Unsupervised and Semi-supervised Clustering: a Brief Survey *

Nizar Grira, Michel Crucianu, Nozha Boujemaa
INRIA Rocquencourt, B.P. 105
78153 Le Chesnay Cedex, France
{Nizar.Grira, Michel.Crucianu, Nozha.Boujemaa}@inria.fr

August 15, 2005

1 Unsupervised Clustering

Clustering (or *cluster analysis*) aims to organize a collection of data items into clusters, such that items within a cluster are more "similar" to each other than they are to items in the other clusters. This notion of similarity can be expressed in very different ways, according to the purpose of the study, to domain-specific assumptions and to prior knowledge of the problem.

Clustering is usually performed when no information is available concerning the membership of data items to predefined classes. For this reason, clustering is traditionally seen as part of unsupervised learning. We nevertheless speak here of *unsupervised* clustering to distinguish it from a more recent and less common approach that makes use of a small amount of supervision to "guide" or "adjust" clustering (see section 2).

To support the extensive use of clustering in computer vision, pattern recognition, information retrieval, data mining, etc., very many different methods were developed in several communities. Detailed surveys of this domain can be found in [25], [27] or [26]. In the following, we attempt to briefly review a few core concepts of cluster analysis and describe categories of clustering methods that are best represented in the literature. We also take this opportunity to provide some pointers to more recent work on clustering.

*in *A Review of Machine Learning Techniques for Processing Multimedia Content*, Report of the MUSCLE European Network of Excellence (6th Framework Programme)

1.1 A Typology of Methods

We start by mentioning some criteria that provide significant distinctions between clustering methods and can help selecting appropriate candidate methods for one's problem:

- *Objective of clustering.* Many methods aim at finding a single *partition* of the collection of items into clusters. However, obtaining a *hierarchy* of clusters can provide more flexibility and other methods rather focus on this. A partition of the data can be obtained from a hierarchy by cutting the tree of clusters at some level.

- *Nature of the data items.* Most clustering methods were developed for numerical data, but some can deal with categorical data or with both numerical and categorical data.

- *Nature of the available information.* Many methods rely on rich representations of the data (e.g. vectorial) that let one define prototypes, data distributions, multidimensional intervals, etc., beside computing (dis)similarities. Other methods only require the evaluation of pairwise (dis)similarities between data items; while imposing less restrictions on the data, these methods usually have a higher computational complexity.

- *Nature of the clusters.* The degree of membership of a data item to a cluster is either in $[0, 1]$ if the clusters are *fuzzy* or in $\{0, 1\}$ if the clusters are *crisp*. For fuzzy clusters, data items can belong to some degree to several clusters that don't have hierarchical relations with each other. This distinction between fuzzy and crisp can concern both the clustering mechanisms and their results. Crisp clusters can always be obtained from fuzzy clusters.

- *Clustering criterion.* Clusters can be seen either as distant *compact* sets or as *dense* sets separated by low density regions. Unlike density, compactness usually has strong implications on the shape of the clusters, so methods that focus on compactness should be distinguished from methods that focus on the density.

Several taxonomies of clustering methods were suggested in [17], [27] or [26]. But given the high number and the strong diversity of the existing clustering methods, it is probably impossible to obtain a categorization that is both meaningful and complete. By focusing on some of the discriminating criteria just mentioned we put forward the simplified taxonomy shown below, inspired by the one suggested in [26].

- *Partitional clustering* aims to directly obtain a single *partition* of the collection of items into clusters. Many of these methods are based on the iterative optimization of a criterion function reflecting the "agreement" between the data and the partition. Here are some important categories of partitional clustering methods:

  - *Methods using the squared error* rely on the possibility to represent each cluster by a prototype and attempt to minimize a cost function that is the sum over all the data items of the squared distance between the item and the prototype of the cluster it is assigned to. In general, the prototypes are the cluster centroids, as in the popular k-means algorithm [31]. Several solutions were put forward for cases where a centroid cannot be defined, such as the k-medoid method [27], where the prototype of a cluster is an item that is "central" to the cluster, or the k-modes method [24] that is an extension to categorical data.

    By employing the squared error criterion with a Minkowski metric or a Mahalanobis metric, one makes the implicit assumption that clusters have elliptic shape. The use of multiple prototypes for each cluster or of more sophisticated distance measures (with respect to one or several cluster models, see e.g. [12]) can remove this restriction.

    Fuzzy versions of methods based on the squared error were defined, beginning with the Fuzzy C-Means [7]. When compared to their crisp counterparts, fuzzy methods are more successful in avoiding local minima of the cost function and can model situations where clusters actually overlap. To make the results of clustering less sensitive to outliers (isolated data items) several fuzzy solutions were put forward, based on robust statistics [20] or on the use of a "noise cluster" [13], [30].

    Many early methods assumed that the number of clusters was known prior to clustering; since this is rarely the case, techniques for finding an "appropriate" number of clusters had to be devised. This is an important issue for partitional clustering in general. For methods based on the squared error, the problem is partly solved by adding a *regularization* term to the cost function. This is the case, for example, for the competitive agglomeration method introduced in [19], where clusters compete for membership of data items and the number of clusters is progressively reduced until an optimum is reached. With such solutions, instead of the number

of clusters one has to control a regularization parameter, which is often more convenient. Another solution is to use a cluster validity index (see section 1.2) to select *a posteriori* the appropriate number of clusters.

– *Density-based methods* consider that clusters are dense sets of data items separated by less dense regions; clusters may have arbitrary shape and data items can be arbitrarily distributed. Many methods, such as DBSCAN [16] (further improved in [34]), rely on the study of the density of items in the neighbourhood of each item. Some interesting recent work on density-based clustering is using 1-class support vector machines [6].

  One can consider within the category of density-based methods the *grid-based* solutions, such as DenClue [23] or CLIQUE [1], mostly developed for spatial data mining. These methods quantize the space of the data items into a finite number of cells and only retain for further processing the cells having a high density of items; isolated data items are thus ignored. Quantization steps and density thresholds are common parameters for these methods.

  Many of the *graph-theoretic* clustering methods are also related to density-based clustering. The data items are represented as nodes in a graph and the dissimilarity between two items is the "length" of the edge between the corresponding nodes. In several methods, a cluster is a subgraph that remains connected after the removal of the longest edges of the graph [25]; for example, in [40] the minimal spanning tree of the original graph is built and then the longest edges are deleted. However, some other graph-theoretic methods rely on the extraction of *cliques* and are then more related to squared error methods. Based on graph-theoretic clustering, there has been significant interest recently in *spectral* clustering using kernel methods [33].

– *Mixture-resolving* methods assume that the data items in a cluster are drawn from one of several distributions (usually Gaussian) and attempt to estimate the parameters of all these distributions. The introduction of the expectation maximization (EM) algorithm in [15] was an important step in solving the parameter estimation problem. Mixture-resolving methods make rather strong assumptions regarding the distribution of the data. The choice of the number of clusters for these methods is thoroughly studied in more recent work such as [3] or [10]. In some cases a model for the noise is explicitely considered.

Most mixture-resolving methods view each cluster as a single simple distribution and thus strongly constrain the shape of the clusters; this explains why we did not include these methods in the category of density-based clustering.

- *Hierarchical clustering* aims to obtain a hierarchy of clusters, called *dendrogram*, that shows how the clusters are related to each other. These methods proceed either by iteratively merging small clusters into larger ones (agglomerative algorithms, by far the most common) or by splitting large clusters (divisive algorithms). A partition of the data items can be obtained by cutting the dendrogram at a desired level.

  Agglomerative algorithms need criteria for merging small clusters into larger ones. Most of the criteria concern the merging of pairs of clusters (thus producing binary trees) and are variants of the classical single-link [35], complete-link [28] or minimum-variance [37], [32] criteria. The use of the single-link criterion can be related to density-based methods but often produces upsetting effects: clusters that are "linked" by a "line" of items cannot be separated or most items are individually merged to one (or a few) cluster(s). The use of the complete-link or of the minimum-variance criterion relates more to squared error methods.

  Many recent hierarchical methods focus on the use of density-like information and don't constrain the shape of the clusters. They often reflect interest in the database community for dealing with huge datasets and for speeding-up access. CURE [22] employs multiple representatives per cluster in order to obtain clusters of arbitrary shape while avoiding the problems of the single-link criterion. OPTICS [2] does not build an explicit clustering of the collection of items, but rather an ordered representation of the data that reflects its clustering structure.

1.2 Cluster Validity Analysis

An unsupervised learning procedure is usually more difficult to assess than a supervised one. Several questions can be asked regarding the application of clustering methods:

- Are there clusters in the data?
- Are the identified clusters in agreement with the prior knowledge of the problem?
- Do the identified clusters fit the data well?

- Are the results obtained by a method better than those obtained by another?

The first question concerns the *cluster tendency* of the data and should in principle be answered before attempting to perform clustering, using specific statistical tests. Unfortunately, such tests are not always very helpful and require the formulation of specific test hypotheses.

The other questions concern the analysis of cluster validity and can only be answered after application of clustering methods to the data. According to [26], one can distinguish between three types of validation procedures:

- *External* validation consists in finding an answer to the second question above and can only be performed when prior knowledge of the problem is available. The prior knowledge may concern general characteristics of the clusters (e.g. expected compactness) or relations between specific items (e.g. items A and B should belong to a same cluster and item C to a different one). Sometimes this knowledge is confirmatory but not prescriptive.

- *Internal* validation concerns the third question above and is based on an evaluation of the "agreement" between the data and the partition. In the following we give a few examples of validity indices that were put forward in the literature for some of the above categories of clustering methods. Note that the definitions of these internal validity indices usually make their direct optimization intractable.

  In [8] several indices for crisp clustering are evaluated: the modified Hubert's statistic, related to the alignment between the dissimilarity matrix and the crisp partition matrix, the Davies-Bouldin index, roughly defined as the ratio between within-cluster scatter and between-cluster separation, and Dunn's index with several alternative definitions (some of which are introduced in [8]) for the diameter of a set and for the distance between sets.

  For fuzzy partitional methods, internal validity indices should take into account both the data items and the membership degrees resulting from clustering. The *average partition density* in [21] is obtained as the mean ratio between the "sum of central members" (sum of membership degrees for the items close to the prototype) of each cluster and the volume of the cluster. The Xie-Beni index put forward in [38] is the ratio between the average intra-cluster variation and the minimum distance between cluster centers, and is thus related to the ratio between intra-cluster and inter-cluster variance.

Among the validity indices suggested for density-based clustering methods, we mention the two in [18]: the first one measures the variation of cluster labels in the neighbourhood of data items, the other evaluates the density on the path between data items.

- *Relative* comparisons attempt to provide an answer to the fourth question above and are usually the main application of the indices defined for the internal validation. Such comparisons are often employed for selecting good values for important parameters, such as the number of clusters.

2 Semi-supervised Clustering

In addition to the similarity information used by unsupervised clustering, in many cases a small amount of knowledge is available concerning either pairwise (must-link or cannot-link) constraints between data items or class labels for some items. Instead of simply using this knowledge for the external validation of the results of clustering, one can imagine letting it "guide" or "adjust" the clustering process, i.e. provide a limited form of supervision. The resulting approach is called *semi-supervised clustering*. We also consider that the available knowledge is too far from being representative of a target classification of the items, so that supervised learning is not possible, even in a transductive form.

Note that class labels can always be translated into pairwise constraints for the labeled data items and, reciprocally, by using consistent pairwise constraints for some items one can obtain groups of items that should belong to a same cluster.

2.1 A Typology of Methods

Two sources of information are usually available to a semi-supervised clustering method: the similarity measure unsupervised clustering would employ and some pairwise constraints (must-link or cannot-link). For semi-supervised clustering to be profitable, these two sources of information should not completely contradict each other.

Unlike traditional clustering, the semi-supervised approach to clustering has a short history and few methods were published until now. The main distinction between these methods concerns the way the two sources of information are combined (see the taxonomy in [5]): either by adapting the similarity measure or by modifying the search for appropriate clusters.

- In *similarity-adapting* methods, an existing clustering algorithm using some similarity measure is employed, but the similarity measure is adapted so that the available constraints can be easier satisfied. Several similarity measures were employed for similarity-adapting semi-supervised clustering: the Jensen-Shannon divergence trained with gradient descent [11], the Euclidean distance modified by a shortest-path algorithm [29] or Mahalanobis distances adjusted by convex optimization [39], [9]. Among the clustering algorithms using such adapted similarity measures we can mention hierarchical single-link [9] or complete-link [29] clustering and k-means [39], [9].

- In *search-based* methods, the clustering algorithm itself is modified so that user-provided constraints or labels can be used to bias the search for an appropriate clustering. This can be done in several ways, such as by performing a transitive closure of the constraints and using them to initialize clusters [4], by including in the cost function a penalty for lack of compliance with the specified constraints [14], or by requiring constraints to be satisfied during cluster assignment in the clustering process [36].

References

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM-SIGMOD International Conference on the Management of Data*, pages 94–105, June 1998.

[2] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM Press, 1999.

[3] Jeffrey D. Banfield and Adrian E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49:803–821, 1993.

[4] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Semi-supervised clustering by seeding. In *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*, pages 19–26, 2002.

[5] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. Comparing and unifying search-based and similarity-based approaches to semi-

supervised clustering. In *Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 42–49, Washington, DC, August 2004.

[6] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2002.

[7] James C. Bezdek, R. Ehrlich, and W. Full. FCM: Fuzzy c-means algorithm. *Computers and Geoscience*, 1984.

[8] James C. Bezdek and Nikhil R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):301–315, 1998.

[9] Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *International Conference on Knowledge Discovery and Data Mining*, pages 39–48, Washington, DC, 2003.

[10] Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

[11] David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised clustering with user feedback, 2000.

[12] Rajesh N. Davé. Use of the adaptive fuzzy clustering algorithm to detect lines in digital images. In *Intelligent Robots and Computer Vision VIII*, volume 1, pages 600–611, 1989.

[13] Rajesh N. Davé and Raghu Krishnapuram. Robust clustering methods: A unified view. *IEEE Transactions on Fuzzy Systems*, 5(2):270–293, 1997.

[14] A. Demiriz, K. Bennett, and M. Embrechts. Semi-supervised clustering using genetic algorithms. In C. H. Dagli et al., editor, *Intelligent Engineering Systems Through Artificial Neural Networks 9*, pages 809–814. ASME Press, 1999.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[17] Brian Everitt. *Cluster analysis*. Halsted Press, New York, 1974, 1993.

[18] Greet Frederix and Eric Pauwels. Two general geometric cluster validity indices. In *Proceedings of the 4th Industrial Conference on Data Mining (ICDM'2004)*, July 2004.

[19] Hichem Frigui and Raghu Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7):1109–1119, 1997.

[20] Hichem Frigui and Raghu Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, 1999.

[21] I. Gath and A. B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on pattern Analysis and Machine Intelligence*, 11(7):773–781, 1989.

[22] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 73–84. ACM Press, 1998.

[23] A. Hinneburg and D. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, 1998.

[24] Zhexue Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (SIGMOD-DMKD'97)*, May 1997.

[25] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[26] Anil K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[27] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 1990.

[28] B. King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86–101, 1967.

[29] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the 19th International Conference on Machine Learning*, pages 307–314. Morgan Kaufmann Publishers Inc., 2002.

[30] Bertrand Le Saux and Nozha Boujemaa. Unsupervised robust clustering for image database categorization. In *Proceedings of the IEEE-IAPR International Conference on Pattern Recognition (ICPR'2002)*, August 2002.

[31] J. McQueen. Some methods for classifcation and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathmatical Statistics and Probabilty*, pages 281–297, 1967.

[32] Fionn Murtagh. Structure of hierarchic clusterings: implications for information retrieval and for multivariate data analysis. *Inf. Process. Manage.*, 20(5-6):611–617, 1984.

[33] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of 14th Advances in Neural Information Processing Systems*, 2002.

[34] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Min. Knowl. Discov.*, 2(2):169–194, 1998.

[35] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy*. Freeman, 1973.

[36] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1103–1110, 2000.

[37] J. H. Ward. Hierachical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[38] Xuanli Lisa Xie and Gerardo Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991.

[39] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512, Cambridge, MA, 2003. MIT Press.

[40] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computing*, C-20:68–86, 1971.