Density-Based Multiscale Data Condensation

Pabitra Mitra, *Student Member*, *IEEE*, C.A. Murthy, and Sankar K. Pal, *Fellow*, *IEEE*

Abstract—A problem gaining interest in pattern recognition applied to data mining is that of selecting a small representative subset from a very large data set. In this article, a nonparametric data reduction scheme is suggested. It attempts to represent the density underlying the data. The algorithm selects representative points in a multiscale fashion which is novel from existing density-based approaches. The accuracy of representation by the condensed set is measured in terms of the error in density estimates of the original and reduced sets. Experimental studies on several real life data sets show that the multiscale approach is superior to several related condensation methods both in terms of condensation ratio and estimation error. The condensed set obtained was also experimentally shown to be effective for some important data mining tasks like classification, clustering, and rule generation on large data sets. Moreover, it is empirically found that the algorithm is efficient in terms of sample complexity.

Index Terms—Data mining, multiscale condensation, scalability, density estimation, convergence in probability, instance learning.

1 Introduction

The current popularity of data mining and data warehousing, as well as the decline in the cost of disk storage, has led to a proliferation of terabyte data warehouses [1]. Mining a database of even a few gigabytes is an arduous task for machine learning techniques and requires advanced parallel hardware and algorithms. An approach for dealing with the intractable problem of learning from huge databases is to select a small subset of data for learning [2]. Databases often contain redundant data. It would be convenient if large databases could be replaced by a small subset of representative patterns so that the accuracy of estimates (e.g., of probability density, dependencies, class boundaries) obtained from such a reduced set should be comparable to that obtained using the entire data set.

The simplest approach for data reduction is to draw the desired number of random samples from the entire data set. Various statistical sampling methods such as random sampling, stratified sampling, and peepholing [3] have been in existence. However, naive sampling methods are not suitable for real-world problems with noisy data since the performance of the algorithms may change unpredictably and significantly [3]. Better performance is obtained using *uncertainty sampling* [4] and active learning [5], where a simple classifier queries for informative examples. The random sampling approach effectively ignores all the information present in the samples not chosen for membership in the reduced subset. An advanced condensation algorithm should include information from all samples in the reduction process.

Some widely studied schemes for data condensation are built upon classification-based approaches, in general, and the k-NN rule, in particular [6]. The effectiveness of the condensed set is measured in terms of the classification accuracy. These methods attempt to derive a minimal consistent set, i.e., a minimal set which correctly classifies all the original samples. The very first developement of this kind is the condensed nearest neighbor rule (CNN) of Hart [7]. Other algorithms in this category including the reduced nearest neighbor and iterative condensation algorithms are summarized in [8]. Recently, a local asymmetrically weighted similarity metric (LASM) approach for data compression [9] is shown to have superior performance compared to conventional k-NN classification-based methods. Similar concepts of data reduction and locally varying models based on neural networks are discussed in [10], [11], [12].

The classification-based condensation methods are, however, specific to (i.e., dependent on) the classification tasks and the models (e.g., k-NN, perceptron) used. Data condensation of more generic nature is performed by classical vector quantization methods [13] using a set of codebook vectors which minimize the quantization error. An effective and popular method of learning the vectors is by using the self-organizing map [14]. However, if the self-organizing map is to be used as a pattern classifier, the codebook vectors may be further refined using the learning vector quantization algorithms [14]. Competitive learning [15] can also be used to obtain such representative points. These methods are seen to approximate the density underlying the data [14]. Since learning is inherent in the methodologies, the final solution is dependent on initialization, choice of learning parameters, and the nature of local minima.

Another group of generic data condensation methods are based on the density-based approaches which consider the density function of the data for the purpose of condensation rather than minimizing the quantization error. These methods do not involve any learning process and, therefore, are deterministic, (i.e., for a given input data set, the output condensed set is fixed). Here, one estimates the density at a point and selects the points having "higher" densities, while ensuring a minimum separation between the selected points. These methods bear resemblance to density-based clustering techniques like the DBSCAN algorithm [16],

• *The authors are with the Machine Intelligence Unit, Indian Statistical Institute, Calcutta 700 035, India.*
E-mail: {pabitra_r, murthy, sankar}@isical.ac.in.

Manuscript received 28 Nov. 2000; revised 29 Mar. 2001; accepted 3 Oct. 2001.
Recommended for acceptance by C. Brodley.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 113205.

popular for spatial data mining. DBSCAN is based on the principle that a cluster point contains in its neighborhood a minimum number of samples, i.e., the cluster point has density above a certain threshold. The neighborhood radius and the density threshold are user specified. Astrahan [17] proposed a classical data reduction algorithm of this type in 1971, in which he used a disc of radius d_1 about a point to obtain an estimate of density at that point. The points are sorted based on these estimated densities and the densest point is selected, while rejecting all points that lie within another disc of radius d_2 about the selected point. The process is repeated until all the samples are covered. However, selecting the values of d_1 and d_2 is a nontrivial problem. A partial solution using a minimal spanning tree-based method is described in [18]. Though the above approaches select the points based on the density criterion, they do not directly attempt to represent the original distribution. The selected points are distributed evenly over the entire feature space irrespective of the distribution. A constant separation is used for instance pruning. Interestingly, Fukunaga [19] suggested a nonparametric algorithm for selecting a condensed set based on the criterion that density estimates obtained with the original set and the reduced set are *close*. However, the algorithm is search-based and requires large computation time.

Efficiency of condensation algorithms may be improved by adopting a multiresolution representation approach. A multiresolution framework for instance-based learning and regression has been studied in [20] and [21], respectively. It uses a k-d tree to impose a hierararchy of data partitions which implicitly condense the data into homogeneous blocks having variable resolutions. Each level of the tree represents a partition of the feature space at a particular scale of detail. Prediction for a query point is performed using blocks from different scales; finer scale blocks are used for points close to the query and cruder scale blocks for those far from the query. However, the blocks are constructed by simple median splitting algorithms which do not directly consider the density function underlying the data. We propose in this article a density-based multiresolution data reduction algorithm that uses discs of adaptive radii for both density estimation and sample pruning. The method attempts to accurately represent the entire distribution rather than the data set itself. The accuracy of this representation is measured using nearest-neighbor density estimates at each point belonging to the entire data set. The method does away with the difficult choice of radii d_1 and d_2 as in Astrahan's method discussed above. In the proposed method, k-NN density estimates are obtained for each point and the points having higher density are selected subject to the condition that the point does not lie in a region "covered" by any other selected point. A selected point "covers" a disc around it with area inversely proportional (by a factor σ, say) to the (estimated) density at that point, as illustrated in Fig. 1. Hence, the regions having higher density are represented more accurately in the reduced data sets compared to sparse regions. The proportionality factor (σ) and k used for k-NN density estimation controls the condensation ratio and the accuracy of representation.

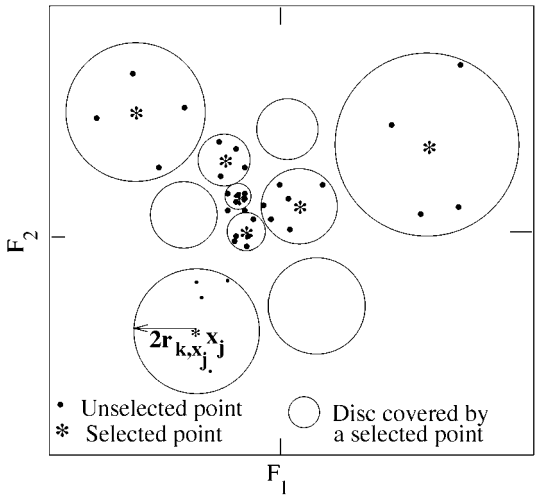

Fig. 1. Multiresolution data reduction.

The condensation algorithm can obtain reduced sets which represent the data at different scales. The parameter k acts as the scale parameter and the data is viewed at varying degrees of detail depending on the value of k. This type of multiscale representation of data is desirable for various applications like data mining. At each scale, the representation gives adequate importance to different regions of the feature space based upon the probability density as mentioned before. The above scheme induces a scale which is both efficient in terms of density estimation error and natural to the data distribution.

It is observed from experiments that the multiresolution approach helps to achieve lower error with similar condensation ratio compared to several related data condensation schemes. The reduced set obtained was found to be effective for a number of data mining applications like classification, clustering, and rule generation. The suggested algorithm is also found to be scalable and efficient in terms of sample complexity, in the sense that the error level decreases quickly with the increase in size of the condensed set. In the next section, we describe briefly aspects of multiscale representation.

2 Multiscale Representation of Data

Multiscale representation of data refers to visualization of the data at different "scales," where the term scale may signify either unit, frequency, radius, window size, or kernel parameters. The importance of scale has been increasingly acknowledged in the past decade in the areas of image and signal analysis and computer vision with the development of several scale inspired models like pyramids, wavelets, and multiresolution techniques. Recently, scale-based methods have also become popular in clustering [22] and density estimation. In these methodologies, the concept of scale has been implemented using variable width Radial Basis Function Network [23], annealing-based clustering with variable temperature [24], and variable window density estimates.

The question of scale is natural to data condensation. At a very coarse scale, the entire data may be represented by only a few number of points and at a very fine scale all the sample points may constitute the condensed set, the scales

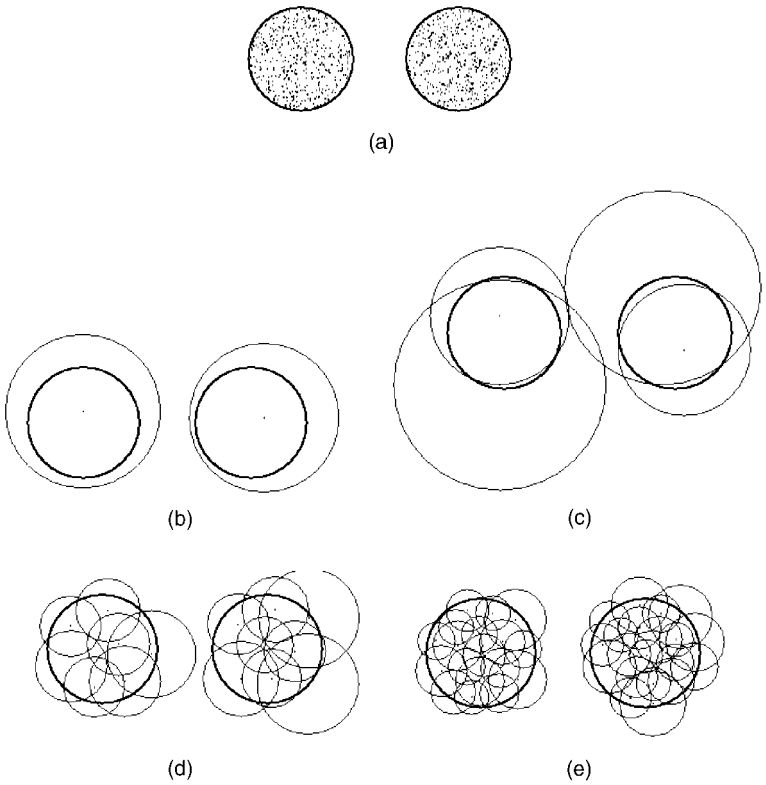

Fig. 2. Representation of a data set at different levels of detail by the condensed sets. "." is a point belonging the condensed set, the circles about the points denote the discs covering that point. The two bold circles denote the boundaries of the data set.

in between representing varying degrees of detail. In many data mining applications (e.g., structure discovery in remotely sensed data, identifying population groups from census data), it is neccesary that the data be represented in varying levels of detail. Data condensation is only a preliminary step in the overall data mining process and several higher-level learning operations may be performed on the condensed set later. Hence, the condensation algorithm should be able to obtain representative subsets at different scales, as demanded, in an *efficient* manner.

The proposed method for data condensation, discussed in Section 1, obtains condensed sets of different degrees of detail by varying a scale parameter k. It may be noted that such variable detail representation may be achieved by other approaches also, including random sampling. However, unlike random sampling, the scales induced by the proposed method are not prespecified by the sizes of the condensed sets but follow the natural characteristics of the data. As far as efficiency of the scaling procedure is concerned, it may be noted that, in most of the multiscale schemes for representing data or signal, including wavelets, efficiency is achieved by a lenient representation of the "unimportant" regions and a detailed representation of the "important" regions, where the notion of importance may vary from problem to problem. We have followed a similar principle in the proposed condensation algorithm where, at each scale, the different regions of the feature space are represented in the condensed set based on the densities of those regions estimated at that particular scale. Fig. 2 illustrates the concept of variable scale representation. The data consists of 2,000 points selected randomly from two nonoverlapping circles of radius 1 unit and centers at (2, 0) and (5, 0), respectively, (Fig. 2a). Figs. 2b, 2c, 2d, and 2e show representation of the data by condensed sets at different levels of detail. It can be seen that in Fig. 2b only two points cover the entire data set. In Fig. 2c, four points are used to represent the entire data set. Fig. 2d and 2e are more detailed representations of the data.

For a particular scale, the basic principle of the proposed data condensation algorithm involves sorting the points based on *estimated densities*, selecting the denser points, and removing other points that lie within certain distances of the selected points in a multiresolution manner. A nonparametric method of estimating a probability density function is the k-nearest-neighbor method. In a k-NN-based estimation technique, the density of a point is computed based upon the area of disc about that point which includes a fixed number, say k, other points [25]. Hence, the radius of the disc is smaller in a densely populated region than in a sparse region. The area of the disc is inversely proportional to the probability density function at the center of the disc. This behavior is advantageous for the present problem from the point of view of multiresolution representation over different regions of feature space. This is the reason that the k-NN density estimate is considered in the proposed condensation algorithm.

Before we present the data condensation algorithm, we describe in brief the k-NN-based density estimation technique in the next section.

3 Nearest-Neighbor Density Estimate

Let $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}$ be independent observations on a p-dimensional random variable $\mathbf{X}$, with a continuous

probability density function f. The problem is to estimate f at a point $\mathbf{z}$.

Let $d(\mathbf{x}, \mathbf{z})$ represent the Euclidean distance between $\mathbf{x}$ and $\mathbf{z}$. A p-dimensional hypersphere of radius r about $\mathbf{z}$ is designated by $S_{r,\mathbf{z}}$, i.e., $S_{r,\mathbf{z}} = \{\mathbf{x} | d(\mathbf{x}, \mathbf{z}) \leq r\}$. The volume or Lebesgue measure of the hypersphere $S_{r,\mathbf{z}}$ will be called A_r. Let us describe a nonparametric method for estimating f suggested by Loftsgaarden [25].

Let $k(N)$ be a sequence of positive integers such that $\lim_{N\to\infty} k(N) = \infty$ and $\lim_{N\to\infty}\ k(N)/N = 0$. Once $k(N)$ is chosen and a sample $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}$ is available, $r_{k(N),\mathbf{z}}$ is determined as the distance from $\mathbf{z}$ to the $(k(N)+1)$th nearest neighbor of $\mathbf{z}$ among $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N$. Hence, an estimate of f is given by

$$\hat{f}_N(\mathbf{z}) = \frac{k(N)}{N} \times \frac{1}{A_{r_{k(N),\mathbf{z}}}}. \tag{1}$$

It can be proven [25] that the density estimate given by (1) is asymptotically unbiased and consistent. It may, however, be noted that k-NN estimates suffer from the "curse of dimensionality" problem in high-dimensional spaces [26].

A condensation algorithm should obtain a subset which is representative of the original data distribution. We discuss some measures of the accuracy of such representations in terms of the error in k-NN density estimate discussed above.

3.1 Measures of Error in Density Estimate

Let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ be N independent samples drawn from a distribution f. The closeness between two estimates g_1 and g_2 of f is measured by a criterion of the form

$$\hat{J} = \frac{1}{N}\sum_{i=1}^{N} D(g_1(\mathbf{x}_i), g_2(\mathbf{x}_i)),$$

where $\mathbf{x}_i$ is the ith sample, and $D(.,.)$ is a measure of the distance between $g_1(\mathbf{x}_i)$ and $g_2(\mathbf{x}_i)$. It may be noted that $\hat{J}$ is a random variable and an estimate of the quantity J, where

$$J = E(\hat{J}) = \int D(g_1(\mathbf{z}), g_2(\mathbf{z}))f(\mathbf{z})d\mathbf{z}.$$

In our case, we have a density estimate $\hat{f}_N$ for f, from $\{\mathbf{x}_1, \ldots \mathbf{x}_N\}$ using the k-NN density estimation method already described. Any other density estimation technique, like kernel estimates, may also be considered. Now, we like to choose n points, $n << N$, from $\mathbf{x}_1, \ldots, \mathbf{x}_N$ such that the density estimate $\hat{\alpha}_n$ obtained from these n points is close to $\hat{f}_N$, where n is not predetermined. In the next section, we present a method that automatically provides the value for n and the set of n points for a given $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. It may be noted that $\hat{J}$ measures the difference between estimates $\hat{f}_N$ and $\hat{\alpha}_n$ and not the error of each of these estimates with respect to the actual distribution. However, if N is large, it is known that $\hat{f}_N$ is a consistent estimate of f [25] (for suitable values of k, as mentioned in (1)). Hence, a small value of $\hat{J}$ indicate, closeness of $\hat{\alpha}_n$ to the actual distribution f.

For D, we use the form similar to log-likelihood ratio used in classification [19],

$$D(\hat{f}_N(\mathbf{x}_i), \hat{\alpha}_n(\mathbf{x}_i)) = \left|\ln \frac{\hat{f}_N(\mathbf{x}_i)}{\hat{\alpha}_n(\mathbf{x}_i)}\right|, \tag{2}$$

A second possibility is a modified version of the kernel of the Kullback-Liebler information number [19] which attaches more weight to the high-density region of the distribution

$$D(\hat{f}_N(\mathbf{x}_i), \hat{\alpha}_n(\mathbf{x}_i)) = \left|\hat{\alpha}_N(\mathbf{x}_i).\ln \frac{\hat{f}_N(\mathbf{x}_i)}{\hat{\alpha}_n(\mathbf{x}_i)}\right|. \tag{3}$$

We use both of these quantities to measure the efficacy of the reduction algorithms in subsequent sections. If the estimates are close enough, both the quantities are close to zero.

4 Proposed Data Reduction Algorithm

The proposed data reduction algorithm involves estimating the density at a point using the methods described in the previous section, sorting the points based on the density criterion, selecting a point according to the sorted list, and pruning all points lying within a disc about a selected point with radius inversely proportional to the density at that point.

Method. Let $B_N = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$ be the original data set. Choose a positive integer k.

1. For each point $\mathbf{x_i} \in B_N$, calculate the distance of the kth nearest neighbor of $\mathbf{x}_i$ in B_N. Denote it by $r_{k,\mathbf{x_i}}$.
2. Select the point $\mathbf{x_j} \in B_N$, having the lowest value of $r_{k,\mathbf{x_j}}$ and place it in the reduced set E. Ties in lowest value of $r_{k,\mathbf{x_j}}$ may be resolved by a convention, say, according to the index of the samples. From (1), it is evident that $\mathbf{x_j}$ corresponds to the point having the highest density $\hat{f}_N(\mathbf{x_j})$.
3. Remove all points from B_N that lie within a disc of radius $2r_{k,\mathbf{x_j}}$ centered at $\mathbf{x_j}$ and the set consisting of the remaining points be renamed as B_N. Note that since $r^p_{k,\mathbf{x_j}}$ (where p is the dimension of the feature space) is inversely proportional to the estimate of the probability density at $\mathbf{x_j}$, regions of higher probability density are covered by smaller discs, and sparser regions are covered by larger discs. Consequently, more points are selected from the regions having higher density.
4. Repeat Step 2 on B_N until B_N becomes a null set.

Thus, the $\mathbf{x}_j$s selected and the corresponding $r_{k,\mathbf{x}_j}$ constitute the condensed (reduced) set.

The procedure is illustrated in Fig. 1 in $F_1 - F_2$ space. As shown in the figure, each selected point (marked "*") is at the center of a disc that covers some region in the feature space. All other points (marked as ".") lying within the disc except the center is discarded. It can be seen that selected points lying in high-density region have discs of smaller radii, while points in sparser region correspond to larger discs, i.e., the data is represented in a multiscale manner over the feature space.

Remarks.

1. The algorithm not only selects the denser data points, but does so in a manner such that the separation between two points is inversely proportional to the probability density of the points. Hence, regions in the feature space having higher density are represented by more points than sparser regions.

This provides a better representation of the data distribution than random sampling because different regions of the feature space are given variable importance on the basis of the probability density of that region, i.e., the representation is multiresolution. A technique for performance enhancement and computational time reduction using such multiresolution representation is discussed in [20].

2. The condensed set obtained may be used to obtain an estimate of the probability density function of the data. This may be done using the k-NN density estimation method discussed in Section 3.
3. The parameter k acts as a scale-parameter for the condensation algorithm. The size of the neighborhood, used for density estimate, as well as the pruning radii are dependent on k and, therefore, vary with scale. The smaller the value of k, the more refined is the scale and vice versa. However, independent of the chosen scale, the representation gives adequate importance to the different regions of the feature space depending on their estimated densities at that scale. This type of multiresolution representation helps preserve salient features which are natural to the data over a wide range of scales. In many situations, the scale to be used for condensation is dictated by the application. However, if no such application specific requirements exist, the condensed set may be selected from the region where the error versus scale curve (which is exponentially decaying in nature) begins to flat off.
4. It may be noted that the choice of k is a classical problem for k-NN-based methods for finite sample sets. Theoretically, the value of k should increase with the size of the data set (N), but at a slower rate than N itself [26]. For data condensation using the proposed method, it has also been observed that the value of k should be increased as the data set size N increases to achieve a constant condensation ratio (CR), though the exact nature of the k versus CR curve is distribution dependent. In the experimental results presented in Section 5.5, we observe that at high values of k (i.e., low values of CR) the k versus CR curve is sufficiently robust over different data sets.
5. The accuracy of k-NN density depends on the value of k used. Admissible values of k may be obtained from considerations discussed above. However, for very small data sets or condensed sets, the choice of lower admissible limit of k is dictated by the data set size.

5 Experimental Results

In this section, we present the results of experiments conducted on some well-known data sets of varying dimension and size. The data sets are described in Table 1. Among them, the Forest Cover Type data represents forest cover of 30 m $\times$ 30 m cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS). It contains 581,012 instances having 54 attributes representing cartographic variables. Each observation is labeled as belonging to one of the 7 different classes (forest cover types). This data set is available from the UCI KDD data set archive. Among the other data sets, the Satellite Image data consists of four 512×512 gray-scale images of different spectral bands obtained by the Indian Remote Sensing satellite of the city of Calcutta in India. Each pixel represents a 36.25 m $\times$ 36.25 m region. The third large data set used is the PUMS census data for the Los Angeles and Long Beach area. The data contains 133 attributes, mostly categorical, and 320,000 samples were used. This data set is also available from the UCI KDD archive. The other data sets, e.g., Wisconsin breast cancer (medical domain data), Pima Indian (also, medical domain data), Vowel (speech data), Iris (flower classification data), ringnorm and twonorm (artificial data), are benchmark data sets widely used in literature. The *Norm* data was artificially generated by drawing 500 i.i.d samples from a normal distribution with

$$\text{mean} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{and covariance matrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

TABLE 1
Data Sets Used in Experiments

Dataset	Samples	Features	Source
Forest Cover	581012	54	UCI KDD Archive
PUMS Census	320000	133	UCI KDD Archive
Satellite Image	262144	4	Reference [27]
Ringnorm	7400	20	Reference [28]
Twonorm	7400	20	Reference [28]
Wisconsin Cancer	684	9	UCI Archive
Pima Indian	768	8	UCI Archive
Vowel	871	3	Reference [29]
Monks-2	432	6	UCI Archive
Iris	150	4	UCI Archive
Norm	500	2	Artificial

The organization of the results is as follows: First, we present and compare the results concerning error in density estimate and condensation ratio for all 10 data sets. Next, we demonstrate the efficacy of our condensation method for three diverse tasks, namely, classification, clustering, and rule generation, on the three large data sets. The Forest Cover Type data is considered to evaluate the *classification* performance, the Satellite Image data is considered for *clustering*, and the PUMS Los Angeles Census data is considered for *rule generation*. Our choice of tasks for the three large data sets described above has been guided by studies performed on them in existing literature as well as the nature of the data sets. Finally, we empirically study the scalability property of the algorithm in terms of sample complexity, i.e., the number of samples in the condensed set required to achieve a particular accuracy level.

5.1 Density Estimation

Here, we compare the error between density estimates obtained using the original data set and the reduced set. The proposed algorithm is compared with three representative data reduction schemes (random sampling, vector quantization-based, and clustering-based) described below. Classification-based data reduction methods like Condensed Nearest Neighbor are not compared, as error in

TABLE 2
Comparison of k-NN Density Estimation Error of Condensation Algorithms (Lower CR)

Dataset	Multiscale Algorithm			Uniform Scale method [17]			SOM			Random sampling		
	CR	LLR	KLI	CR	LLR	KLI	CR	LLR	KLI	CR	LLR	KLI
	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD		Mean SD	Mean SD		Mean SD	Mean SD
Norm	3.0 0.001	1.16 0.09	0.16 0.04	3.0 0.001	1.33 0.12	0.20 0.04	3.0	**1.21 0.08**	**0.17 0.004**	3.0	1.38 0.27	0.25 0.10
					(3.76, 1.72)	(2.34, 1.71)		**(1.69, 1.73)**	**(0.02, 1.81)**		(2.56, 1.78)	(2.77, 1.77)
Iris	2.5 0.000	1.83 0.08	0.40 0.04	2.5 0.000	2.02 0.17	0.68 0.08	2.5	2.00 0.01	0.44 0.005	2.5	2.85 0.98	1.01 0.23
					(3.35, 1.76)	(10.38, 1.76)		(7.0, 1.81)	(3.29, 1.81)		(3.44, 1.81)	(8.66, 1.81)
Vowel	3.4 0.00	1.40 0.16	0.10 0.01	3.4 0.001	1.67 0.28	0.165 0.01	3.4	**1.43 0.005**	0.11 0.00	3.4	1.95 0.55	0.41 0.11
					(2.77, 1.74)	(15.24, 1.71)		**(0.88, 1.81)**	(3.32, 1.81)		(3.18, 1.78)	(9.30, 1.81)
Pima	3.2 0.002	1.15 0.11	18.1 1.03	3.2 0.001	1.31 0.17	21.1 4.0	3.2	1.24 0.04	20.4 1.01	3.2	1.99 0.91	25.1 9.1
					(2.62, 1.73)	(2.41, 1.81)		(2.55, 1.78)	(5.22, 1.71)		(3.04, 1.81)	(2.53, 1.81)
Cancer	4.3 0.002	1.37 0.17	17.1 1.4	4.3 0.003	1.61 0.28	19.0 1.04	4.3	1.54 0.11	19.4 0.50	4.3	1.805 0.57	24.0 9.01
					(2.43, 1.76)	(3.80, 1.72)		(2.23, 1.81)	(5.35, 1.78)		(2.43, 1.81)	(2.54, 1.81)
Monk	4.1 0.00	0.64 0.01	0.65 0.04	4.1 0.001	0.70 0.04	0.72 0.05	4.1	0.67 0.01	0.68 0.01	4.1	0.83 0.16	0.88 0.16
					(4.82, 1.81)	(3.62, 1.72)		(7.03, 1.71)	(2.41, 1.81)		(1.86, 1.81)	(2.61, 1.81)
Tnorm	1.0 0.00	0.43 0.01	1.70 0.10	1.0 0.00	0.57 0.07	1.97 0.17	1.0	0.46 0.00	1.81 0.01	1.0	0.59 0.19	2.01 0.56
					(6.56, 1.81)	(4.54, 1.73)		(9.95, 1.81)	(3.30, 1.81)		(5.86, 1.81)	(1.81, 1.78)
Rnorm	2.0 0.00	0.40 0.05	2.11 0.22	2.0 0.001	0.54 0.07	2.95 0.22	2.0	**0.41 0.001**	2.24 0.001	2.0	0.70 0.15	3.01 0.91
					(5.40, 1.73)	(8.95, 1.71)		**(0.63, 1.81)**	(1.96, 1.81)		(6.23, 1.78)	(3.19, 1.81)
Forest	0.1 0.001	0.82 0.01	2.71 0.02	0.1 0.004	2.0 0.02	4.7 0.55	0.1	1.40 0.00	3.20 0.01	0.1	3.8 1.7	7.0 2.50
					(175, 1.76)	(11.99, 1.81)		(192.36, 1.81)	(72.68, 1.76)		(5.81, 1.81)	(5.69, 1.81)
Sat.Img.	0.2 0.001	0.78 0.01	1.18 0.09	0.2 0.002	0.92 0.02	1.40 0.25	0.2	0.88 0.01	1.28 0.00	0.2	1.09 0.15	1.79 0.27
					(20.76, 1.76)	(8.21, 1.81)		(23.45, 1.71)	(3.68, 1.81)		(6.84, 1.81)	(7.10, 1.78)
Census	0.1 0.002	0.27 0.00	1.55 0.10	0.1 0.004	0.31 0.02	1.70 0.15	0.1	0.30 0.01	1.61 0.01	0.1	0.40 0.17	1.90 0.45
					(6.63, 1.81)	(2.76, 1.72)		(14.07, 1.81)	(1.98, 1.81)		(2.53, 1.81)	(2.52, 1.81)

"CR" denotes condensation ratio in %, "LLR" denotes the log-likelihood error and "KLI" denotes the Kullback-Liebler information number, the numbers in the parenthesis indicate the computed and tabled values of the test statistic, respectively. A higher computed value compared to tabled value indicates statistical significance. The values marked bold denote lack of statistical significance.

density estimates is not the optimality criterion for such methods. The methods compared are: random sampling with replacement, the self-organizing map (SOM) [14], and Astrahan's clustering-based uniform scale method [17]. In Astrahan's method (explained in Section 1), for the purpose of density estimation, we use radius

$$d_1 = \sqrt{sup_{i=1,..,n}(inf_{j=1,..,n} d(\mathbf{x_i}, \mathbf{x_j}))}$$

and radius $d_2 = \gamma d_1$ for pruning, where γ is a tunable parameter controlling the condensation ratio. The above expression for d_1 produces a radius close to that obtained using the minimal spanning tree-based method described in [18]. The following quantities are compared for each algorithm:

1. The condensation ratio (CR), measured as the ratio of the cardinality of the condensed set and the original set, expressed as percentage.
2. The log-likelihood (LLR) ratio for measuring the error in density estimate with the original set and the reduced set, as described in (2).
3. The Kullback-Liebler information number (KLI), also for measuring the error in density estimate ((3)).

In our experiments for each data, 90 percent of the samples are selected as training set and the remaining samples are used as test set. Eleven such independent random training-test set splits are obtained and the mean and standard deviation (SD) of the errors are computed over 11 runs. (Sample size of 11 is considered to have a degree of freedom $11 - 1 = 10$, corresponding to which test statistics are available in tables.) Tests of significance were performed for the inequality of means (of the errors) obtained using the proposed algorithm and the other condensation schemes compared. Since both mean pairs and the variance pairs are unknown and different, a generalized version of t-test is appropriate in this context. The above problem is the classical Behrens-Fisher problem in hypothesis testing, a suitable test statistic is described and tabled in [30] and [31], respectively.[1] In Tables 2, 3, 4, and 5, we report, along with the individual means and SDs, the value of test statistic computed and the corresponding tabled values at an error probability level of 0.05. If the computed value is greater than the tabled value, the means are significantly different.

The experiments have been performed for different values of condensation ratios and for each algorithm. However, in Tables 2 and 3, comparison is presented on the basis of error in density estimate for similar values of CR. Alternatively, one could have also compared CR for similar values of error in density estimate. In Tables 2 and 3, results are presented for two different sets of values of CR, about 0.1-3 percent and about 5-20 percent (of the original data set and not the training set), respectively. Experiments were also performed for other values of condensation ratio, e.g., 40 percent and 60 percent for space limitations, we do

1. The test statistic is of the form $v = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\lambda_1 s_1^2 + \lambda_2 s_2^2}}$, where $\bar{x}_1, \bar{x}_2$ are the means, s_1, s_2 the standard deviations and $\lambda_1 = 1/n_1, \lambda_2 = 1/n_2$, n_1, n_2 are the number of observations.

TABLE 3
Comparison of k-NN Density Estimation Error of Condensation Algorithms (Higher CR)

Dataset	Multiscale Algorithm			Uniform Scale method [17]			SOM			Random sampling		
	CR	LLR	KLI	CR	LLR	KLI	CR	LLR	KLI	CR	LLR	KLI
	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD		Mean SD	Mean SD		Mean SD	Mean SD
Norm	20 0.001	0.38 0.001	0.08 0.00	20 0.002	0.43 0.002	0.10 0.001	20	0.40 0.001	0.09 0.00	20	0.49 0.09	0.11 0.01
					(74.16, 1.76)	(61.59, 1.78)		(46.9, 1.72)	(74.16, 1.76)		(4.05, 1.81)	(9.94, 1.81)
Iris	20 0.00	0.82 0.001	0.19 0.001	20 0.001	0.91 0.001	0.25 0.001	20	0.87 0.001	0.22 0.001	20	1.04 0.40	0.40 0.16
					(211, 1.72)	(140, 1.72)		(117, 1.72)	(9.90, 1.81)		(1.82, 1.81)	(4.35, 1.81)
Vowel	20 0.001	0.88 0.07	0.05 0.001	20 0.002	0.97 0.10	0.09 0.001	20	**0.90 0.001**	0.07 0.001	20	1.25 0.25	0.21 0.04
					(2.61, 1.74)	(93.8, 1.72)		**(0.93, 1.81)**	(46.90, 1.72)		(4.73, 1.81)	(13.2, 1.81)
Pima	20 0.001	0.50 0.05	8.8 0.32	20 0.002	0.62 0.09	10.0 0.81	20	0.59 0.002	9.1 0.10	20	0.81 0.25	14.03 4.1
					(3.86, 1.78)	(4.56, 1.76)		(5.96, 1.81)	(2.96, 1.81)		(4.16, 1.81)	(4.21, 1.81)
Cancer	20 0.001	0.68 0.05	9.1 0.4	20 0.002	0.81 0.07	10.4 0.70	20	0.77 0.01	9.8 0.01	20	0.92 0.22	11.9 2.09
					(5.01, 1.76)	(5.34, 1.74)		(5.85, 1.81)	(5.63, 1.81)		(3.52, 1.81)	(4.36, 1.81)
Monk	20 0.002	0.31 0.001	0.32 0.005	20 0.002	0.34 0.002	0.35 0.002	20	0.32 0.001	0.33 0.001	20	0.42 0.04	0.44 0.04
					(44.5, 1.78)	(18.47, 1.78)		(33.01, 1.81)	(6.40, 1.81)		(9.11, 1.81)	(9.87, 1.81)
Tnorm	20 0.000	0.22 0.001	0.80 0.005	10 0.001	0.29 0.005	1.04 0.02	10	0.25 0.00	0.88 0.01	10	0.35 0.08	1.21 0.17
					(45.53, 1.81)	(38.61, 1.81)		(70.35, 1.71)	(26.40, 1.81)		(5.40, 1.81)	(7.99, 1.81)
Rnorm	20 0.000	0.25 0.005	0.91 0.002	10 0.001	0.29 0.01	1.07 0.07	10	0.26 0.00	1.01 0.00	10	0.32 0.09	1.21 0.35
					(11.86, 1.78)	(7.57, 1.81)		(6.63, 1.81)	(32.52, 1.81)		(2.57, 1.81)	(2.84, 1.81)
Forest	5 0.001	0.54 0.005	0.91 0.002	5 0.002	0.62 0.005	1.71 0.007	5	0.57 0.002	1.04 0.005	5	1.72 0.25	4.91 1.17
					(37.5, 1.72)	(364, 1.81)		(18.4, 1.78)	(80.0, 1.76)		(15.6, 1.81)	(11.4, 1.81)
Sat.Img.	5 0.001	0.41 0.005	0.71 0.01	5 0.001	0.50 0.007	0.81 0.02	5	0..47 0.002	0.80 0.01	5	0.62 0.10	0.92 0.14
					(34.70, 1.76)	(14.83, 1.76)		(36.95, 1.78)	(21.10, 1.71)		(6.95, 1.81)	(4.96, 1.81)
Census	5 0.002	0.17 0.001	0.80 0.01	5 0.002	0.22 0.002	0.91 0.007	5	0.19 0.00	0.88 0.005	5	0.28 0.01	1.00 0.17
					(74.16, 1.76)	(27.98, 1.78)		(46.90, 1.81)	(21.95, 1.78)		(36.3, 1.81)	(3.89, 1.81)

not present those results. The error values were computed using (2) and (3) with same value of k as used for condensation. It may be noted that optimal choice of k is a function of the data size.

It is seen from the results (Tables 2 and 3) that our multiscale method achieves consistently better performance than Astrahan's method, random sampling, and SOM for both sets of condensation ratios. For each condensation ratio (two condensation ratios are considered), for each index of comparison (two indices are considered) of density estimation error, and for each data set (eleven data sets including three large data sets), the proposed method is found to provide better performance than each of the other three data condensation methodologies compared. Regarding statistical significance tests it can be seen from Tables 2 and 3 that, out of 132 comparisons, the proposed method is found to provide *significantly* better results in 127 comparisons. Only while comparing with SOM for the *Norm*, Vowel, and Rnorm data sets, the performance of the proposed method was found to be better, but not significantly. Experiments were performed for other values of the condensation ratio and similar performance was obtained.

For the purpose of comparison, the condensed sets obtained using different algorithms were also used for kernel density estimates. The kernel estimate is given by $\hat{\beta}_n(\mathbf{x}) = 1/n \sum_{j=1}^{n} K(\mathbf{x}, \mathbf{y}_j)$, where $\mathbf{y}_j$ are points belonging to the reduced set and $K(,)$ is the kernel function. We used a Gaussian kernel of the form

$$K(\mathbf{x}, \mathbf{y}_j) = \left[(h^2 2\pi)^{-d/2}\right] \exp\left\{-\frac{1}{2h^2}\delta(\mathbf{x}, \mathbf{y}_j)\right\},$$

where d is the dimension, h bandwidth, and $\delta(\mathbf{x}, \mathbf{y}_j)$ the Euclidean distance between $\mathbf{x}, \mathbf{y}_j$. The bandwidth h was chosen as

$$h = \sqrt{sup_{i=1,..,n}(inf_{j=1,..,n} d(\mathbf{y_i}, \mathbf{y_j}))},$$

where $\mathbf{y}_i, \mathbf{y}_j$ are points in the condensed set. The reason for selecting the above bandwidth can be explained in terms of minimal spanning trees [18]. The bandwidth satisfies both the conditions for consistent kernel density estimation. The error measures are presented in Tables 4 and 5 for the same two groups of condensed sets as considered in Tables 2 and 3, respectively. It is seen from Tables 4 and 5 that, when using kernel estimates, the proposed algorithms produces less error than all the related schemes for all data sets. Statistical significance tests are also presented for all the comparisons and, in 129 of 132 comparisons, the proposed method performs significantly better than the other three algorithms.

We also compared our algorithm with Fukunaga's nonparametric data condensation algorithm [19] only for the *Norm* data set. For a log-likelihood error of 0.5, the condensation ratio achieved by this method was 50 percent, while the corresponding figure was 23.4 percent for our method. On the *Norm* data set, while the CPU time required by the proposed algorithm was 8.10 seconds, the above mentioned algorithm required $2,123.05$ seconds.

In Fig. 3, we plot the points in the condensed set along with the discs covered by them at different condensation ratios for the proposed algorithm and for Astrahan's method. The objective is to demonstrate the multiresolution characteristics

TABLE 4
Comparison of Kernel (Gaussian) Density Estimation Error of Condensation Algorithms

Dataset	Multiscale Algorithm		Uniform Scale method [17]		SOM		Random sampling	
	LLR	KLI	LLR	KLI	LLR	KLI	LLR	KLI
	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD
Norm	1.04 0.07	0.14 0.03	1.15 0.09	0.17 0.03	1.10 0.07	**0.15 0.004**	1.29 0.25	0.23 0.09
			(3.05, 1.74)	(2.24, 1.74)	(1.92, 1.72)	**(1.04, 1.81)**	(3.05, 1.81)	(3.67, 1.78)
Iris	1.72 0.05	0.37 0.02	1.91 0.14	0.59 0.04	1.88 0.01	0.41 0.002	2.78 0.95	0.98 0.17
			(4.04, 1.78)	(15.56, 1.76)	(9.92, 1.81)	(6.29, 1.81)	(3.52, 1.81)	(11.27, 1.81)
Vowel	1.35 0.09	0.09 0.005	1.61 0.17	0.16 0.01	**1.38 0.002**	0.10 0.00	1.88 0.47	0.37 0.08
			(4.27, 1.76)	(19.8, 1.76)	**(1.05, 1.81)**	(6.32, 1.81)	(3.50, 1.81)	(11.05, 1.81)
Pima	1.07 0.08	17.2 0.81	1.27 0.11	19.9 2.2	1.18 0.01	19.1 0.88	1.91 0.90	23.2 8.9
			(4.65, 1.74)	(3.64, 1.78)	(4.31, 1.81)	(5.02, 1.72)	(2.94, 1.81)	(2.12, 1.81)
Cancer	1.34 0.16	16.8 1.4	1.57 0.20	18.8 0.91	1.51 0.09	19.1 0.47	1.78 0.55	23.3 8.80
			(2.84, 1.74)	(3.78, 1.78)	(2.92, 1.78)	(4.93, 1.79)	(2.43, 1.81)	(2.31, 1.81)
Monk	0.62 0.01	0.63 0.04	0.68 0.03	0.71 0.04	0.66 0.01	0.67 0.01	0.82 0.11	0.87 0.14
			(6.00, 1.78)	(4.47, 1.74)	(8.94, 1.74)	(3.08, 1.81)	(6.00, 1.81)	(5.21, 1.81)
Tnorm	0.42 0.01	1.64 0.05	0.56 0.05	1.92 0.11	0.45 0.00	1.78 0.001	0.57 0.10	1.97 0.44
			(8.68, 1.81)	(6.51, 1.74)	(9.49, 1.81)	(5.53, 1.81)	(4.72, 1.81)	(2.33, 1.81)
Rnorm	0.38 0.03	2.02 0.17	0.53 0.05	2.80 0.19	0.40 0.001	2.19 0.01	0.69 0.09	2.89 0.82
			(8.13, 1.76)	(6.51, 1.74)	(9.49, 1.81)	(5.53, 1.81)	(4.72, 1.81)	(2.33, 1.81)
Forest	0.80 0.007	2.69 0.01	1.95 0.01	4.4 0.53	1.38 0.00	3.10 0.01	3.70 1.43	7.0 2.50
			(325, 1.74)	(10.2, 1.81)	(366, 1.81)	(91, 1.72)	(6.55, 1.81)	(5.45, 1.81)
Sat.Img	0.75 0.005	1.09 0.02	0.88 0.01	1.28 0.09	0.82 0.005	1.22 0.00	0.98 0.10	1.72 0.22
			(36.77, 1.76)	(6.52, 1.81)	(31.3, 1.72)	(20.55, 1.81)	(7.26, 1.81)	(9.02, 1.81)
Census	0.25 0.00	1.46 0.04	0.29 0.01	1.59 0.09	0.27 0.005	1.52 0.005	0.37 0.10	1.82 0.40
			(12.6, 1.81)	(4.17, 1.78)	(12.6, 1.81)	(4.71, 1.81)	(3.79, 1.81)	(2.83, 1.81)

(Lower CR, same condensed set as Table 2.)

of the algorithm in contrast to a fixed resolution method. It is observed that our algorithm represents the original data in a multiresolution manner; the denser regions are more accurately represented compared to the sparser regions. The regions covered by the representative points are uniform for Astrahan's method [17]. It may be observed from the figure that multiscale representation is most effective in terms of error when the condensed set is sparse, i.e., the condensation ratio is low (Fig. 3a).

5.2 Classification: Forest Cover Data

As mentioned in Section 5, the data represents forest cover types of 30 m $\times$ 30 m cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS). There are 581,012 instances, with 54 attributes representing cartographic variables (hillshade, distance to hydrology, elevation, soil type, etc.), of which 10 are quantitative and 44 binary. The quantitative variables were scaled to the range $[0, 1]$. The task is to classify the observations into seven categories representing the forest cover types, namely, Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, Krummholz. About 80 percent of the observations belong to classes Spruce/Fir and Lodgepole Pine.

We have condensed the training set using different condensation algorithms including the proposed one. The different condensed sets obtained were then used to design a k-NN classifier (1-NN for LASM) and a multilayer perceptron (MLP) for classifying the test set. The goal was to provide an evidence that the performance of our multiresolution condensation algorithm does not depend on the final use of the condensed set. The following data reduction methods are compared:

1. **Random sampling with replacement to obtain a specific condensation ratio.** The condensed set is a representative of the underlying distribution, but, at low condensation ratios (say, 0.1 percent), it was found to have high variance (about 25 percent of the mean value of classification accuracy).
2. **Stratified sampling**. Instead of sampling uniformly over the entire population, subclasses of interest (*strata*) are identified and treated differently. For the given data, we considered *class stratification*, i.e., the number of samples selected from each class is proportional to the size of the class in the original set.
3. **Condensed nearest neighbor (CNN)** [7]. The condensation ratio is varied by changing the parameter k used for k-NN classification. The condensed set obtains a high concentration of points near the class boundaries and, hence, distorts the distribution. It may be mentioned that arbitrarily low-condensation ratios cannot be achieved using CNN.
4. **Local asymmetrically weighted similarity metric (LASM)** [9]. The condensed set is obtained by random sampling, but the metric used for nearest neighbor

TABLE 5
Comparison of Kernel (Gaussian) Density Estimation Error of Condensation Algorithms

Dataset	Multiscale Algorithm		Uniform Scale method [17]		SOM		Random sampling	
	LLR	KLI	LLR	KLI	LLR	KLI	LLR	KLI
	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD	Mean SD
Norm	0.35 0.001	0.07 0.00	1.40 0.001	0.09 0.001	0.37 0.001	0.08 0.001	0.47 0.05	0.10 0.01
			(117, 1.72)	(66, 1.81)	(47, 1.72)	(33.1, 1.81)	(7.95, 1.81)	(9.94, 1.81)
Iris	0.79 0.001	0.17 0.001	0.88 0.001	0.23 0.001	0.86 0.001	0.21 0.001	1.00 0.28	0.37 0.10
			(211, 1.72)	(140, 1.72)	(140, 1.72)	(93.8, 1.72)	(2.48, 1.81)	(6.63, 1.81)
Vowel	0.86 0.05	0.04 0.001	0.95 0.09	0.08 0.001	**0.88 0.001**	0.05 0.001	1.17 0.22	0.20 0.04
			(2.90, 1.74)	(93.8, 1.72)	**(1.32, 1.81)**	(23.45, 1.72)	(4.55, 1.81)	(13.26, 1.81)
Pima	0.47 0.04	8.20 0.28	0.60 0.07	9.10 0.54	0.56 0.001	8.8 0.04	0.80 0.17	14.00 4.10
			(5.34, 1.74)	(4.90, 1.74)	(7.46, 1.81)	(7.03, 1.81)	(6.27, 1.81)	(4.68, 1.81)
Cancer	0.67 0.04	8.70 0.35	0.79 0.05	9.80 0.76	0.74 0.005	9.50 0.01	0.90 0.19	11.5 2.01
			(6.21, 1.76)	(4.66, 1.74)	(5.75, 1.81)	(7.57, 1.81)	(3.92, 1.78)	(4.55, 1.81)
Monk	0.30 0.001	0.31 0.004	0.34 0.001	0.34 0.001	0.31 0.001	0.32 0.001	0.41 0.03	0.44 0.02
			(93.8, 1.72)	(24.1, 1.81)	(23.4, 1.72)	(8.04, 1.78)	(12.15, 1.81)	(21.14, 1.81)
Tnorm	0.21 0.001	0.78 0.004	0.28 0.004	0.99 0.01	0.23 0.00	0.86 0.005	0.34 0.05	1.19 0.10
			(56.3, 1.81)	(64.6, 1.78)	(66.3, 1.81)	(41.4, 1.76)	(8.62, 1.81)	(13.5, 1.81)
Rnorm	0.23 0.002	0.88 0.001	0.28 0.005	1.02 0.05	0.24 0.001	0.97 0.001	0.31 0.05	1.17 0.28
			(30.8, 1.78)	(9.28, 1.81)	(14.8, 1.78)	(211, 1.72)	(4.64, 1.81)	(3.43, 1.81)
Forest	0.53 0.004	0.90 0.002	0.61 0.004	1.70 0.005	0.55 0.001	0.98 0.004	1.70 0.17	4.90 1.00
			(46.9, 1.72)	(492, 1.78)	(16.08, 1.79)	(59.3, 1.74)	(22.8, 1.81)	(13.2, 1.81)
Sat.Img	0.40 0.004	0.70 0.005	0.47 0.007	0.80 0.01	0.45 0.001	0.77 0.005	0.59 0.05	0.90 0.10
			(28.8, 1.74)	(29.6, 1.74)	(40, 1.78)	(32, 1.72)	(12.5, 1.81)	(6.62, 1.81)
Census	0.16 0.001	0.78 0.01	0.22 0.001	0.91 0.005	0.17 0.00	0.87 0.004	0.27 0.01	0.98 0.11
			(140, 1.72)	(35, 1.76)	(33.1, 1.81)	(27.7, 1.78)	(36.3, 1.81)	(6.00, 1.81)

(Higher CR, same condensed set as Table 3.)

classification varies locally and is learned from the training set. The value of reinforcement rate used is $\alpha = 0.2$ and the punishment rate used is $\beta = 1.0$.

5. **Method of Astrahan** [17]. As explained in the last section, this is an uniform scale density-based method.
6. **Learning vector quantization** [14]: We have considered the LVQ3 version of the algorithm for comparison. Initial codebook vectors obtained using a self-organizing map are refined here using the LVQ3.

As in the case of density estimate experiments (Section 5.1), we have selected 90 percent of the data randomly as training set and the remaining data was used as test set. Such data splits were performed 11 times independently and the mean and standard deviation (SD) of the classification accuracy on test set and condensation ratios (CR) obtained for each such splits are presented. Statistical significance tests were also performed to test the inequality of means of the classification accuracy. As before, we present the computed value of the test statistic and the tabled value if the computed value is greater than the tabled value the means are significantly different. We also present the CPU times required by the condensation algorithms on a Digital Alpha 800MHz workstation. The figures shown here are the average values taken over 11 runs.

In Table 6, we compare the effect of each method on classification accuracy for condensation ratios of 0.1 percent and 5 percent. Note that the lowest condensation ratio that could be achieved for the Forest data using CNN is 3.1 percent, hence, comparison with CNN is presented only for the 5 percent case.

It can be seen from Table 6 that the proposed methodology achieves higher classification accuracy than the other methods and that this difference is statistically significant. For classification, the same value of k as used for condensation is considered, except for LASM, where 1-NN is used. For classification using MLP, the proposed method and LVQ performs similarly. Results for LASM are not presented for MLP since, if no specialized metric is used, LASM represents just a random subset. The performances of both random sampling and stratified sampling were found to be catastrophically poor. The uniform scale method of Astrahan performs poorer than the proposed method, LVQ and LASM.

5.3 Clustering: Satellite Image Data

The satellite image data contains observations of the Indian Remote Sensing (IRS) satellite for the city of Calcutta, India. As mentioned in Section 5, the data contains images of four spectral bands. We present in Fig. 4a, for convenience, the image for band 4. Here, the task is to segment the image into different land cover regions, using four features (spectral bands). The image mainly consists of six classes e.g., clear water (ponds, fisheries), turbid water (the river Ganges flowing through the city), concrete (buildings, roads, airport tarmacs), habitation (concrete structures but less in density), vegetation (crop, forest areas), and open

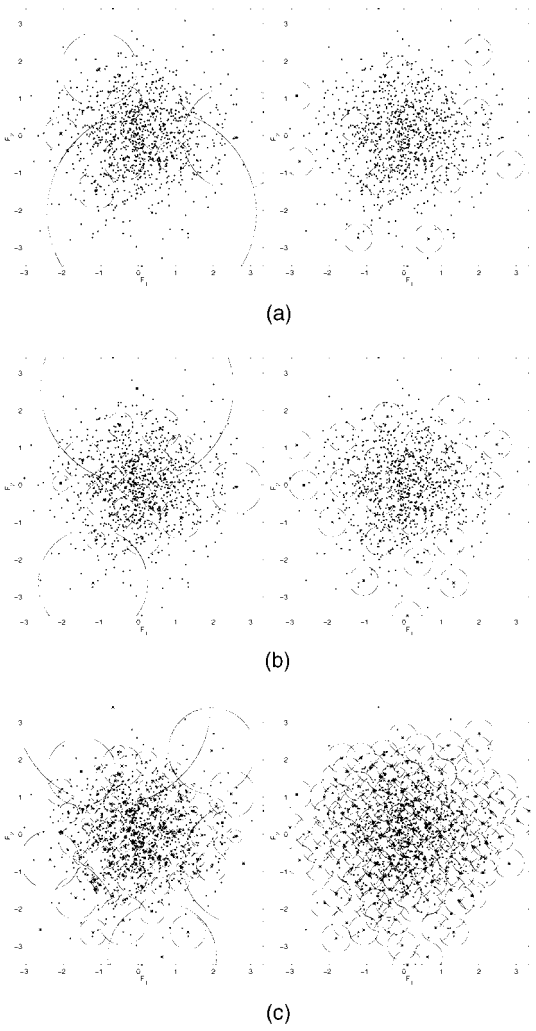

Fig. 3. Plot of the condensed points (of the *Norm* data) for the proposed algorithm and Astrahan's method, for different sizes of the condensed set. Bold dots represent a selected point and the discs represent the area of $F_1 - F_2$ plane covered by a selected point at their center Left: Mulitscale method. Right: Astrahan's method.

spaces (barren land, playgrounds). Fuzzy segmentation of the image is reported in detail in [27].

Using our methodology, we extract six prototype points from the entire data set. The remaining points are placed in the cluster of the prototype point to whose sphere (disc) of influence the particular point belongs. Thus, the condensation process implicitly generates a clustering (partition/segmentation) of the image data.

We compare the performance of our algorithm with two other related clustering methods, namely, k-means algorithm [26] and Astrahans density-based uniform scale method [17]. For the k-means algorithm, we have considered $k = 6$ since there are six classes and the best result (as evaluated by a cluster quality index) obtained out of 10 random initializations is presented. In Astrahan's method, six prototype points are obtained, the remaining pixels are then classified by minimum distance classification with these six points.

The results are presented in Figs. 4b, 4c, 4c, and 4d. Fig. 4d is seen to have more structural details compared to Figs. 4b and 4c. From the segmented image obtained using the proposed method, more number of landmarks known from ground truths can be detected by visual inspection. The segmentation results of the remote sensing images obtained above are also evaluated quantitatively using an index β.

Let n_i be the number of pixels in the ith $(i = 1, \ldots, c)$ region obtained by the segmentation method. Let X_{ij} be the vector (of size 4×1) of the gray values of the jth pixel $(j = 1, \ldots, n_i)$ for all the images in region i, and $\bar{X}_i$ the mean of n_i gray values of the ith region. Then, β is defined as [27]:

$$\beta = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n_i} (X_{ij} - \bar{X})^T (X_{ij} - \bar{X})}{\sum_{i=1}^{c} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^T (X_{ij} - \bar{X}_i)}, \tag{4}$$

where n is the size of the image and $\bar{X}$ is the mean gray value of the image. It may be noted that X_{ij}, $\bar{X}$, and $\bar{X}_i$ are all 4×1 vectors.

Note that the above measure is nothing but the ratio of the total variation and within-class variation and is widely used for feature selection and cluster analysis [27]. For a given image and c (number of clusters) value, the higher the homogeneity within the segmented regions, the higher the β value. The proposed method has the highest β as can be seen in Table 7.

5.4 Rule Generation: Census Data

The original source for this data set is the IPUMS project. As mentioned in Section 5, we use $320,000$ samples. The data contains 133 attributes, mostly categorical (integer valued). A study commonly performed on census data is to identify contrasting groups of populations and study their relations. For this data, we investigate two groups of population, namely, those who have undergone/not undergone "higher education," measured in terms of number of years in college. It is interesting and useful to generate logical rules depending on the other available attributes which classify these groups. We consider the attribute educational record, "*edrec*," and investigate two sets of population, one having more than $4\frac{1}{2}$ years of college education, and the other below that. The task is to extract logical inference rules for the sets.

As a similarity measure between two samples, we use the *Value Difference Metric* (VDM) [8]. Using the VDM, the distance between two values x and y of a single attribute a is defined as

$$vdm_a(x, y) = \sum_{a=1}^{C} \left(\frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right)^2, \tag{5}$$

where $N_{a,x}$ is the number of times attribute a had value x; $N_{a,x,c}$ is the number of times attribute a had value x, and the output class was c; and C is the number of output classes (two in our case). Using this distance measure, two values of an attribute are considered to be closer if they have more similar classification, regardless of the magnitude of the values. Using the value difference metric, the distance between two points having m independent attributes is defined as

$$\mathrm{VDM}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{a=1}^{m} vdm_a^2(\mathbf{x}_a, \mathbf{y}_a)}. \tag{6}$$

We use the popular C4.5 [32] program to generate logical rules from the condensed data sets. We restrict the rule sizes

TABLE 6
Classification Performances for Forest Covertype Data

Condensation Algorithm	Condensation Ratio (%)		Classification Accuracy (%) using k-NN			Classification Accuracy (%) using MLP			CPU time (hrs)
	Mean	SD	Mean	SD	(test stat.)	Mean	SD	(test stat.)	
Proposed method	0.1	0.004	83.10	1.90		70.01	0.90		4.29
LVQ3	0.1	-	75.01	1.01	(12.50, 1.76)	68.08	0.80	(3.33, 1.72)	2.02
LASM	0.1	-	74.50	2.52	(9.08, 1.72) (1-NN)	-	-		5.90
Astrahan	0.1	0.004	66.90	2.10	(18.97, 1.72)	59.80	0.53	(32.81, 1.73)	4.10
Stratified sampling	0.1	-	44.20	5.9	(20.81, 1.81)	36.10	5.95	(18.75, 1.81)	-
Random sampling	0.1	-	37.70	10.04	(14.73, 1.81)	29.80	8.2	(16.16, 1.81)	-
Proposed method	5.0	0.01	97.00	1.81		80.02	1.40		4.52
LVQ3	5.0	-	88.01	1.04	(14.34, 1.76)	74.00	0.92	(11.99, 1.73)	4.05
LASM	5.0	-	87.55	2.50	(10.17, 1.73) (1-NN)	-	-		7.11
Astrahan	5.0	0.01	82.09	2.53	(16.05, 1.73)	66.00	1.4	(23.48, 1.71)	4.40
CNN	5.05	1.01	81.17	3.80	(2.64, 1.73)	75.02	4.1	(1.52, 1.78)	5.51
Stratified sampling	5.0	-	55.20	7.1	(18.92, 1.81)	40.10	7.01	(18.52, 1.81)	-
Random sampling	5.0	-	44.70	8.02	(21.09, 1.81)	35.80	8.8	(16.40, 1.81)	-

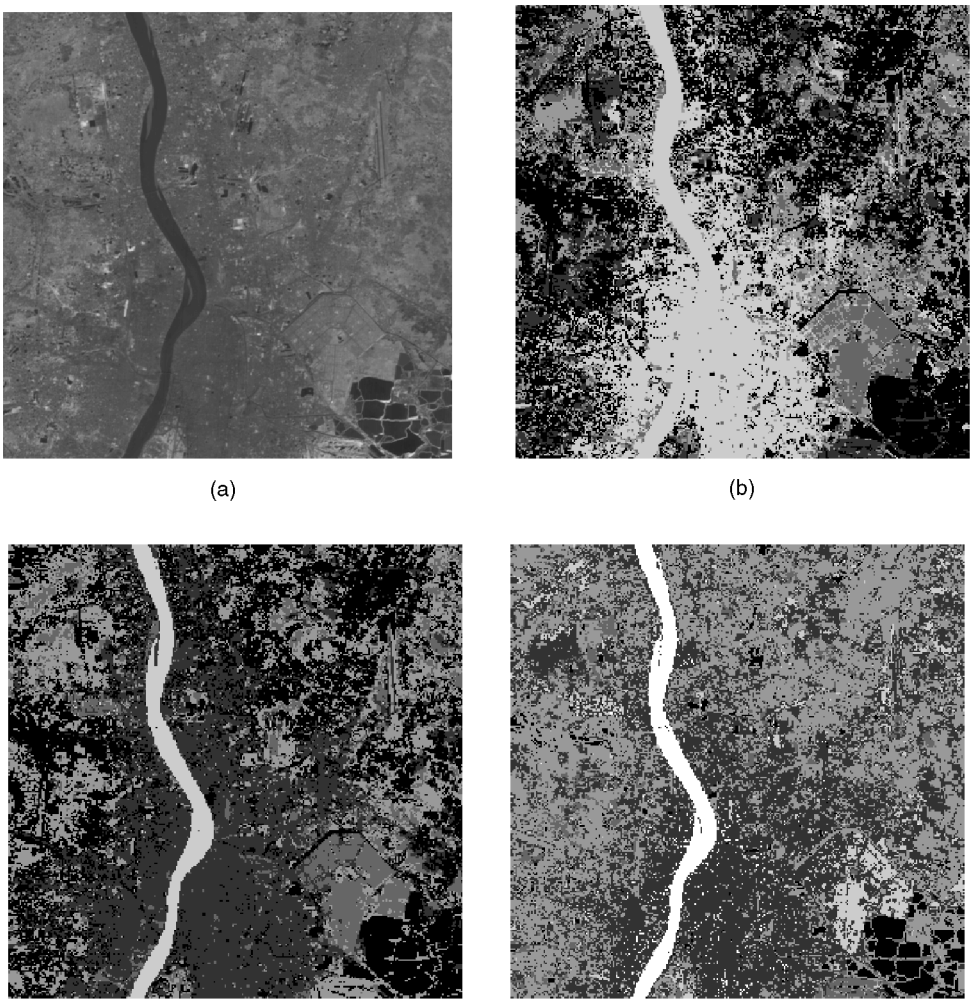

Fig. 4. IRS images of Calcutta. (a) Original Band 4 image and segmented images using (b) k-means algorithm, (c) Astrahan's method, and (d) proposed multiscale algorithm.

TABLE 7
β Value and CPU Time of Different Clustering Methods

Method	k-means	Astrahan's	Proposed
β	5.30	7.02	9.88
CPU time (hrs)	0.11	0.71	0.75

upto conjunction of three variables only. As before, 90 percent of the data was selected as training set and the rules are evaluated on the remaining data. Eleven such splits were obtained and the means and standard deviations (SD) are presented.

For the purpose of comparison with our method, the C4.5 program was also run on condensed sets obtained using: Random sampling, Stratified sampling, Density-based Uniform Scale method of Astrahan [17], and Condensed Nearest Neighbor. Following quantities are computed in Table 8:

1. Condensation Ratio (CR),
2. Number of Rules Generated,
3. Accuracy of Classification on test set (we also present statistical tests of significance for comparing the other methods with the proposed method),
4. Percentage of uncovered samples, and
5. CPU time.

The comparison is performed for a constant condensation ratio of 0.1 percent. However, for CNN a CR of only 2.2 percent could be achieved by varying k. The classification accuracy of the proposed method is higher than random sampling, stratified sampling, and CNN; it is also significantly higher than Astrahan's method. It is also observed that the uncovered region is minimum for the rules generated from the subset obtained by the proposed algorithm. The rule base size is far smaller than random, statistical sampling, and Astrahan's method. Therefore, the rules generated from the condensed set are compact, yet have high accuracy and cover as compared to other sets.

5.5 Experiments on Scalability

The scaling property of the condensation algorithm is also studied in a part of the experiment. For this, we examine the *sample complexity* of the algorithm, i.e., the size of condensed set required to achieve an accuracy level (measured as error in density estimate). In Fig. 5, the log-likelihood error is plotted against the cardinality of the condensed set (as a fraction of the original set), for three typical data sets, namely, *Norm* (of known distribution), Vowel (highly overlapping), and Wisconsin (large dimension). The solid curve is for the proposed methodology, while the dotted one is for random sampling. It can be seen that the proposed methodology is superior to random sampling.

5.6 Experiments on Choice of k

In Section 4, we have described the role of k in the proposed algorithm. As k increases, the size of condensed set reduces and vice versa. Here, we provide some experimental results in support of the discussion. The effect of varying parameter k on the condensation ratio (CR) is shown in Fig. 6, for the three aforesaid data sets (Section 5.6). It can be observed that, for values of k in the range $\approx$ 7-20, the curves attain low CR values and are close to each other for all the three data sets. For the Vowel data, a CR value of 3.4 percent was obtained at $k = 31$. It may be noted that the curve for the *Norm* (smallest) data set is shifted to the left compared to the other two curves.

6 Conclusions and Discussion

This paper presented an algorithm for nonparametric data condensation. The method follows the basic principles of nonparametric data reduction present in literature, but the sample pruning step is done in a multiresolution manner rather than with uniform resolution. It is based on the density underlying the data. The proposed approach was found to have superior performance as compared to some existing data reduction schemes in terms of error in density estimate both for small and large data sets having dimension ranging from 2-133. Classification, clustering, and rule generation performances using the condensation algorithm were studied for three large data sets. The algorithm does not require the difficult choice of radii d_1 and d_2, which are critical for Astrahan's method, only the choice of parameter k is necessary. Choice of k is guided by the size of the original data set and the accuracy/condensation ratio desired. The parameter k also provides a parameterization of the concept of scale in data condensation and the scales induced follow the natural characteristics of the data and, hence, efficient.

As far as the computational complexity is concerned, the algorithm can be considered to have three computational

TABLE 8
Rule Generation Performance for the Census Data

Condensation method	CR(%)		# of Rules (rounded to integer)		Classification accuracy (%)			Uncovered samples (%) (%)		CPU time (hrs)
	Mean	SD	Mean	SD	Mean	SD	(Test Stat.)	Mean	SD	
Random sampling	0.1	-	448	88	32.1	8.8	(8.43, 1.81)	40.01	5.5	-
Stratified sampling	0.1	-	305	45	38.8	5.5	(9.71, 1.78)	37.0	5.5	-
CNN	2.2	0.050	270	53	32.0	4.1	(17.55, 1.78)	55.0	4.1	2.80
Astrahan [17]	0.1	0.004	245	50	48.8	4.0	(4.89, 1.78)	25.0	3.1	4.22
Proposed	0.1	0.004	178	30	55.1	1.5		20.2	1.80	4.10

Figures in parentheses indicate the computed value of test statistic and tabled value, respectively.

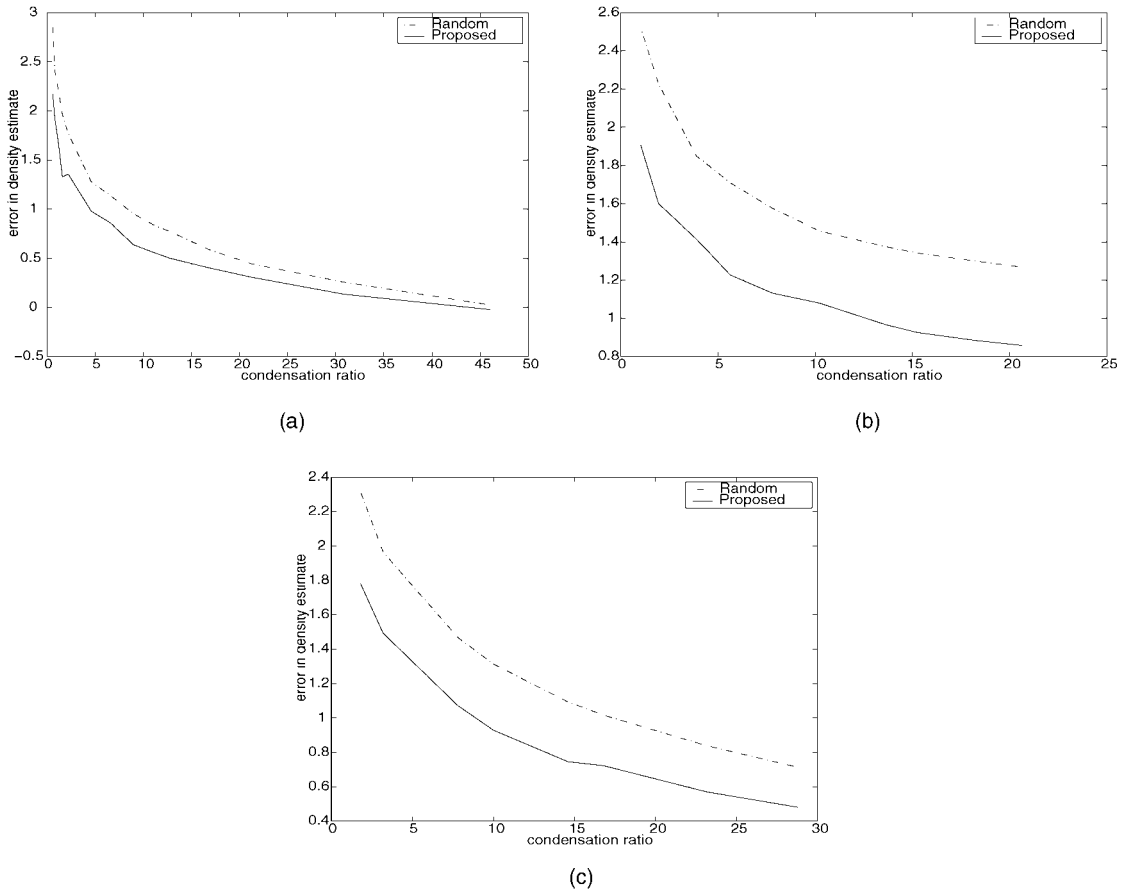

Fig. 5. Variation in error in density estimate (log-likelihood measure) with the size of the Condensed Set (expressed as percentage of the original set) with the corresponding for (a) the *Norm* data, (b) Vowel data, and (c) Wisconsin Cancer data.

steps. In the first step, for each point in the original set, the distance of the kth nearest neighbor is computed. In the second step, the point having the minimum value of the distance is selected and in the third step, all points lying within a radius of $2r_{k,\mathbf{x}_j}$ of a selected point is removed. It is observed that the second and third steps increase in speed since the size of the original set decreases progressively (the rate is dependent on k and the data distribution). The first step is the most time consuming one and it requires ($\mathcal{O}(kN^2)$), where N is the number of data points. A way of reducing the time complexity of nearest-neighbor calculation is to use approximate nearest neighbor (ANN) computations using specialized data structures like k-d trees [33]. Probabilistic nearest-neighbor search methods have also been suggested [34], having expected $\mathcal{O}(1)$ time complexity, and $\mathcal{O}(N)$ storage complexity.

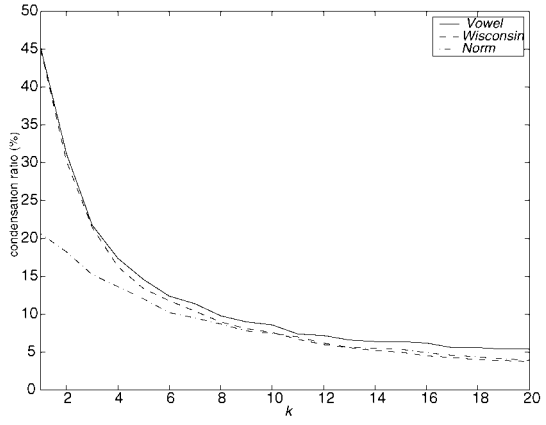

Fig. 6. Variation of condensation ratio CR (%) with k.

The guiding principle of our algorithm is to minimize the error in terms of density estimate rather than the classification score. The justification is to obtain a generic representative condensed set independent of the task performed with it later. In many data mining applications, the final task is not always known beforehand or there may be multiple tasks to be performed. In the above circumstances, such a condensed representation is more useful. We have performed experiments to show that a condensed set obtained by our method performs well for diverse data mining tasks such as classification, clustering, and rule generation.

REFERENCES

[1] U. Fayyad and R. Uthurusamy, "Data Mining and Knowledge Discovery in Databases," *Comm. ACM,* vol. 39, no. 11, pp. 24-27, Nov. 1996.

[2] F. Provost and V. Kolluri, "A survey of Methods for Scaling Up Inductive Algorithms," *Data Mining and Knowledge Discovery,* vol. 2, pp. 131-169, 1999.

[3] J. Catlett, "Megainduction: Machine Learning on Very Large Databases," PhD thesis, Dept. of Computer Science, Univ. of Sydney, Australia, 1991.

[4] D.D. Lewis and J. Catlett, "Heterogeneous Uncertainty Sampling for Supervised Learning," *Machine Learning: Proc. 11th Int'l Conf.*, pp. 148-156, 1994.
[5] N. Roy and A. McCallum, "Towards Optimal Active Learning through Sampling Estimation of Error Reduction," *Proc. 18th Int'l Conf. Machine Learning (ICML-2001),* 2001.
[6] B.V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Patterns Classification Techniques.* Los Alamitos, Calif.: IEEE CS Press, 1991.
[7] P.E. Hart, "The Condensed Nearest Neighbor Rule," *IEEE Trans. Information Theory,* vol. 14, pp. 515-516, 1968.
[8] D.R. Wilson and T.R. Martinez, "Reduction Techniques for Instance-Based Learning Algorithms," *Machine Learning,* vol. 38, no. 3, pp. 257-286, 2000.
[9] F. Ricci, P. Avesani, "Data Compression and Local Metrics for Nearest Neighbor Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, pp. 380-384, 1999.
[10] M. Plutowski and H. White, "Selecting Concise Training Sets from Clean Data," *IEEE Trans. Neural Networks,* vol. 4, no. 2, pp. 305-318, 1993.
[11] D.L. Reilly, L.N. Cooper, and C. Elbaum, "A Neural Model for Category Learning," *Biological Cybernetics,* vol. 45, pp. 35-41, 1982.
[12] J. Platt, "A Resource-Allocating Network for Function Interpolation," *Neural Computation,* vol. 3, pp. 213-255, 1991.
[13] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine,* vol. 1, pp. 4-29, 1984.
[14] T. Kohonen, "The Self-Organizing Map," *Proc. IEEE,* vol. 78, pp. 1464-1480, 1990.
[15] L. Xu, A. Krzyzak, and E. Oja, "Rival Penalised Competitive Learning for Cluster Analysis, RBF Net and Curve Detection," *IEEE Trans. Neural Networks* vol. 4, pp. 636-649, 1993.
[16] J. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD '96),* pp. 226-231, 1996.
[17] M.M. Astrahan, "Speech Analysis by Clustering, or the Hyperphoneme Method," Stanford A. I. Project Memo, Stanford Univ., Calif., 1970.
[18] D. Chaudhuri, C.A. Murthy, and B.B. Chaudhuri, "Finding a Subset of Representative Points in a Dataset," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 24, pp. 1416-1424, 1994.
[19] K. Fukunaga and J.M. Mantock, "Nonparametric Data Reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, pp. 115-118, 1984.
[20] K. Deng and A.W. Moore, "Multiresolution Instance-Based Learning," *Proc. Int'l Joint Conf. Artificial Intelligence,* 1995.
[21] A.W. Moore, J. Schneider, and K. Deng, "Efficient Locally Weighted Polynomial Regression Predictions," *Machine Learning: Proc. 14th Int'l Conf.,* pp. 236-244, 1997.
[22] Y. Leung, J.-S. Zhang, and Z.-B. Xu, "Clustering by Scale-Space Filtering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, pp. 1396-1410, 2000.
[23] S. Chakravarthy and J. Ghosh, "Scale-Based Clustering Using Radial Basis Function Network," *IEEE Trans. Neural Networks,* vol. 7, pp. 1250-1261, 1996.
[24] Y-F. Wong and E. C. Posner, "A New Clustering Algorithm Applicable to Polarimetric and SAR Images," *IEEE Trans. Geoscience and Remote Sensing,* vol. 31, no. 3, pp. 634-644, Apr. 1993.
[25] D.O. Loftsgaarden and C.P. Quesenberry, "A Nonparametric Estimate of a Multivariate Density Function," *Annals of Math. Statistics,* vol. 36, pp. 1049-1051, 1965.
[26] K. Fukunaga, *Introduction to Statistical Pattern Recognition and Machine Learning*. New York: Academic, 1972.
[27] S.K. Pal, A. Ghosh, and B. Uma Shankar, "Segmentation of Remotely Sensed Images with Fuzzy Thresholding, and Quantitative Evaluation," *Int'l J. Remote Sensing,* vol. 21, no. 11, pp. 2269-2300, 2000.
[28] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees.* Wadsworth Inc., 1984.
[29] S.K. Pal and S. Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing Paradigm.* New York: John Wiley, 1999.
[30] E.L. Lehmann, *Testing of Statistical Hypotheses.* New York: John Wiley, 1976.
[31] A. Aspin, "Tables for Use in Comparisons Whose Accuracy Involves two Variances," *Biometrika,* vol. 36, pp. 245-271, 1949.
[32] J.R. Quinlan, *C4.5, Programs for Machine Learning.* Calif.: Morgan Kaufman, 1993.
[33] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching," *J. ACM,* vol. 45, pp. 891-923, 1998.
[34] A. Faragó, T. Linder, and G. Lugosi, "Nearest Neighbor Search and Classification in $O(1)$ Time," *Problems of Control and Information Theory,* vol. 20, no. 6, pp. 383-395, 1991.

Pabitra Mitra received the BTech degree in electrical engineering from the Indian Institute of Technology, Kharagpur, in 1996. He worked as a scientist with the Institute for Robotics and Intelligent Systems, India. Currently, he is a senior research fellow at the Indian Statistical Institute, Calcutta. His research interests are in the area of data mining and knowledge discovery, pattern recognition, learning theory, and soft computing. He is a student member of the IEEE.

C.A. Murthy received the BStat (Hons), MStat, and PhD degrees from the Indian Statistical Institute (ISI). He visited Michigan State University, East Lansing, in 1991-92, and the Pennsylvania State University, University Park, in 1996-97. He is a professor in the Machine Intelligence Unit of ISI. His fields of interest include pattern recognition, image processing, machine learning, neural networks, fractals, genetic algorithms, wavelets, and data mining. He received the best paper award in 1996 in computer science from the Institute of Engineers, India. He is a fellow of the National Academy of Engineering, India.

Sankar K. Pal (M '81-SM '84-F '93) (04230363) received the MTech and PhD degrees in radio physics and electronics in 1974 and 1979, respectively, from the University of Calcutta. In 1982, he received another PhD degree in electrical engineering along with a DIC from Imperial College, University of London. He is a professor and distinguished scientist at the Indian Statistical Institute, Calcutta. He is also the founding head of Machine Intelligence Unit. He worked at the University of California, Berkeley and the University of Maryland, College Park during 1986-87 as a Fulbright Postdoctoral Visiting Fellow, at the NASA Johnson Space Center, Houston, Texas during 1990-92, and in 1994 as a guest investigator under the NRC-NASA Senior Research Associateship program; and at the Hong Kong Polytechnic University, Hong Kong, in 1999 as a visiting professor. He served as a distinguished visitor of the IEEE Computer Society (USA) for the Asia-Pacific Region during 1997-99. He is a fellow of the IEEE, Third World Academy of Sciences, Italy, and all the four National Academies for Science/Engineering in India. His research interests include pattern recognition, image processing, data mining, soft computing, neural nets, genetic algorithms, and fuzzy systems. He is a coauthor/coeditor of eight books including *Fuzzy Mathematical Approach to Pattern Recognition*, John Wiley (Halsted), New York, 1986, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*, John Wiley, New York 1999, and has published more than 300 research publications. He has received the 1990 S.S. Bhatnagar Prize (which is the most coveted award for a scientist in India), 1993 Jawaharlal Nehru Fellowship, 1993 Vikram Sarabhai Research Award, 1993 NASA Tech Brief Award, 1994 IEEE Transactions on Neural Networks Outstanding Paper Award, 1995 NASA Patent Application Award, 1997 IETE-Ram Lal Wadhwa Gold Medal, 1998 Om Bhasin Foundation Award, 1999 G.D. Birla Award for Scientific Research, the 2000 Khwarizmi International Award (1st winner) from the Islamic Republic of Iran, the 2001 Syed Hussain Zaheer Medal from Indian National Science Academy, and the 2001 FICCI award. Professor Pal has been an associate editor for *IEEE Transactions on Neural Networks* (1994-98), *Pattern Recognition Letters*, the *International Journal Pattern Recognition and Artificial Intelligence*, *Neurocomputing, Applied Intelligence*, *Information Sciences, Fuzzy Sets and Systems*, and *Fundamenta Informaticae*. He is a member of the executive advisory editorial board, *IEEE Transactions Fuzzy Systems*, *International Journal on Image and Graphics*, and *International Journal of Approximate Reasoning*, and a guest editor of many journals including *IEEE Computer*.