Multi density DBSCAN

Conference Paper · September 2011

DOI: 10.1007/978-3-642-23878-9_53 · Source: DBLP

Wesam Ashour
Islamic University of Gaza

MULTI-DENSITY DBSCAN USING REPRESENTATIVES: MDBSCAN-UR

Rwand Ahmed **Eman El-Zaza** **Wesam Ashour**

Master of Computer Engineering Dept, Islamic university, Gaza
Gaza, Palestine

Islamic University-Gaza

Email: {rwando, enoizi, washour } @iugaza.edu

Abstract

DBSCAN is one of the most popular algorithms for cluster analysis. It can discover clusters with arbitrary shape and separate noises. But this algorithm cannot choose its parameter according to distributing of dataset. It simply uses the global uses minimum number of points (MinPts) parameter, so that the clustering result of multi-density database is inaccurate. In addition, when it is used to cluster large databases, it will cost too much time. For these problems, we propose MDBSCAN-UR algorithm which is based on spatial index and grid technique witch make the clustering using representatives points that capture the shape and extent of the cell that they chosen in and that in order to enhance the time complexity. In this paper, we apply an unsupervised machine learning approach based on DBSCAN algorithm. We use local MinPts for every cell in the grid to overcome the problem of undetermined the clusters in multi-density data set with DBSCAN correctly, as we show that the experimental evaluation of our algorithm MDBSCAN-UR is effective and efficient.

Keywords*:* Clustering, DBSCAN, Multi-density, representative.

1. Introduction:

Clustering, which divides the data to disparate clusters, is a crucial part of data mining. The objects within a cluster are “similar,” whereas the objects of different clusters are “dissimilar” [1]. Clustering is one of the most useful tasks in data mining process. There are many algorithms that deal with the problem of clustering large number of objects. The

different algorithms can be classified regarding different aspects. These methods can be categorized into partitioning methods [19,20,21], hierarchical methods [19,22,23], density based methods [24,25,26], grid based [27,28,29] methods, and model based methods [30,31]. DBSCAN checks the Eps-neighborhood of each point in database. If Eps- neighborhood of a point p contains more than MinPts, a new cluster with p as a core object is created. It then iteratively collects directly density-reachable objects from these core objects, which may involve the merge of a few density-reachable clusters. The process terminates when no new point can be added to any cluster[12]. The conventional DBSCAN and its improved algorithm presented in paper [2, 9-11] can only process the numerical data. They are incapable of processing data with categorical attributes. Usually, the densities of dataset used in cluster analyses are different. However, until now there is no a very effective algorithm to get the accurate density of the dataset with multi-density. DBSCAN [2], density-based clustering not only availably avoids noises but also effectually clusters various datasets. Whereas, for the multi-density dataset, DBSCAN is not a good algorithm for the runtime complexity is higher[8]. In order to lower the time complexity, the academia has presented a grid-based cluster technique[3], which divides the data space into disjunctive grid. The data in the same grid can be treated as a unitary object, and all the operations of clustering are on the grid [3]. This paper introduces MDBSCAN-UR, based on grid multi-density cluster algorithm and Using Representatives. Firstly, MDBSCAN-UR uses the space dividing technique. This technique defines a single grid as a part, and gets the local-MinPts parameter of every part based on grid-density, then chooses some representative points from each grid then using DBSCAN[8]. The experiment results shows the accuracy and efficiency of MDBSCAN-UR . The key idea of density-based clustering is that for each object of a cluster the neighborhood of a given radius Eps has to contain at least a minimum number of objects (MinPts), i.e. the cardinality of the neighborhood has to exceed a threshold. To know formal definitions for this notion of a clustering see [2, 8].

Semi-supervised clustering with constraints

Semi-supervised clustering, which uses class labels or pairwise constraints on some examples to aid unsupervised clustering, has been the focus of several recent projects [13]. Existing methods for semi-supervised clustering fall into two general approaches: constraint-based and distance based methods. At present, many scholars incorporated pairwise constraints into state-of-art clustering algorithms. Kiri Wagstaff et al. [14] incorporated pairwise constraints into k-means algorithm so as to satisfy these constraints in the process of clustering; Sugato Basu et al. [13] proposed the PCK-Means algorithm which modifies the objective function of clustering so that these constraints can be satisfied in some degree, however it must rely on parameters and a large number of constraints; Nizar Grira et al. [15] proposed the PCCA algorithm which was used to image database categorization; Davidson et al. [16] enhanced the hierarchical clustering with pairwise constraints, and presented intractability results for some constraint
combinations [17]; Wei Tang et al. [18] proposed a feature projection method with pairwise constraints ,which can handle the high-dimension sparse data effectively. The following of this paper is organized as follows: chapter 2 covers the idea of DBSCAN and its disadvantage; chapter 3 proposes MDBSCAN-UR and describes the key points of MDBSCAN-UR; chapter 4 analyzes the result of experimentation, chapter 5 gets the conclusions.

2. Preliminaries

2.1. Introduction of DBSCAN

The key idea of density-based clustering is that for each object of a cluster the neighborhood of a given radius (Eps) has to contain at least a minimum number of objects (MinPts), i.e. the cardinality of the neighborhood has to exceed a threshold. DBSCAN checks the Eps-neighborhood of each point in the database. If the number of points in some cell p (NEps(p)), has points more than MinPts, a new cluster C containing the points in NEps(p) is created. Then, the Eps-neighborhood of all points q in C which has not yet been processed is checked. If NEps(q) contains points more than MinPts, the neighborhood of q which is not contained in C are added to the cluster and their Eps-neighborhood is checked in the next step[6].

2.2 The problems of DBSCAN

DBSCAN algorithm has some problems:

1. An important property of many datasets is that their intrinsic cluster structure cannot be characterized by global MinPts. For example, in the dataset depicted in Figure 1, it is impossible to simultaneously detect clusters A, B, C1, C2, and C3 using one global-MinPts. A decomposition based on global MinPts would only consist of cluster A, B, and C, or C1, C2, and C3. In the second case, the objects between A and B are noises.

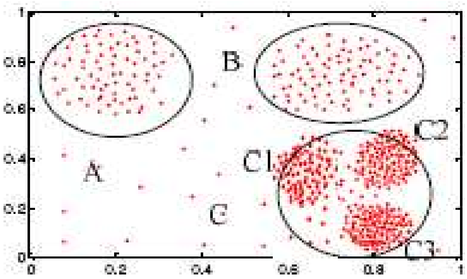

Figure 1. Multi-Density Database

2. User can specify difficulty the values of parameters Eps and MinPts .

3. The runtime complexity and implementation of DBSCAN are not linearly.

3. Multi-Density DBSCAN Cluster MDBSCAN-UR

3.1 Algorithm description

In order to reduce the time complexity, we propose a grid-based cluster technique, [3] which divides the data space into cells. We treat all data in the same cell as an object, and all the operations of clustering are on the grid. The main contribution in this paper is that it deal with two approaches when make clustering in data set with multi-densities, so that the two chooses out the same result with a flexible options. The first option is to deal with a specific cell with its local density so that vary the parameter (Eps) from cell to cell in the grid and make the parameter (MinPts) to be constant, and the second option is to make the parameter (Eps) to be constant over all cells and vary the parameter (MinPts) from cell to cell. Parameters chose are depend on the local density of the cells in the grid. An example is shown in the Table 1, MDBSCAN-UR algorithm adopts a middle ground between the centroid-based and the all-point extremes[5]. A constant number c of well scattered points in each cell in the grid are first chosen. These scattered points capture the shape and extent of the cluster. The chosen scattered points are next shrunk towards the centroid of the cell by a fraction alpha . These scattered points after shrinking are used as representatives of its cell. The cells with the closest pair of representative points are the cells that are merged at each step of the algorithm. The scattered points approach employed by our algorithm alleviates the shortcomings of both the all-points as well as the centroid-based approaches[5].

Table 1 : Example of applying equation 1 and equation 2 in some data sets.

Cells	No of points	Local density	Eps (const r.p = 10)	r.p (const Eps = 5 cm2)
Cell 1	2500	250 pts/cm2	Radius = 25 ;//25* 10 = 250	Representative points = 50 // 50 * 5 = 250
Cell 2	1500	150 pts/cm2	Radius = 15 // 15 * 10 = 150	Representative points = 30 // 30 * 5 = 150
Cell 3	1000	100 pts/cm2	Radius = 10 // 10 * 10 = 100	Representative points = 20 // 20 * 5 = 100
Cell 4	500	50 pts/cm2	Radius = 5 // 5 * 10 = 50	Representative points = 10 // 10 * 5 = 50

3.2 The General process of clustering algorithm

1. Datasets input and Data standardization. 2. Dividing the data space into cells to become a grid. 3. Determine the local representative points in each cell. 4. Local-clustering using DBSCAN algorithm. 5. Noises and Border processing.

3.3. The process strategy of key steps

3.3.1. dividing dataset into cells

Partitioning is dividing the data space into cells. so the number of points in a cell and in the neighborhood are not similar as shown in Figure 2 and Figure 3.

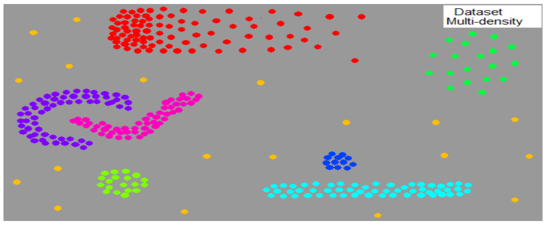

Figure 2. Multi-density Dataset

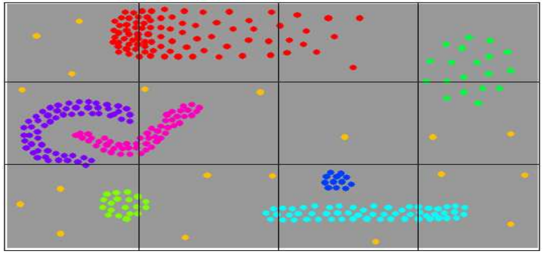

Figure 3. Stage 1 divide the dataset to cells

3.3.2. Chosen representative points

a constant number c of well scattered points in each cell are chosen. The scattered points in some specific cell will capture the shape and extent of that cell as shown in Figure 4.

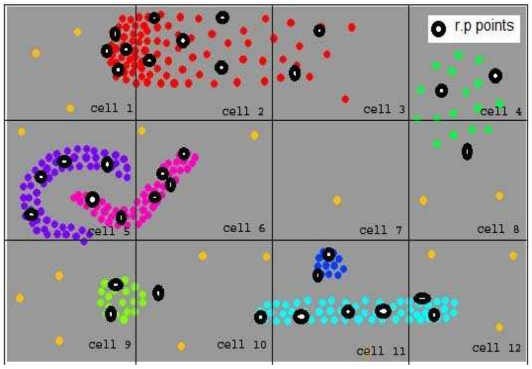

Figure 4. Stage 2 take a well scattered representative points in each cell.

3.3.3. Selecting Parameters of MinPts and Eps.

In each cell one approach is used in selecting the MinPts and Eps. Either select the MinPts for each cell individually and let the Eps to be constant for all cells or select the Eps for each cell individually and let the MinPts to be constant for all cells.
Example apply the two Equations:
Local density(some cell) = r.p * Eps ; r.p const and Eps variable Equation (1)
Local density(some cell) = r.p*Eps ; Eps const and r.p variable Equation (2)
Apply Equation 1: Using same Eps with varying MinPts:
We apply the second MDBSCAN-UR equations on three cells as shown below in Figure 5; using same MinPts in all cells to merge but in different Eps from cell to cell; i.e. the MinPts is 5 at all cells but, at the most left cell the Eps is most smaller because this cell is the most dense; at the middle cell the Eps is wider because this cell is less dense, at the right cell the Eps is the most wider because it is the lowest density.

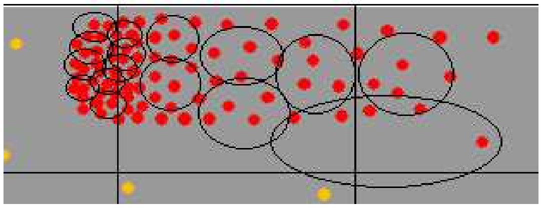

Figure 5: using same MinPts with varying Eps

Apply Equation 2: Using same Eps with varying MinPts:
We apply the first MDBSCAN-UR equations on three cells as shown below in Figure 6; using same Eps in all cells to merge but in different number of MinPts from cell to cell;. i.e. at the most left cell the MinPts is 5; at the middle cell the MinPts is 3, at the right cell MinPts is 2.

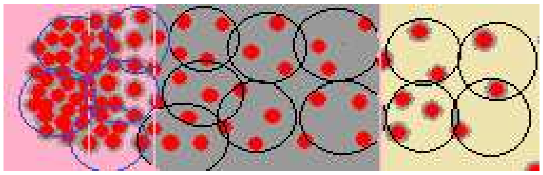

Figure 6: using same Eps with varying MinPts

3.3.4. Clustering.

MDBSCAN-UR mainly gets the idea of locally clustering, identifying a local-MinPts for each cell in the grid. For each cell, processing clustering with their local-MinPts to form a number of distributed local clusters. For the data density within the same cluster should be similar, if two cells have same density and can be merged with the Eps chosen , can be merged to a single cluster.

The stages of MDBSCAN-UR algorithm is given below:

Divide the dataset into cells to form grid. Then scan each cell to compute the local density, after that Choose a representatives points in each cell in the grid. After that we apply DBSCAN method on each cell in the grid at once at a specific parameters as computed in the previous stage. Then make merge method if necessary between cells[5]. Finally Handel outliers and output Clusters , outlier.

3.4.5. Noise Elimination

Noises distribution is not very sparse, but its amount is too small to form a cluster. So, we set a parameter according to the size of dataset, and that size is differ from cell to cell in the grid. When the amount of data in a cluster is less than the threshold size of representatives points ; cores , the entire cluster will be treat as noise. In DBSCAN, if the border object is in the scope of Eps neighborhood of different core objects, it is classified into the cluster to sort firstly. In MDBSCAN-UR algorithm, we set such object to the cluster whose representative core object is most nearest from this object.

4. Experimental Evaluation and Analysis

In this section, we evaluate the performance of MDBSCAN-UR, and compare it with DBSCA using two different types of data set artificial and real as following:

4.1 Artificial data set

Data Set 1: name is clusterxyz.dat with 600 points to clustered and its output using DBSCAN shown in figure 1000 there is 3clusters, but not as needed.

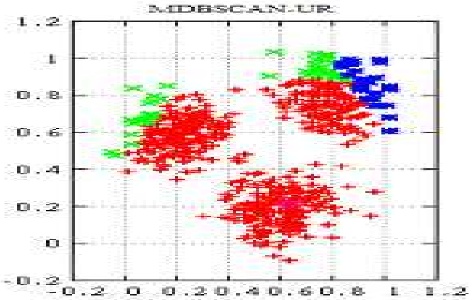

Figure 7: Data set 1 output using DBSCAN

If we apply our algorithm and divide the space to three cells we get three clusters with points as the following:

```
Number of Points to be in Cluster1:
89

Number of Points to be in Cluster2:
154

Number of Points to be in Cluster3:
357
```

Data Set 2: name is S1.txt with 5000 points to clustered and its output using DBSCAN shown in figure 1001 there is 15clusters.

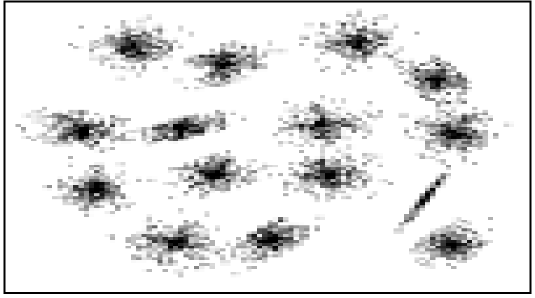

Figure 8: Data set 2 output using DBSCAN

If we apply our algorithm and divide the space to four cells we get 15 clusters with points as the following:

```
Number of Points in cell 1: 1075
with 4 cluster

Number of Points in cell 2: 1292
with 4 cluster

Number of Points in cell 3: 1318
with 3 cluster

Number of Points in cell 4: 1315
with 4 cluster
```

Data Set 3: name is 2d-4c-no0.data with 1572 points to clustered in different shapes and its output using DBSCAN shown in figure 1002 there is 4 clusters.

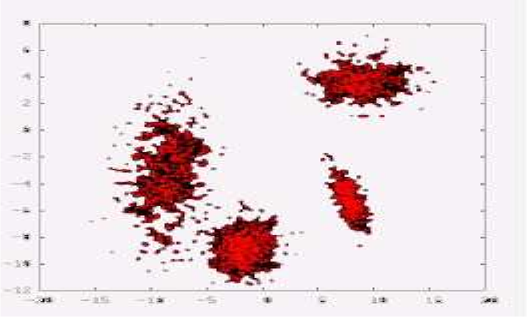

Figure 9: Data set #3 output using DBSCAN

If we apply our algorithm and divide the space to four cells we get four clusters with points as the following:

```
Number of Points to be in
Cluster1: 1119
```

```
Number of Points to be in
Cluster2: 257

Number of Points to be in
Cluster3: 67

Number of Points to be in
Cluster4: 129
```

4.2 Real data sets

1. The Iris1 data set is a data set with 150 random samples of flowers.

2. In the Glass1 data set has 10 attributes with 214 number of instances. In Table2 we assign the result of applying the source code of DBSCAN algorithm and MDBSCAN-UR algorithm on the same data sets Iris and Glass. Table 2 shows the number of clusters in both algorithms for the same data set and the points range in every cluster as example when test the Iris data set using DBSCAN we get three clusters C1, C2 and C3. the Iris has 150 points, the first 50 points appear in cluster C1, and the next 50 points appear in C2 and the last 50 points appear in C3.

.

Table 2. The results of applying MDBSCAN-UR to the Iris and Glass data sets.

Data set	Clusters number	DBSCAN	Error percentage	Clusters number	MDBSCAN-UR	Error percentage
Iris	3	C1:50, C2: 50; C3: 50	30%	3	C1:50, C2: 50; C3: 50	27%
Glass	6	C1:70, C2: 76; C3: 17; C4: 13; C5: 9; C6: 29	33%	5	C1:90, C2: 84; C3: 20; C4: 10; C5: 10	30%

5. Conclusions

In this paper, we illustrated a new multi-density cluster algorithm based on grid and representative points with analyzing DBSCAN algorithm and overcome the problems of it . However unlike the standard methods, our method takes into account, not just the number of local MinPts per cell in the grid, but also the radius from time to time in different cells. We perform an experimental evaluation to the performance of MDBSCAN-UR .The results of our experiments show that MDBSCAN-UR is effective and efficient.

6. References

1. J. W. Han, M. Kanber. Data “Mining: Concepts and Techniques” .Morgan Kaufmann Publishers, 2001.
2. Martin Ester, Hans-Peter Kriegel. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proc of 2nd Int. Conf. on KDD’96, 1996: 226-231.
3. Zeng Donghai. The Study of Clustering Algorithm Based on Grid-Density and Spatial Partition Tree. XiaMen University, PRC, 2006
4. A. H. Pilevar. M. Sukumar. GCHL: a grid-c1ustering algorithm for high-dimensional Data Mining. SBIA 2004, LNAI 3171.
5. S. Guha, R. Rastogi, K. Shim : CURE: An Efficient Clustering Algorithm for Large Databases. Stanford University
6. C. Xiaoyun, M. Yufang, Z. Yan, W. Ping: GMDBSCAN: Multi-Density DBSCAN Cluster Based on Grid, School of Information Science and Engineering, Lanzhou University
7. Asuncion, A. & Newman, D.J. UCI Machine Learning Repository [http://www.ics.uci.edu/ ~mlearn/MLRepository.html]. 2007
8. C. Xiaoyun, M. Yufang, Z. Yan, W. Ping “GMDBSCAN: Multi-Density DBSCAN Cluster Based on Grid” School of Information Science and Engineering, Lanzhou University Lanzhou 730000, PRC China.
9. Zhou S,Zhou A, et al. A Fast Density-Based Clustering Algorithm[J]. Journal of Computer Research and Development, 2003, 37(11):1287-1292

10. Zhou S,Fan Y, Zhou A. SDBSCAN: A Sampling-Based DBSCAN Algorithm for Large-Scale Spatial Databases. Journal of Chinese Computer Systems[J], 2000,21(12): 1270-1274
11. Zhou A, Zhou S, Cao J, et al. Approaches for scaling DBSCAN algorithm to large spatial

database[J].Journal of computer science and technology,2000,15(06): 509-526

12. T. Huang ,Y. Yu, K Li, W. Zeng "Reckon the Parameter of DBSCAN for Multi-density Data Sets with Constraints" Dept. of Computer Science, School of Mathematics & Computer Science Fujian Normal University Fuzhou, China; 2009

13. S. Basu, A. Banerjee and R. Mooney, "Active semisupervision for pairwise constrained clustering," Proc. of the SIAM Int. Conf. on Data Mining (SDM-2004), MIT Press, Apr. 2004, pp.333-344, doi: 10.1.1.5.877.

14. K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl, "Constrained K-Means clustering with background knowledge," Proc. of 18th Int. Conf. on Machine Learning (ICML-2001), Morgan Kaufmann Publishers, Jun. 2001, pp.577-584, doi: 10.1.1.20.7363.

15. N. Grira, Crucianu and N. Boujemaa, "Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration," IEEE Int. Conf. on Fuzzy Systems (Fuzz- IEEE 2005), IEEE Press, May 2005, pp.22-25, doi: 10.1.1.109.938.

16. I. Davidson and S. S. Ravi, "Agglomerative hierarchical clustering with constraints: theoretical and empirical results," Proc. of Principles of Knowledge Discovery from Databases (PKDD 05), Springer, Oct. 2005, pp.59-70, doi: 10.1007/11564126_11.

17. I. Davidson and S. S. Ravi, Intractability and clustering with constraints," Proc. of 24th International Conference on Machine Learning (ICML 2007), ACM Press, Jun. 2007, pp.201-208, doi: 10.1145/1273496.1273522.

18. T. Wei, X. Hui, Z. Shi and W. Jie, "Enhancing semisupervised clustering: a feature projection perspective," Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 07), ACM Press, Aug. 2007, pp. 707-716, doi: 10.1145/1281192.1281268.

19. L. Kaufman and P. J. Rousseeuw, Finding groups in Data: an Introduction to cluster, John Wiley & Sons, 1990.

20. J. Han, M. Kamber, and A. K. H. Tung, Spatial Clustering Methods in data mining: A Survey, Geographic Data Mining and Knowledge Discovery, 2001.

21. P. Bradly, U. Fayyad, and C. Reina, Scaling clustering algorithms to large databases, In proc. 1998 Int. Conf. Knoweldge Discovery and Data mining, 1998.

22. T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH: an efficient data clustering method for very large databases, In Proc. 1996 ACMSIGMOD Int. Conf. Management of data (SIGMOD'96), 1996.

23. S. Guha, R. Rastogi, and K. Shim, Cure : An efficient clustering algorithm for large databases, In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98), 1998.

24. M. Ester, H. P. Kriegel, J. sander, and X. Xu, A density based algorithm for discovering clusters in large spatial databases, In Proc. 1996 Inc. Conf. Knowledge discovery and Data mining (KDD'96).

25. M. Ankerst, M. Breunig, H.P. kriegel, and J. Sander, OPTICS: Ordering points to identify the clustering structure, In Proc. 1999 ACM-SIGMOD Int. Conf. Management of data (SIGMOD'96), 1999.

26. A. Hinneburg and D. A. Keim, An efficient approach to clustering in large multimedia databases with noise, In Proc. 1998 Int. Conf. . Knowledge discovery and Data mining (KDD'98), 1998.

27. W. Wang, J. Yang, and R. Muntz, STING: A statistical information grid approach to spatial data mining". In Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97), 1997.

28. G. Sheikholeslami, S. Chatterjee, and A. Zhang, Wave Cluster : A multi- resolution clustering approach for very large spatial databases, In Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97), 1998.

29. R. Agrawal. J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining application, In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98), 1998.

30. J. W. Shavlik, T.G. Dietterich, Reading in machine learning, 1990.

31. T. Kohonen, Self organized formation of topologically correct feature maps, Biological Cybernetics, 1982.