measured on the right.
     BOTTOM: width of the margin at the bottom.
     TOP: width of the margin at the top.
     DIAGONAL: length of the image diagonal.
     This data set is available at [2].

Fig. 7 shows the results of applying IWCSFToM to the iris, algae and bank data set. The real data sets are projected onto a two dimensional grid of latent points (10 x 10). For the iris data set, top , we can see one cluster is distant and separated, while there is difficulty to separate the other two completely. For the algae data set, middle, we have 7 clusters separated, and the two clusters to the left are grouped together. For the bank data set, bottom, the two clusters are identified successfully.

4 Conclusion

We have previously investigated several new families of clustering algorithms [6, 7]. In this paper, we have shown how to get the benefits from the new algorithms but now based on alternative similarity functions. We have used artificial data with deliberately poor initialization since with real data, we often do not know what is a good and what is a poor initialization. This is exacerbated by the 'curse of dimensionality': even when we decide to use means of data as cluster prototypes, these can often lie far from the actual data which may be found in the outer shell of a high dimensional sphere.

Finally we have extended the proposed algorithm for visualisation by incorporating a latent or hidden space which underlies the prototypes' positions in data space. By constraining the latent points' positions to certain values, we can ensure that we have pre-

ISSN: 1790-5109 ISBN: 978-960-6766-41-1

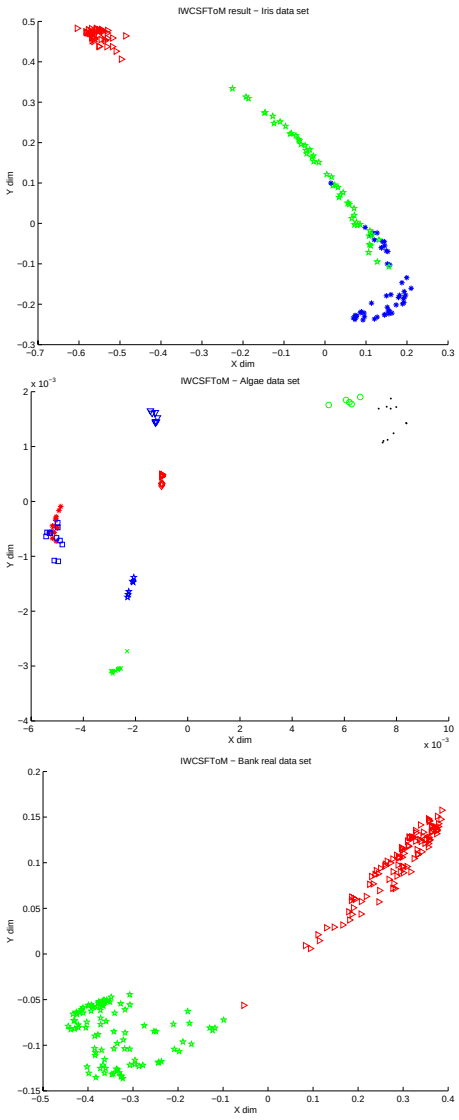

Figure 7: The results of applying IWCSFToM on the real data sets, iris, top, algae, middle and bank, bottom.

served local neighbourhood relations with the prototypes' positions in data space. We have illustrated the effect of this with particular emphasis on the power of our method as a visualization tool. We have shown typical results with the projections of real data into an underlying two dimensional latent space.

We will, in future, investigate alternative similarity functions to evaluate which is best for which data and under what conditions. There may be no globally optimal similarity function but there may be underlying principles which we can use to determine the optimal similarity function for a particular type of data.

References:

[1] `http://mlearn.ics.uci.edu/databases/`.

[2] `http://www.quantlet.com/mdstat/scripts/mva/htmlbook/mvahtmlnode129.html#tabbank`.

[3] E. Anderson. The irises of the gaspe peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

[4] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. *In Bay Area Theory Symposium, BATS 06*, 2006. `http://www.stanford.edu/~sergeiv/papers/kMeansPP-soda.pdf`.

[5] W. Barbakh. The family of inverse exponential k-means algorithms. *Computing and Information Systems*, 11(1):1–10, February 2007. ISSN 1352-9404.

[6] W. Barbakh, M. Crowe, and C. Fyfe. A family of novel clustering algorithms. In *7th international conference on intelligent data engineering and automated learning, IDEAL2006*, pages 283–290, September 2006. ISSN 0302-9743 ISBN-13 978-3-540-45485-4.

[7] W. Barbakh and C. Fyfe. Clustering with reinforcement learning. In *International Conference on Intelligent Data Engineering and Automated Learning IDEAL'07*, pages 507–516, December 2007. LNCS 4881.

[8] W. Barbakh and C. Fyfe. Inverse weighted clustering algorithm. *Computing and Information Systems*, 11(2):10–18, May 2007. ISSN 1352-9404.

[9] W. Barbakh and C. Fyfe. Tailoring local and global interactions in clustering algorithms. Technical Report 40, School of Computing, University of Paisley, March 2007. ISSN 1461-6122.

[10] C. M. Bishop, M. Svensen, and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1997.

[11] C. M. Bishop, M. Svensen, and C. K. I. Williams. Magnification factors for the gtm algorithm. In *Proceeding of the IEE 5th International Conference on Artificial Neural Networks, Cambridge, UK*, pages 64–69, 1997.

ISSN: 1790-5109 ISBN: 978-960-6766-41-1

[12] C. M. Bishop, M. Svensen, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21(1):203–224, 1998.

[13] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, pages 179–188, 1936.

[14] B. Flury and H. Riedwyl. *Multivariate Statistics: A practical approach*. Cambridge University Press, 1988.

[15] C. Fyfe. Two topographic maps for data visualization. *Data Mining and Knowledge Discovery*, 14:207–224, 2007. ISSN 1384-5810.

[16] J. Hartigan and M. Wang. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[17] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 1984.

[18] T. Kohonen. *Self-Organising Maps*. Springer, 1995.

[19] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition 36*, pages 451–461, 2003.

[20] S. P. Lloyd. *Least squares quantization in pcm*. Technical note, Bell Laboratories, 1957. Published in 1982 in IEEE Transactions on Information Theory 28, 128-137.

[21] D. J. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press., 2003.

[22] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297, 1967.

[23] S. McGlinchey and M. Pena. Quantization errors in the harmonic topographic mapping. In *The 9th WSEAS International Conference on applied mathematics, MATH 06*, pages 105–110, May 2006.

[24] M. Pena, W. Barbakh, and C. Fyfe. *Principal Manifolds for Data Visualization and Dimension Reduction*, chapter Topology-Preserving Mappings for Data Visualisation, pages 132–152. Springer, 2007. ISBN 978-3-540-73749-0.

[25] M. Pena and C. Fyfe. The harmonic topographic map. Technical Report 35, School of Computing, University of Paisley, 2005.

[26] M. Svensen. *GTM: The Generative Topographic Mapping*. PhD thesis, Aston University, Birmingham, UK, 1998.

[27] P. Tino and I. Nabney. Hierarchical gtm: constructing localized non-linear projection manifolds in a principled way. *(IEEE) Transaction on Pattern Analysis and Machine Intelligence*, 24(5):639–656, 2001.

[28] B. Zhang. Generalized k-harmonic means – boosting in unsupervised learning. Technical Report HPL-2000-137, HP Laboratories, Palo Alto, October 2000.

[29] B. Zhang, M. Hsu, and U. Dayal. K-harmonic means - a data clustering algorithm. Technical Report HPL-1999-124, HP Laboratories, Palo Alto, October 1999.