LA - a Clustering Algorithm with an Automated Selection of Attributes, which is Invariant to Functional Transformations of Coordinates

Mikhail V. Kiselev
Megaputer Intelligence Ltd., 38 B.Tatarskaya, Moscow 113184 Russia
megaputer@glas.apc.org

Sergei M. Ananyan
IUCF, Indiana University, 2401 Sampson Lane, Bloomington, IN 47405 USA
sananyan@indiana.edu

Sergey B. Arseniev
Megaputer Intelligence Ltd., 38 B.Tatarskaya, Moscow 113184 Russia
megaputer@glas.apc.org

ABSTRACT

A clustering algorithm called LA is described. The LA algorithm can be applied to the data represented as a set of values of attributes. The algorithm is based on comparison of the n-dimensional density of the data points in various regions of the space of attributes $p(x_1,...,x_n)$ with an expected homogeneous density obtained as a simple product of the corresponding one-dimensional densities $p_i(x_i)$. The regions with a high value of the ratio $\frac{p(x_1,...,x_n)}{p_1(x_1)...p_n(x_n)}$ are considered to contain clusters. The attributes may be of either numerical or non-numerical (categorical) type. A set of attributes which provides the most contrast clustering is selected automatically. The results obtained with the help of the LA algorithm are invariant to any clustering space coordinate reparametrizations, i. e. to one-dimensional monotonous functional transformations $x' = f(x)$. Another valuable property of the algorithm is the weak dependence of the computational time on the number of data points. The LA algorithm is implemented in the PolyAnalyst data mining system.

KEYWORDS: clustering, PolyAnalyst system

1. Introduction.

Clustering is one of the typical problems solved by data mining methods [Cheeseman 1990; Jain, Dubes 1988]. This is the process of grouping cases or database records into subsets such that the degree of similarity between cases in one group is significantly higher than between members of different groups. These groups are called clusters. An exact definition of the similarity between cases, as well as other details, vary in different clustering methods. For example, some algorithms include every case in some cluster while others recognize only the most dense groups of similar records without including the rest of cases in any cluster; some methods can obtain clusters that overlap and so on.

At present various clustering algorithms are utilized. Most often used algorithms can be roughly associated in the following groups.

1. Joining methods. In these methods smaller clusters are consequently merged in larger clusters. Initially each data point represents a single cluster. During each step of the algorithm two clusters are selected which are closest with respect to some metrics and are unified into one new cluster. The process continues until a certain specified number of clusters remains. Various modifications of the algorithm use different measures of distance between clusters, for example, maximum euclidean distance between points belonging to different clusters, mean euclidean distance or minimum euclidean distance.

2. K-means methods. These methods find an *a priori* specified number of clusters such that the variation of attribute values inside clusters would be significantly less than the variation between clusters. The significance of this difference is evaluated in terms of p-values. The algorithm exchanges data points between clusters in order to increase the clustering significance (to decrease the respective p-value).

3. Seeding algorithms [Milligan 1980]. In these methods a certain number of initially selected data points serve as the seeds for growing clusters. These algorithms also generate an *a priori* specified number of clusters. In the beginning each cluster consists of only one selected point (a seed). During one step of the algorithm each cluster associates a data point which is the nearest one in terms of a certain affinity measure. A cluster stops growing when there remains no more sufficiently close points.

4. Density-based algorithms. The space of attribute values is broken into a set of regions. Those regions which have significantly higher point density are considered as containing clusters of data points.

5. Algorithms based on neural networks. Various neural network architectures have been proposed for the solution of clustering problems. The examples of these architectures are ART (adaptive resonance theory) [Carpenter and Grossberg 1987; Williamson 1995] and its various modifications, SOM (self-organizing maps) [Kohonen 1995], counterpropagation [Hecht-Nielsen 1990], and a number of others.

Yet, despite the variety of the approaches and methods, practical data mining problems require further improvement of clustering algorithms. In our opinion, many modern clustering algorithms have the following weak sides:

1. **High computational complexity**. The computational time of many clustering algorithms depends on the number of records at least as $O(N^2)$. This becomes a very serious problem when databases with a hundred thousand or million records are explored.

2. **Insufficient performance with multi-dimensional data**. In databases where every record contains a large number of numerical, boolean and categorical fields the right choice of attributes for the clustering procedure often determines the quality of the result obtained. Some fields may be completely irrelevant for breaking the data into a set of clusters, some fields may be highly correlated, etc. An automated selection of several attributes most crucial for clustering out of, say, hundreds of fields present in the database would be a very desirable feature for clustering algorithms implemented in a data mining system. Yet only a few of existing algorithms offer such a possibility.

3. **Sensitivity to functional transformations of attributes**. Suppose we would like to find clusters in a database describing customers of some retailer. Every customer is described by her or his age and monthly income. These variables are measured in different units. Since many clustering algorithms use euclidean metrics, which in our case can be written as $dist(R_1,R_2)=\sqrt{A(age_1-age_2)^2+(income_1-income_2)^2}$, a different choice of the constant *A* would give us a different set of clusters. The standard attribute normalization transformation $x'=\frac{x-\bar{x}}{\sigma_x}$ which makes all numerical attributes centered at 0 with the dispersion equal to 1 does not solve the problem. The use of this transformation would imply that the constant *A* depends on dispersions of variables. Therefore the elimination of even a single data point with the value of *x* deviating strongly from $\bar{x}$ may significantly influence the result of clustering. Besides, it is evident that clustering performed in terms of *(age, log(income))* instead of *(age, income)* leads in general to completely different results. Even a linear reparametrization of attributes can change the results. This can hardly be considered a good feature because it makes us doubt that the results of clustering procedure are objective enough.

4. **Lack of effective significance control**. The clustering procedures implemented in many existing data mining systems and statistical packages find clusters even in the data consisting of artificially generated random numbers with a uniform distribution. It would be highly desirable that clusters found by data analysis systems express objective and statistically significant properties of data - not simply statistical fluctuations.

In the present paper we describe a clustering algorithm called LA (the abbreviation stands for Localization of Anomalies - point density anomalies are implied), which is free of the drawbacks listed above. In section 2 we cover the techniques underlying the algorithm. The properties of the LA algorithm are discussed in section 3. Section 4 contains two examples of application of LA algorithm. Finally, we present our conclusion in section 5.

2. Automated clustering of database records including multiple numerical and non-numerical fields.

Prior to discussing our algorithm we say a few words about our understanding of the term "cluster". In many approaches a set of clusters found by the corresponding algorithm should be considered as a property of the concrete dataset which was explored. An individual cluster is characterized completely by the set of datapoints that belong to it.

We consider a cluster as a region in the space of attribute values which has a significantly higher concentration of datapoints than other regions. Thus, it is described mainly by boundaries of this region and it is assumed that other sufficiently representative datasets from the universum of data belonging to the same application domain will also have a higher density of points in this region. Therefore the discovered set of clusters may not include all the records in the database. Beside that, the problem of the determination of statistical significance of the clustering becomes very important.

In our approach each cluster is represented as a union of multi-dimensional rectangular regions described by a set of inequalities $x < a$ or $x \geq a$ for numerical fields x and by a set of equalities $c = A$ for categorical fields c.

Our algorithm is applied to a database DB which can be logically represented as a rectangular table with N rows and M columns. This set of attributes (columns) will be denoted as **A**. First we consider databases with numerical fields only. Some remarks concerning the extension of this method to categorical variables will be given below. Thus, database DB can be represented as a finite set of points in the M-dimensional space $\Re^M$. Coordinates in $\Re^M$ will be denoted as x_i, $i=1,...,M$.

The LA algorithm consists of two logical components. The purpose of the first component is the selection of the best combination of attributes x_i which provides the most significant and contrast clustering. The second component finds clusters in the space of a fixed set of attributes x_i. We begin our consideration with the second component.

Suppose that we fix m attributes from M attributes presented in the database DB. It will become clear later that we should impose the following important limitation on the number of attributes:

$$1 < m \leq \tfrac{1}{2} \log_3 N \ . \qquad (1)$$

The inequality shows that the more records exist in the database, the higher-dimensional clusters can be discovered. This does not mean that the algorithm described below cannot be used for a successful analysis of databases with a greater number of attributes. One should keep in mind that the first component of the LA algorithm selects the best m attributes out of the total number M of all database fields.

Our approach is based on breaking the space of attribute values $\Re^m$ in a certain set of regions $\{E_i\}$ and comparing the density of points in each region E_i. Namely, we cut $\Re^m$ by hyperplanes x_i = const and take the rectangular regions formed by these hyperplanes as E_j. We call such set of regions the grid $\{E_i\}$. The hyperplanes forming the grid may be chosen by various methods. However it is important that datapoints would be distributed among the cells E_i as evenly as possible. We use the system of hyperplanes satisfying the following four conditions:

1. Every axis should be divided by at least three hyperplanes.

2. Every coordinate axis is intersected by the same number of hyperplanes. Let us denote this number H - 1. Thus, the grid is determined by a matrix A_{ij} with m rows and H - 1 columns so that the j-th hyperplane intersecting the i-th coordinate axis is defined by the equation $x_i = A_{ij}$.

3. For each i, the hyperplanes $x_i = A_{ij}$ cut $\Re^m$ to H slices in such way that the numbers of datapoints in different slices would be as close to each other as possible.

4. The number of cells in the grid $\{E_i\}$ should be approximately equal to the average number of datapoints in one cell and therefore should be close to $\sqrt{N}$. This represents a reasonable compromise between the roughness of the found cluster structure and the representativeness of the point subpopulation in each cell.

It can be easily shown that condition (1) is a consequence of these requirements.

Consider one cell E_i. Let n be the number of datapoints in this cell. The cell E_i can be considered as a direct product of the attribute axes segments: $E_i = S_1 \times \ldots \times S_m$. Let us denote the number of points with the value of the j-th attribute falling into the segment S_j as M_j. If the points do not form clusters in the space of attributes x_i which are considered as independent then the relative density of points in E_i, is approximately equal to multiplication of one-dimensional relative densities of points in segments S_j:

$$\frac{n}{N} \approx p_j = \frac{M_1 \ldots M_m}{N^m}. \qquad (2)$$

A significantly higher value of $\frac{n}{N}$ would indicate that E_i should be considered as (a part of) a cluster. In the case of $m = 1$ the approximate equality (2) is trivially exact. Thus the minimum dimension of the clustering space m is 2. To find clusters consisting of rectangular regions with anomalous point density we use the following procedure.

For each cell E_i with the number of points greater than $Np_i = \frac{M_1 \ldots M_m}{N^{m-1}}$ we calculate the probability that the high density of points in this cell is a result of the statistical fluctuation. Namely, we determine for each cell E_i the value of $s_i = b(n, N, \frac{M_1 \ldots M_m}{N^m}) = b(n, N, p_i)$ where $b(k, K, p)$ is a tail area probability of the binomial distribution with the number of trials K and the event probability p. A list of all E_i ordered by ascending values of s_i is created. Denote the ordered sequence of the intervals as $\{E'_j\}$. For each cell E'_j we know the number of points lying in the cell, n_j, and the value of p_j. For each j we calculate value $s_{CUM_j} = b(\sum_{i=1}^{j} n_i, N, \sum_{i=1}^{j} p_i)$. Let us denote the value of j for which s_{CUM_j} is minimal as j_{BEST}; this minimum value of s_{CUM_j} will be denoted as s_{BEST}. This value corresponds to the most contrast, most significant division of all cells E_i into "dense" and "sparse" ones. Let us consider the cells E'_j with $j \le j_{BEST}$. In this set of cells we search for subsets of cells C_k such that all of them satisfy the following conditions: 1) either the subset C_k contains only one cell or for each cell E belonging to the subset C_k there exists another cell in C_k which has a common vertex or border with cell E; 2) if two cells belong to different subsets they have no common vertexes or borders. We call these subsets clusters.

Thus, for each subset $\mathbf{a}$ of attributes $\mathbf{a} \subset \mathbf{A}$, $|\mathbf{a}| = m$ satisfying the condition (1) we can determine a set of clusters $\mathbf{C}(\mathbf{a})$, the clustering significance $s_{BEST}(\mathbf{a})$, and the total number of points in all clusters $K(\mathbf{a})$. Now let us discuss the procedure which selects the best

combination of attributes for clustering. The purpose of this procedure is finding a subset of attributes which has the maximum value of some criterion. In most cases it is natural to choose 1 - s_{BEST} as such a criterion. Other possible variants are the number of points in clusters or the number of clusters. It is often required that the clustering procedure should elicit at least two clusters and also that 1 - s_{BEST} should be greater than a certain threshold confidence level. It is obvious that in order to satisfy the first requirement each coordinate should be divided in at least three sections. Depending on the actual conditions of the data exploration carried out (possible time limitation) various modifications of the procedure can be utilized. We consider two extreme cases.

a. Full search. All combinations of m attributes ($1 < m \le \frac{1}{2}\log_3 N$) are tried. The best combination is selected.

b. Linear incremental search.

Step 1. All combinations of two attributes are tried. The best pair is included in list of selected attributes **SEL**. The respective value of the criterion will be denoted as R(**SEL**).

Step 2. If $|\mathbf{SEL}| \ge \frac{1}{2}\log_3 N$ or **SEL** includes all attributes the process stops and **SEL** is the result.

Step 3. All combinations of attributes consisting of all the attributes from **SEL** plus one attribute not included in **SEL** are tried. Let the best combination be $\mathbf{SEL}' = \mathbf{SEL} \cup \{a\}$. If $R(\mathbf{SEL}') \le R(\mathbf{SEL})$ the process stops and **SEL** is selected as a final set of attributes.

Step 4. Set $\mathbf{SEL} = \mathbf{SEL}'$ and go to Step 2.

An abundance of intermediate variants of this procedure can be constructed.

The presence of non-numerical (unordered) attributes does not change the algorithm significantly. The discreteness of the attribute means that the respective axis is already divided to segments corresponding to different values of the attribute. Thus, in this case the grid cells are defined by inequalities $x < a$ or $x \ge a$ for numerical attributes and by equalities $c = A$ for categorical attributes. Only the following serious modification of the algorithm should be made. Since the values of categorical attributes are unordered the dense grid cells are merged into clusters on the basis of numerical attribute values only. For this reason in the set of attributes selected for clustering at least one attribute should be numerical.

3. Properties of LA algorithm.

It can be easily proven that the considered LA algorithm has the following properties:

1. If we replace a numerical attribute x with its functional derivative $f(x)$, where f is a monotonous function and use $f(x)$ instead of x, this will not change the clustering results. The algorithm will detect the same number of clusters and the same sets of records will enter the same clusters.

2. The computational time depends on the number of records N only weakly. The measurements show that the most time consuming operations are the sorting of the values of attributes when the grid $\{E_i\}$ is constructed and the determination of s_i values for each grid cell E_i. The former operation requires $O(mN\log N)$ time, the latter - $O(\sqrt{N})$. In the asymptotic region $N \to \infty$ the first operation provides the main contribution. The exact computational time of LA algorithm depends on the version of the procedure used for selecting the best attributes. One can see that for a fast linear search the computational time is $O(M^3 N\log N)$; for the most slow full search it is $O(2^M MN\log N)$ (when $N >> 3^{2M}$). We can see that the dependence on the number of records is quite weak while the dependence on the number of fields is much stronger even for the faster modification of the algorithm.

3. The LA algorithm works best in the case of a great number of records. The less records are explored, the less fine cluster structure is recognized. In the worst case, when a cluster of the size approximately equal to one cell is intersected by a hyperplane it may not be detected by the algorithm.

4. The LA algorithm is noise tolerant. Indeed, the algorithm is based not on the distances or other characteristics of single points but on the properties of substantial subsets of data. Thus an addition of a relatively small subpopulation of points with different statistical properties ("noise") cannot influence the results obtained by the algorithm substantially.

4. Examples.

The high efficiency of the LA algorithm for the solution of practical data mining problems has been confirmed by its successful application in several fields. At present the LA algorithm is implemented as a data exploration engine in the PolyAnalyst data mining system [Kiselev, Ananyan, Arseniev 1997; Kiselev 1995]. It was most often used for customer profiling, analysis of electorate and similar problems requiring the exploration of demographic information. The experience of the application of the LA algorithm shows that the algorithm performs best for the analysis of large databases (with 20,000 or more records). In the case of smaller databases the discovered cluster structure may be too rough. In this section we consider two sample clustering problems. The first one illustrates the application of the LA algorithm to the analysis of a demographic database; in the second example a small artificially generated database is explored.

Example 1. In this example we analyzed the public domain data included in the machine learning benchmark database set of the University of California, Irvine available at its WWW site (`http://www.ics.uci.edu/~mlearn/MLSummary.html`). The database named "ADULT" contains general demographic information on 32,561 people, supplemented by the data about their incomes. The data was donated by the US Census Bureau.

The results of clustering this database using the LA method are shown in Table 1. Two attributes have been chosen for clustering: age (columns) and education level expressed by numbers (rows). Each cell displays the number of cases (database records) belonging to the cell. The shaded cells correspond to four clusters found. We can see a cluster that includes high school and college students, a cluster that consists of older people having professional school education, the largest cluster that consists of qualified professionals whose education increases proportionally to age, etc. We can see that the algorithm is based on a relative density of points. Say, the upper left cell with 296 records is included into cluster while the cell two rows below, containing 663 records, is not included.

Table 1. Clustering US Census Bureau database using LA algorithm.

	<20.5	20.5-23.5	23.5-26.5	26.5-29.5	29.5-32.5	2.5-35.5	5.5-38.5	38.5-41.5	41.5-44.5	44.5-48.5	48.5-53.5	53.5-59.5	>=59.5
<6.5	296	147	138	173	162	143	147	127	121	172	243	303	473
6.5-8.5	558	96	102	89	96	102	82	62	55	81	97	79	109
8.5-9.5	663	702	767	851	861	941	865	769	655	836	920	781	890
9.5-10.5	861	958	558	504	533	520	519	478	497	532	493	411	427
10.5-12.5	27	175	190	239	250	239	258	235	184	231	184	110	127
12.5-13.5	3	278	592	514	514	479	492	452	458	503	413	323	334
>=13.5	2	6	77	145	161	213	220	295	304	367	366	272	284

Example 2. Using this example we try to illustrate the limits of applicability of the LA algorithm, and measure its ability to detect small diffuse clusters in small databases. This example deals with the artificially generated random data including two numerical fields - x and y, $0 \le x, y \le 1$. The set of points includes four subsets with different point distribution laws:

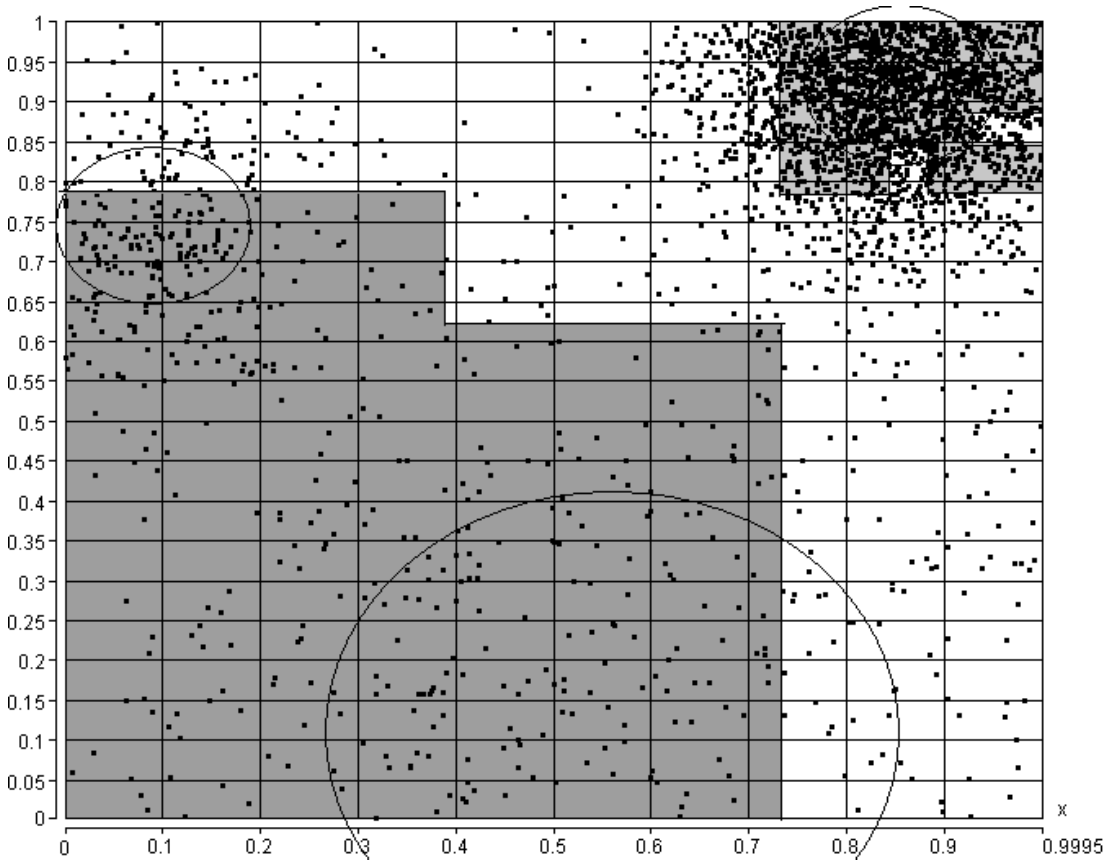

Figure 1. Clustering randomly generated data.

a. Uniformly distributed points, N = 300 (“noise”).

b. Normally distributed points with the center at (0.09, 0.743) and dispersion 0.1, N = 253 (“small sharp cluster”).

c. Normally distributed points with the center at (0.86, 0.926) and dispersion 0.1, N = 2150 (“large sharp cluster”).

d. Normally distributed points with the center at (0.571, 0.114) and dispersion 0.3, N = 184 (“small diffuse cluster”).

The total number of points is 2,887. The results obtained by the LA algorithm are shown on Figure 1. The position and dispersion of the three clusters are denoted by circles. The LA algorithm elicits two clusters. The respective rectangular regions are shaded on the figure. We see that while the algorithm localized the large sharp cluster satisfactorily, it could not distinguish between two other clusters and merged them in one cluster. Since these clusters include much fewer points, the size of the grid cells in the regions with low and average values of x and y is quite large when compared with the size of the clusters. This fact complicates the exact identification of clusters b and d, which is a difficult task because of the diffuseness of these clusters. Nevertheless, the obtained result allows us to conclude that while the main application area of the LA algorithm is large databases it works reasonably well also in the case of a few thousand records.

5. Conclusion.

We have described a new algorithm for finding clusters in data called LA. Our algorithm can select automatically an optimal subset of the database fields for clustering. The algorithm is invariant to a monotonous functional transformation of numerical attributes and has a weak dependence of the computational time on the number of records in the database. The algorithm is based on the comparison of the *n*-dimensional density of the data points in various regions of the space of attributes with an expected homogeneous density obtained as a simple product of the corresponding one-dimensional densities. The LA algorithm is implemented as a component of the PolyAnalyst data mining system. As a module of this system it has been successfully applied in the fields of banking, database marketing, and sociological studies. An empirical evaluation shows that the algorithm furnishes the best results when relatively large databases (with 20,000 or more records) are explored.

References.

1.Carpenter, G. and Grossberg, S. A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine, *Computer Vision, Graphics, and Image Processing*, 37:54-115, 1987.

2. Cheeseman, P. On Finding the Most Probable Model. In: *Computational Models of Scientific Discovery and Theory Formation*, Shrager, J. and Langley, P. (eds.). Los Gatos, CA: Morgan Kaufmann, pp 73-95, 1990.

3. Hecht-Nielsen, R. *Neurocomputing*, Reading, MA: Addison-Wesley, 1990.

4. Jain, A. K. and Dubes, R. C. *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

5. Kiselev, M.V. PolyAnalyst 2.0: Combination of Statistical Data Preprocessing and Symbolic KDD Technique, In: *Proceedings of ECML-95 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, Heraklion, Greece, pp. 187-192, 1995.

6. Kiselev, M.V., Ananyan, S. M., and Arseniev, S. B. Regression-Based Classification Methods and Their Comparison with Decision Tree Algorithms, In: *Proceedings of 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, Trondheim, Norway, Springer, pp 134-144, 1997.

7. Kohonen, T. *Self-Organizing Maps*, Berlin: Springer-Verlag, 1995.

8. Milligan, G.W. An estimation of the effect of six types of error perturbation on fifteen clustering algorithms, *Psychometrika*, vol 45, pp 325-342, 1980.

9. Williamson, J. R. Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. Technical Report CAS/CNS-95-003, Boston University, Center of Adaptive Systems and Department of Cognitive and Neural Systems, 1995.