An ad rem unsupervised classification review

Barton, Alan

https://doi.org/10.4224/40003384

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

ERB-1143

NRC·CNRC

An ad rem Unsupervised Classification Review

Barton, A.
December 2006

Canada

NRC 48807

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC·CNRC

An ad rem Unsupervised Classification Review

Barton, A.
December 2006

Canada

An ad rem Unsupervised Classification Review*

Alan J. Barton†
National Research Council Canada
Institute for Information Technology
M50, 1200 Montreal Rd.
Ottawa, ON K1A 0R6
alan.barton@nrc-cnrc.gc.ca

ABSTRACT

A brief survey of some unsupervised learning and clustering algorithms is performed based on a classical pattern recognition book. Other unsupervised approaches are also briefly introduced in order to broaden the content of the survey of such a large body of possible approaches. An example from paleontology is used to motivate the unsupervised learning problem, while examples from proteomics, geophysical prospecting and digital remote sensing are very briefly mentioned. In addition, an unsupervised learning procedure (the Isodata clustering algorithm) was implemented and results reported.

Keywords

pattern recognition, unsupervised learning, clustering

1. INTRODUCTION

The real world consists of objects that have properties associated with them; some of which may be measured in natural or human-made ways. For example, natural phenomenon occur such that molecules may be trapped within the confines of other molecular substances. In particular, we know that dinosaurs may have existed on the planet Earth due to the fact that we have found the fossilized remains of their bones (and other indications such as footprints in stone), not by direct observation of a dinosaur (although distant relatives such as birds, turtles and crocodiles do exist, so one might argue that dinosaurs still do exist) but rather through the interpretation of the discovered remains and their proximity relationships with nearby artifacts. How did humans obtain the knowledge that dinosaurs existed in the past? Certainly, the first human to discover such fossils did not have a teacher that told them that it was a fossil, but, in fact, it was an unsupervised process in which a generalization was made based on other facts. In particular, the fossils looked like bones, so perhaps they were bones, they were large, so perhaps they were from a large animal of some kind, and they were surmised to be very old. These facts (and certainly others, since the author is not a paleontologist, such as Charles Darwin was) enabled the generalized description of the potential existence of an animal. It may have been the case the the first guess was that of a mythical creature such as a large and powerful serpent or other reptile, with magical or spiritual qualities otherwise known as a dragon, or perhaps a first guess could have been that of giants with a single eye in the middle of their forehead and a foul disposition otherwise known as Cyclopes. But today, the current best guess is that of a dinosaur.

So even humans, given a set of facts and no one to teach them what those facts represent, can be misled into pseudo-concepts (a flat world becomes a round world... our home, Earth, changes from the center of the universe to merely one planet in the universe... very small particles (atoms) are theorized... and then subatomic particles... etc.). This, then, is the goal; to learn something new, in an unsupervised (rather than supervised e.g. [3]) sense but with the property that the learned concepts are as close to representing the real world observations as possible (Does this sound reminiscent of a portion of the definition of science?). As such, a survey of the concepts within a classical pattern recognition book[6] is reported.

2. UNSUPERVISED LEARNING

A data set may be described as consisting of attributes (columns) and samples (rows). Such a data set may have labels associated with each sample giving an indication of membership to a particular class. The usual membership function would be a binary relation with the property that a sample does or does not belong to that particular class. Of course, such a membership function may be generalized (as has been done) but regardless of the particular membership function, the point is that one was given the class of each training sample. As was suggested in the introduction, if one has never seen a particular problem before, then one would want to "learn" the classes contained within the data (assuming that no additional information other than the data set itself was available and that each sample should logically only have membership to one class – not a necessary restriction, but one that is usually made). This process is called unsupervised learning.

2.1 Semi-supervised Learning

Supervised learning is such that all of the training samples have labels (i.e. associated classes), while unsupervised learning is on the other extreme of the spectrum; namely, that not one training sample has a label. Certainly, if such a spectrum exists, then perhaps one might have a situation where some of the training samples have labels, and some do not. This latter situation is call semi-supervised learning[5] and usually consists of a few labeled samples that were costly to obtain and a larger set of unlabeled data that was

*Report for the course COMP5107 entitled *Statistical And Syntactic Pattern Recognition* at Carleton University

†Master of Computer Science (in progress)

much less costly to obtain. For example, the latter data may aid the more accurate estimate of the distributions of the attributes and in that sense (and others) is useful to have for the learning process.

2.2 Motivation

There are three main reasons[6] for pursuing unsupervised learning, which are ordered such that less and less supervision is given:

- Collecting and labeling many samples may be costly. Perhaps learning a coarse classifier on a small, labeled set of samples may be performed and then fine-tuned by using a large set of unlabeled samples. This seems to now be known by the name of semi-supervised learning[5].

- Once a classifier has been placed into a production environment, the new samples that are given to the classifier may drift with time. If such a classifier were able to track such temporal changes in an unsupervised manner, then perhaps improved classification would result.

- During the early stages of Exploratory Data Analysis (EDA), now more commonly known as Data Mining, one may be given a data set for which no information is known. It then becomes of interest to learn if an inherent nature or structure exists within the data. For if such consistencies (e.g. abnormalities, oddities, specific events, groups, etc) exist, then perhaps the design of a classifier for that data would need to be significantly re-thought.

2.3 Assumptions

A first approximation at tackling the unsupervised learning problem is to assume a functional form for the underlying probability density and then learn the value of an unknown parameter vector. This approach has some known problems (e.g. the functional form of the probability density may not be known, more than one density may statistically fit the data very well, etc.) and so it may be reformulated as one of partitioning the data into subgroups (clusters). Such a partitioning is known as clustering.

It is also known that the concepts that an investigator is pursuing within the data may not support a crisp partitioning and that a fuzzy partitioning may be more appropriate. For example, in the single class problem of known cave measurements and measurements that are not known to be cave or non-cave[10], one goal would be to find the degree of membership of an unknown sample measurement with that of the known (labeled) cave measurement. One cannot simply perform a crisp partitioning of the data, due to the geophysical properties of the concept of cave. That is, caves do not simply exist and then not exist. Sometimes there are also small spaces that are within the ground (or simply loosely compacted ground) that tend to resemble some of the properties of caves, but not all, for which, depending on the application, may not be of practical interest. Another example of unsupervised learning is within the context of analysing proteomics data for which no classes exist[4].

3. UNSUPERVISED APPROACHES

There are many approaches that attempt to address the problem that is inherent within an unsupervised setting. Only a small subset of the possibilities are reported from the literature[6].

3.1 Form of Model Known

If a systematic progression from simple to more complex is used then the simpler approach (meaning more assumptions are used, and that the problem has been restricted from the most general incarnation) starts by assuming that the model is known, but the parameters are not. In particular, the following assumptions are made:

1. there are a known number of classes (call that number, c) to which all samples (X_i) belong

2. before any measurements are taken, the probability of membership within a particular class is known. That is, the *a priori* probabilities $P(\omega_j)$ for each class are known for $j = 1, 2, \cdots, c$. (Also called *mixing parameters*)

3. the models are known. That is, the class-conditional probability densities $p(X|\omega_j, \theta_j)$ are known for $j = 1, 2, \cdots, c$. (Also called a *component density*)

4. the model parameters are not known. That is, the c vectors $\theta_1, \theta_2, \cdots, \theta_c$ are not known.

This, then, leads to the fact that the probability density function for the samples (a particular sample is X) is given by: $p(X|\theta_1, \theta_2, \cdots, \theta_c) = \sum_{j=1}^{c} p(X|\omega_j, \theta_j) \cdot P(\omega_j)$, which is known as a *mixture density* because of the obvious fact that the density is composed of a set, or mixture, of component densities. The basic goal is to estimate the unknown parameter vector $\theta = \theta_1, \theta_2, \cdots, \theta_c$, so that the mixture may be decomposed into its component densities; in effect solving the original statement of the problem in this context (i.e. when the form of the model is known and the parameters are not). But can it be done? More specifically, if more than one value of θ leads to the same value of the observed value $p(X|\theta)$, then there clearly exists an ambiguity. More concretely, a density $p(X|\theta)$ is said to be *identifiable* if $\theta \neq \theta'$ implies that there exists an X such that $p(X|\theta) \neq p(X|\theta')$ In the study of unsupervised learning, the restriction to *identifiable* mixtures clearly simplifies the task. Luckily, most mixtures that are commonly encountered[6] are *identifiable*, with the exception that discrete mixtures tend to be less co-operative. A simple example of a non-identifiable mixture, is a special case when normal densities are considered, with $P(\omega_1) = P(\omega_2)$ and $p(X|\theta) = \frac{P(\omega_1)}{\sqrt{2\pi}} \cdot e^{\frac{-1}{2} \cdot (X - \theta_1)} + \frac{P(\omega_2)}{\sqrt{2\pi}} \cdot e^{\frac{-1}{2} \cdot (X - \theta_2)}$ because θ_1 and θ_2 may be interchanged without affecting the value of $p(X|\theta)$.

3.1.1 General Maximum Likelihood Estimates

If given a set of unlabeled samples $\mathcal{X} = \{X_1, X_2, \cdots, X_n\}$ that were drawn independently from the mixture density previously defined, then the likelihood of actually drawing the particular observed values in our set is, by definition, the joint density: $p(\mathcal{X}|\theta) = \prod_{k=1}^{n} p(X_k|\theta)$. The value of θ may be estimated in various ways, and, in particular, the maximum likelihood estimate ($\hat{\theta}$) is the value of θ that maximizes $p(\mathcal{X}|\theta)$. Some general necessary conditions for $\hat{\theta}$ may then

be derived from these facts (not described here). In addition, the results may be generalized to include the *a priori* probabilities ($P(\omega_i)$) among those things that are unknown. That is, we would then have a maximum likelihood estimate for both θ (called $\hat{\theta}$) and for $P(\omega_i)$ (called $\hat{P}(\omega_i)$).

3.1.2 Normal Maximum Likelihood Estimates

The general maximum likelihood result may be applied to the case when the form of the model is the multivariate normal density. That is, $p(X|\omega_i, \theta_i) \sim N(\mu_i, \Sigma_i)$. There are 4 parameters that may be either known or unknown (μ_i, Σ_i, $P(\omega_i)$, and c). This leads to quite a few different cases (in fact, $2^4 = 16$ cases) of which, only some will be explicated:

1. Unknown: μ_i. Known: Σ_i, $P(\omega_i)$, and c.
   In this case the maximum likelihood estimate for μ_i is: $\hat{\mu}_i = \frac{\sum_{k=1}^{n} P(\omega_i|X_k,\hat{\mu})\cdot X_k}{\sum_{k=1}^{n} P(\omega_i|X_k,\hat{\mu})}$, which, intuitively, is a weighted average of the samples. Unfortunately, this equation does not give $\hat{\mu}_i$ explicitly. After some algebra, an iterative formula for $\hat{\mu}_i$ may be obtained, which is basically a local optimization procedure for maximizing the log-likelihood function (not shown here). As such, no global solution would be guaranteed to be obtained. For example, given a two-component normal mixture, with roughly equal a priori probabilities, and given that a sample data set could be generated from the mixture, two possible solutions would be obtained that would both be approximately correct, based on the particular data set generated.

2. Unknown: μ_i, Σ_i, and $P(\omega_i)$. Known: c.
   In this case, the maximum likelihood principle yields useless singular solutions. For example, the likelihood may be made arbitrarily large, so the maximum becomes unique. However, empirically, the maximum likelihood principle yields meaningful solutions. The local-maximum-likelihood estimates are:
   $\hat{P}(\omega_i) = \frac{1}{n} \cdot \sum_{k=1}^{n} \hat{P}\left(\omega_i|X_k,\hat{\theta}\right)$
   $\hat{\mu}_i = \frac{\sum_{k=1}^{n} \hat{P}\left(\omega_i|X_k,\hat{\theta}\right)\cdot X_k}{\sum_{k=1}^{n} \hat{P}\left(\omega_i|X_k,\hat{\theta}\right)}$
   $\hat{\Sigma}_i = \frac{\sum_{k=1}^{n} \hat{P}\left(\omega_i|X_k,\hat{\theta}\right)\cdot(X_k-\hat{\mu}_i)(X_k-\hat{\mu}_i)^T}{\sum_{k=1}^{n} \hat{P}\left(\omega_i|X_k,\hat{\theta}\right)}$
   with $\hat{P}\left(\omega_i|X_k,\hat{\theta}\right)$ being suitably defined in terms of $\hat{\mu}_i$, $\hat{\Sigma}_i$, and $\hat{P}(\omega_i)$. Again, multiple solutions are possible. In addition, a larger computational overhead is present, with, for example, repeated inversion of the sample covariance matrix being made. However, simplifications may be possible, by, for example, assuming the classes have equal covariance matrices, or assuming that they are each diagonal.

An elementary, approximate method for simplifying computation and accelerating convergence is the Isodata procedure ([9], [7], [1]), which is a typical methodology from a class of procedures known as *clustering* algorithms. It is an iterative optimization algorithm, like, for example, the well known k-means (or c-means) family of algorithms.

Isodata may be viewed as a way to obtain maximum likelihood estimates for the means. In general, when the component densities (within the mixture density) overlap is small, the maximum likelihood approach and the Isodata procedure may produce similar results.

Algorithm 1 Basic Isodata Clustering Procedure

1: Choose some initial values for the means $\hat{\mu}_1, \hat{\mu}_2, \cdots, \hat{\mu}_c$.
2: **repeat**
3: Classify the n samples by assigning them to the class of the closest mean. For example, by computing the squared Euclidean distance, instead of the more computationally expensive Mahalanobis distance.
4: Recompute the means as the average of the samples in their class.
5: **until** no mean has changed

3.1.3 General Bayes Classifier

Maximum likelihood methods consider the parameter vector θ to be unknown, whereas the Bayesian approach assumes that θ is a random variable with a known *a priori* distribution $p(\theta|\mathcal{X})$. Formally, the unsupervised Bayesian approach is quite similar to that of the supervised Bayesian approach. Basic assumptions for the former over that of the maximum likelihood approach are:

1. there are a known number of classes (call that number, c) to which all samples (X_i) belong
2. before any measurements are taken, the probability of membership within a particular class is known. That is, the *a priori* probabilities $P(\omega_j)$ for each class are known for $j = 1, 2, \cdots, c$. (Also called *mixing parameters*)
3. the models are known. That is, the class-conditional probability densities $p(X|\omega_j, \theta_j)$ are known for $j = 1, 2, \cdots, c$. (Also called a *component density*).
4. the model parameters are not known. That is, the parameter vector $\theta = (\theta_1, \theta_2, \cdots, \theta_c)$ is not known.
5. some knowledge about the parameters is known. That is, the *a priori* density $p(\theta)$ is known.
6. the rest of the knowledge about θ is contained within a set of n samples $\mathcal{X} = \{X_1, X_2, \cdots, X_n\}$ that were drawn independently from the mixture density: $p(X|\theta) = \sum_{j=1}^{c} p(X|\omega_j, \theta_j) \cdot P(\omega_j)$.

The basic equation for unsupervised Bayesian learning is obtained using Bayes rule, and is: $p(\theta|\mathcal{X}) = \frac{p(\mathcal{X}|\theta)\cdot p(\theta)}{\int p(\mathcal{X}|\theta)\cdot p(\theta) d\theta}$. That is, given the set of samples ($\mathcal{X}$), we want to learn the parameter vector θ for our known densities, and so it may be given as this ratio, where we are basically dividing our point estimate for θ by an average over all possible values of θ in order to normalize. This is a compressed version of that from the classical book, so any inaccuracies are mine.

The Bayesian and maximum likelihood learning approaches differ in the use of the fact that the a priori density $p(\theta)$ is used in the former and not in the latter (i.e. prior knowledge is used).

3.2 Form of Model Unknown

Perhaps one might be faced with the problem that they are given a data set, but they do not know the form of the model (density) from which the data derives. One way to solve the exact problem arising from this situation is an approach called *decision-directed* approximation. The idea is straightforward (perhaps not the implementation). One

uses *a priori* information to design a classifier that may be applied to the data in order to add labels, from which a supervised approach may then be tried. Many variants and hybridizations are possible that change when the labeling is performed versus when the classifier is updated. The Basic Isodata Procedure is an example of a such a decision-directed approximation approach. There are, of course, as with any heuristic approach such as this, many possible problems associated with the complete decision-directed procedure that may occur; any one of which might lead to incorrect results. For example, if an unfortunate sequence of samples is presented to the classifier for learning (assuming the classifier is updated in a way that is dependent on the presentation sequence) then such a classifier will label the original data with labels that do not represent the sample's true class membership; clearly a less than ideal situation.

3.2.1 Clustering

The unsupervised learning problem has a set of samples as input. Such input may be re-interpreted as a set of points (or cloud(s)) in a d-dimensional space, from which statistics (lower moments), such as the sample means or sample covariances could be computed. Obviously if the complete distribution could be computed (all of the infinity of moments) for the data, then a complete and compact description would be available for use by an algorithm that would result in no loss of information. As a practical measure, one may use c normal mixtures in order to estimate the true density(ies), but this would be imposing structure onto the data, rather than *finding structure from the data.* An alternative approach would be non-parametric estimates of the unknown mixture density. But if subclasses are one possible goal, then *clustering* may be a more direct methodology towards learning something from an unknown data set. The idea being that groups should be found that are highly internally similar and, simultaneously, externally quite dissimilar (i.e. compact, separated groups).

3.2.2 Similarity Measures

What is a natural group within a data set? For if we can answer this fundamental question, then our clustering procedure will simply need to produce these natural groups as its output. For a higher level conceptual example, consider 4 geometric shapes; a line segment, a triangle, a square and a circle. Which ones should be considered a natural group? Perhaps the latter 3, because they all enclose an area? or perhaps the first 3 because they all consist of straight line segments? Or perhaps the triangle and the square are drawn on a piece of paper in closer proximity, while the line segment and circle are closer? Now that we have these 3 possible clusterings, which one should be considered more appropriate? Clearly, the exact definition of similarity plays an absolutely crucial role in determining a possible answer to the natural grouping question.

As another example, if Euclidean distance is chosen as a measure of dissimilarity, then the input feature space will be isotropic (Greek *iso*, meaning alike or same, and *tropos*, meaning turning. See `http://amsglossary.allenpress.com/glossary/browse?s=i&p=51`). This means that the clusters that would be output will be invariant to translations or rotations (rigid-body motions of the data points) but that they would not, in general, be invariant to linear or other transformations that distort the distance relationships. The main point is simply that normalization (and/or other transformations) may dramatically change the results of a particular clustering algorithm and should be performed with care (if performed at all). For if the data consists of very highly dimensional data, for which we, as humans, cannot directly visualize, then how would we, in addition, also understand what a transformation would do to the resulting clustering unless care is taken?

3.2.3 Cluster Criterion Measures

Once a set of c clusters have been output from a clustering algorithm, one may be interested in the quality of the result. Formally, a set of n samples $\mathcal{X} = \{X_1, X_2, \cdots, X_n\}$ is partitioned into exactly c disjoints subsets $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_c$. Of course, this may be generalized so that the subsets do not need to be disjoint, but merely need to cover the set $\mathcal{X}$. In any case, given the restricted definition, one may define cluster evaluation criteria, of which examples are:

1. Sum of squared error criterion. Let n_i be the number of samples in $\mathcal{X}_i$ and let μ_i be the mean of those samples, then this criterion is:

   $$J_e = \sum_{i=1}^{c} \sum_{x \in \mathcal{X}_i} \|X - \mu_i\|^2.$$

   Intuitively, J_e measures the total squared error incurred in representing the n samples by the c cluster centers. An optimal partitioning is one that minimizes J_e. These kinds are partitions are also called *minimum variance* partitions.

2. Minimum variance criterion. J_e may be rewritten in the following form:

   $$J_e = \tfrac{1}{2} \sum_{i=1}^{c} n_i \cdot \overline{s}_i,$$

   where

   $$\overline{s}_i = \tfrac{1}{n_i^2} \sum_{X \in \mathcal{X}_i} \sum_{X' \in \mathcal{X}_i} \|X - X'\|^2.$$

   This form emphasizes the the Euclidean distance (a dissimilarity) is being used, and that other possibilities are possible.

3. Scattering Criterion. The scatter matrix for the i-th cluster $S_i = \sum_{X \in \mathcal{X}_i} (X - \mu_i)(X - \mu_i)^T$ leads to the concepts of the within-cluster scatter matrix, calculated as $S_W = \sum_{i=1}^{c} S_i$, the between-cluster scatter matrix, calculated as $S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^T$, and the total scatter matrix $S_T = S_W + S_B$. This criterion has a tradeoff, in which when the between-cluster scatter goes up, the within-cluster scatter goes down (in value). In the univariate case, the *trace* and *determinant* of this scatter matrix S_T have equivalent values. The trace of the scatter matrix (sum of diagonal elements) $\implies$ square of scattering radius $\implies$ sum of squared error criterion $\implies$ Trace$(S_W) = J_e$. The determinant of the scatter matrix measures the square of the scattering volume because it is proportional to the product of the variances in the directions of the principal axes. The determinant based criterion is:

$$J_d = |S_W| = |\sum_{i=1}^{c} S_i|.$$

4. Invariant Criterion. The eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_d$ of $S_W^{-1} \cdot S_B$ are invariant under nonsingular linear transformations of the data. Based on this, many possible criterion are possible. For example, maximizing the trace of $S_W^{-1} \cdot S_B = \sum_{i=1}^{d} \lambda_i$ is one approach. In general, invariant criterion functions are more likely to posses multiple local extrema.

3.2.4 Clustering as Iterative Optimization

Once a similarity measure and a cluster criterion measure have been selected, the process of clustering becomes a problem in discrete optimization (because we have a discrete number of classes and a discrete number of samples). In theory, this may be solved by exhaustive enumeration of all possible clusterings. For n samples and c classes, this means $\frac{1}{c!}\sum_{i=1}^{c} \begin{pmatrix} c \\ i \end{pmatrix} \cdot (-1)^{c-i} i^n$ (or approximately $\frac{c^n}{c!}$) ways of partitioning the set exist. For example, the best set of 5 clusters in 100 samples would require enumerating 10^{67} partitionings. Therefore, with such a large search space, heuristics are used in order to reduce the amount of work needed to find a global optimum, for which a local optimum is the only thing that may be guaranteed. An example of a basic procedure for minimizing the squared error criterion is illustrated in A_2.

Algorithm 2 Basic Minimum Squared Error Procedure

1: Select an initial partition of the n samples into clusters and compute J_e and the means $\hat{\mu}_1$, $\hat{\mu}_2$, $\cdots$, $\hat{\mu}_c$.
2: **repeat**
3: Select the next candidate sample $\hat{X}$. Suppose that $\hat{X}$ is currently in $\mathcal{X}_i$.
4: **if** Cardinality of $\mathcal{X}_i$ is greater than 1 **then**
5: $\rho_j = \begin{cases} \frac{n_j}{n_j+1} \left\| \hat{X} - \mu_j \right\|^2, j \neq i \\ \frac{n_i}{n_i-1} \left\| \hat{X} - \mu_i \right\|^2, j = i \end{cases}$
6: Transfer $\hat{X}$ to partition $\mathcal{X}_k$ if $\rho_k \leq \rho_j$ for all j.
7: Update J_e, $\hat{\mu}_i$, and $\hat{\mu}_k$.
8: **end if**
9: **until** Termination criteria has been satisfied. E.g. J_e has not changed in n attempts.

When this algorithm is compared with the Basic Isodata algorithm, it is seen that while the latter waits for all n samples to be reclassified before updating, the former updates after each sample as been reclassified. This algorithm has the problems that it is more prone (experimentally shown) to be trapped in local minima and the final paritioning depends on the order of presentation of the samples to the algorithm. As for all hill-climbing algorithms (i.e. greedy), this algorithm depends on the initial partitioning into the clusters.

Another approach to clustering, is that of agglomerative or divisive hierarchical clustering depending on whether the hierarchy is being built bottom-up or top-down. For the former, each sample is placed into its own cluster (so we have n clusters) and the two most similar clusters are selected to be merged. This continues until only 1 cluster is left, that contains all of the samples. This is more formally specified in A_3, which has incorporated a truncation of the resultant dendogram by the specification of a parameter c indicating how many clusters are believed to be within the data. Certainly c should be specified as 1 for an initial exploration of the data (if nothing is known *a priori*) in order to analyse the complete dendogram. In other words, if c is specified to something greater than 1, then implicitly, the dendogram is being cut at a similarity level for which the user does not know, and does not have any control.

Algorithm 3 Basic Agglomerative Clustering Procedure

1: Let $\hat{c} = n$ and $\mathcal{X}_i = \{X_i\}$, $i = 1, 2, \cdots, n$.
2: **while** $\hat{c} > c$ **do**
3: Find nearest pair of distinct clusters, say $\mathcal{X}_i$ and $\mathcal{X}_j$.
4: Merge $\mathcal{X}_i$ and $\mathcal{X}_j$, delete $\mathcal{X}_j$, and decrement $\hat{c}$ by one.
5: **end while**

Certainly, the nearest pairs of clusters in A_3 implies that a measure of nearness has been defined from one set (cluster) to another set (cluster). For example, between the two clusters, the distance between the nearest samples could be defined to be the distance between the two clusters. Or, perhaps, the distance between the farthest samples could be defined to be the distance between the clusters. In the former case, it could lead to a *minimal spanning tree algorithm* over the samples. The former distance tends to favour elongated clusters (e.g. banana shaped clusters could be found) whereas the latter tends to discourage elongated clusters. Obviously, other measures could be defined. For example, the mean or mode could be used. Selecting one definition over the other should be performed after careful analysis of the data and the domain for which the data was collected, for one does not want to impose structure onto a data set; rather, structure should be found within it.

3.2.5 Inducing a Metric

If our data set does not have a metric defined upon it, but rather a *dissimilarity* for every pair of samples, then we may induce a metric. That is, if we define a dissimilarity that is the minimum (as in the nearest pairs sense) of the dissimilarities of all objects in the two clusters that we are considering for merging, then the hierarchical agglomerative clustering algorithm will yield a dendrogram with least dissimilar clusters closest together. As such, when the dendogram is "unwound", it can be seen that all samples may be ordered by their position in the tree; giving a total order of the samples. The "unwinding" of the dendogram is performed level-wise, meaning that the depth of a node in the tree determines its order because no two nodes (samples) share the same level because of the way in which the dendogram was constructed.

3.2.6 Graph Theoretic Methods

The dendogram produced by the agglomerative hierarchical method is known by the term "tree" in the theory of graphs. As such, clustering problems may be posed in terms of graph theory. For example, the minimum spanning tree may be converted into a nearest neighbour dendogram, and hence we may obtain a clustering (partition) from this information.

3.3 Unknown Number of Clusters

Since we are within the context of unsupervised learning, it could certainly be the case that the number of clusters

contained within our samples is unknown. For example, the number of subtypes of brain cancer may be unknown to everyone, and we may be interested in learning if there are such subtypes and exactly how many of them there are so that proper treatment and may start to be investigated for each subtype.

One approach to dealing with this issue is to plot the cluster criterion measure against the number of clusters (e.g. $c = 1, 2, 3, \cdots$) and determine if a natural relationship indicates an appropriate number of clusters. Another approach involves heuristically determining a test for rejection of a specified number of clusters at a specific significance level. However, this cluster validity problem is essentially unsolved from the point of view of using the data to determine the appropriate number of clusters. Perhaps there are domain dependent ways in which experiments may be performed that could validate a particular number of clusters.

3.4 Low Dimensional Representations

Given a data set, one might be interested in obtaining a lower dimensional representation that preserves as much as possible the original structure of the data. Obtaining such a representation of the higher dimensional space may lead to a better understanding of the relationships that are occurring between the samples both in a local and a global sense.

Classical approaches are through *principal components analysis* (which, in fact, produces an orthogonal mapping that is *equal in dimension* to the original space, but for which one may disregard some of the dimensions with low variation) and *factor analysis*. These representations are forming linear combinations of the original features.

A modification of the hierarchical clustering algorithm may be made in order to produce a feature reduction algorithm, as shown in A_4. This algorithm tries to reduce

Algorithm 4 Hierarchical Dimensionality Reduction

1: Let $\hat{d} = d$ and $\mathcal{F}_i = \{x_i\}$, $i = 1, 2, \cdots, d$.
2: **while** $\hat{d} \neq d'$ **do**
3: Compute the correlation matrix and find the most correlated pair of distinct clusters of features, say $\mathcal{F}_i$ and $\mathcal{F}_j$.
4: Merge $\mathcal{F}_i$ and $\mathcal{F}_j$, delete $\mathcal{F}_j$, and decrement $\hat{d}$ by one.
5: **end while**

the number of dimensions from the starting d dimensions to the final (requested) d' dimensions by iteratively (greedily) merging the features one at a time by selecting features based on the correlation.

To remind ourselves, the variance of a vector $\underline{X}$ is defined to be $\sigma^2 = var(\underline{X}) = E[(\underline{X} - \mu)^2]$, which may be generalized to the concept of covariance between two vectors $\underline{X}$ and $\underline{Y}$ by $\sigma^2_{\underline{X},\underline{Y}} = cov(\underline{X}, \underline{Y}) = E[(\underline{X} - \mu_{\underline{X}})(\underline{Y} - \mu_{\underline{Y}})^T]$, where $\mu_{\underline{X}} = E[\underline{X}]$ and $\mu_{\underline{Y}} = E[\underline{Y}]$ (the respective expected values). In particular, the covariance between each pair of vectors from a set of n vectors $X_1, X_2, X_3, \cdots, X_n$ may be expressed in matrix form:

$$\Sigma = \begin{bmatrix} \sigma_{X_1,X_1} & \sigma_{X_1,X_2} & \cdots & \sigma_{X_1,X_n} \\ \sigma_{X_2,X_1} & \sigma_{X_2,X_2} & \cdots & \sigma_{X_2,X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_n,X_1} & \sigma_{X_n,X_2} & \cdots & \sigma_{X_n,X_n} \end{bmatrix},$$

and the relationship between covariance and correlation is defined via the following equation:

$$\rho_{X_1,X_2} = \frac{cov(X_1, X_2)}{\sigma_{X_1} \cdot \sigma_{X_2}}.$$

One serious criticism of the presented approaches, is that they are all concerned with reproducing the representation of the data (e.g. keeping the most variable features) but for classification, the interest lies in *discrimination*, not representation.

4. OTHER UNSUPERVISED ALGORITHMS

There are, of course, other unsupervised learning algorithms than those listed within one section of a classical pattern recognition book[6]. Such an example was detailed in various incarnations[2], and was originally described as the Leader Algorithm ([8] p.74), which begins with some motivation for this particular quick partition algorithm: *It is desired to construct a partition of a set of M cases, a division of the cases into a number of disjoint sets or clusters. It is assumed that a rule for computing the distance D between any pair of objects, and a threshold T are given. The algorithm constructs a partition of the cases (a number of clusters of cases) and a leading case for each cluster, such that every case in a cluster is within a distance T of the leading case. The threshold T is thus a measure of the diameter of each cluster. The clusters are numbered $1, 2, 3, ..., K$. Case I lies in cluster $P(I)[1 \leqslant P(I) \leqslant K]$. The leading case associated with cluster J is denoted by $L(J)$. The algorithm makes one pass through the cases, assigning each case to the first cluster whose leader is close enough and making a new cluster, and a new leader, for cases that are not close to any existing leaders.*

A translation of the Leader Algorithm using modern terminology was performed[2] (e.g. removing **goto** statements) and a number of variants were implemented. In addition, the original algorithm used the terminology $D(i, j)$[1], but it is believed that $D(i, L(j))$[2], may be more clear, and so the translation described in A_5 uses this change of terminology.

5. IMPLEMENTATION AND RESULTS

The basic Isodata procedure was implemented and run using simulated data. In particular, the true ω_1 mean vector μ_1 was specified as:

$$-8.48, -6.05, -7.10, -8.54, -9.16, -6.00, -6.69, -7.50.$$

While a covariance matrix that was used to generate the data associated with ω_1 was randomly generated that had the property that it was positive semi-definite:

4.52	1.26	1.73	1.41	1.48	1.34	1.63	1.93
1.26	4.45	1.52	1.11	1.33	0.91	1.77	1.50
1.73	1.52	4.82	1.51	1.65	1.77	1.18	1.52
1.41	1.11	1.51	5.00	1.05	2.33	1.82	1.97
1.48	1.33	1.65	1.05	4.86	0.74	1.73	2.34
1.34	0.91	1.77	2.33	0.74	5.27	2.24	0.90
1.63	1.77	1.18	1.82	1.73	2.24	5.18	1.66
1.93	1.50	1.52	1.97	2.34	0.90	1.66	5.83

[1]meaning that case i is measured in terms of distance to cluster j

[2]meaning that case i and case $L(j)$ are measured in terms of distance to each other, where $L(j)$ is the leader for cluster j

Algorithm 5 Hartigan's Leader Algorithm (Translation)

Input: Data X, number of cases M, distance threshold T_d
Algorithm Negative Properties: *i)* the first data object always defines a cluster and therefore, appears as a leader *ii)* the partition formed is not invariant under a permutation of the data objects *iii)* the algorithm is biased, as the first clusters tend to be larger than the later ones since they get first chance at "absorbing" each object

```
1: k ⇐ 1                                   ▷ The current number of clusters
2: P(1) ⇐ 1          ▷ Classify the first case into the first cluster
3: L(1) ⇐ 1          ▷ Define the leading case of the first cluster
4: for i ⇐ 2 to i ≤ M by i ⇐ i + 1 do      ▷ For every case but the first in the data set
5:    P(i) ⇐ −1      ▷ Case i is not assigned to a cluster yet
6:    for j ⇐ 1 to j ≤ k by j ⇐ j + 1 do   ▷ For each currently known cluster
7:       if D(i, L(j)) ≤ T_d then          ▷ Current case is within the threshold
8:          P(i) ⇐ j                      ▷ Case i is assigned to cluster j
9:          break for
10:      end if
11:   end for
12:   if P(i) = −1 then ▷ Case i isn't close enough to one of the existing leaders
13:      k ⇐ k + 1                         ▷ Create a new cluster
14:      P(i) ⇐ k                          ▷ Classify case i to the new cluster
15:      L(k) ⇐ i                          ▷ Define the leader of the new cluster
16:   end if
17: end for
```

Since a 2-class problem was being investigated, the true ω_2 mean vector μ_2 was specified to be:

6.64, 6.15, 7.57, 7.77, 8.74, 5.26, 9.23, 7.91.

While a covariance matrix for class ω_2 was also generated with the positive semi-definite property:

7.41	2.08	0.88	2.41	3.23	0.70	1.82	1.15
2.08	4.75	1.21	0.76	2.83	2.16	1.40	2.29
0.88	1.21	5.54	2.93	1.16	1.72	1.30	1.10
2.41	0.76	2.93	8.94	1.31	2.67	2.16	2.03
3.23	2.83	1.16	1.31	8.33	0.88	2.88	0.90
0.70	2.16	1.72	2.67	0.88	4.02	1.74	2.08
1.82	1.40	1.30	2.16	2.88	1.74	6.21	0.76
1.15	2.29	1.10	2.03	0.90	2.08	0.76	7.11

After the 2 classes were generated using the information above, the samples were randomly shuffled together (like two halves of a deck of cards being shuffled back together) in order to present a data set for the Isodata procedure that was not an easy unsupervised learning problem. After Isodata was executed, it added labels to each of the 100 artificial samples (50 per class, so it was a balanced learning problem). The learned mean vectors for the 2 classes are:

$\hat{\mu}_1$ = −8.851422, −6.302922, −7.102150, −8.531888, −9.380122, −5.562690, −6.959866, −7.887627

$\hat{\mu}_2$ = 6.061690, 5.905239, 7.681886, 6.929916, 8.453957, 5.121132, 8.754080, 7.716288

When the learned $\hat{\mu}_1$ is compared with the true μ_1 it can be seen that they are quite close in actual value. The difference is due to the fact that the samples were *randomly generated* from the true mean vector, so it would be highly improbably for the samples to have a sample mean equal to the population mean from which they were generated. It can be seen to be a similar situation when comparing the estimated $\hat{\mu}_2$ and the true $\hat{\mu}_2$ for the other class. That is, they are also quite similar estimates for the true population mean vector.

Since this was an unsupervised problem, the samples given to the Isodata procedure did not have labels associated with them. Hence, the problem was for Isodata to apply a classification to the samples (given that Isodata was told that the data contained 2 classes). As such, when the randomized samples labels were compared to their known values, it was seen that Isodata achieved 100% accuracy. This is certainly a positive result because the simulated data was well separated, and such an accuracy would be expected, if not demanded from the unsupervised learning procedure.

Investigating a little bit deeper, it can be seen in Fig.1 that the distances for all of the points within each class (Class 1 points on the left and Class 2 points on the right of the figure) lie within an interval that is separated from each class mean by about 2.4 and 3 measurement units. The distribution of the distances is such that the majority of the points are close to the center, while fewer and fewer points lie further and further away. This is certainly as expected and hoped for within a well formed data set and unsupervised learning problem.

6. CONCLUSION

The Isodata clustering algorithm was implemented and was able to successfully learn a 2 class classification of a well separated simulated data set.

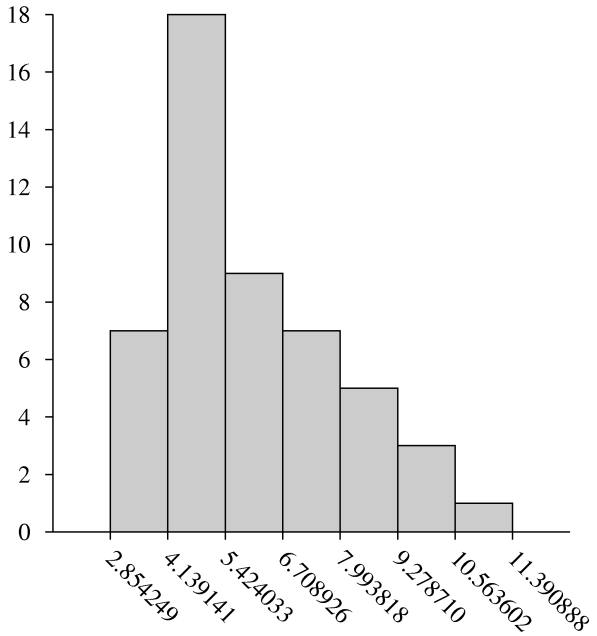
The review has shown examples of unsupervised learning algorithms that make certain assumptions about the underlying properties of the input data. In particular, the largest assumption seems to be very near to the beginning of the unsupervised learning methodology; that measurements of real world objects are recorded, for example, in the binary, integer or real domains. But what of real world objects that may have other types of data associated with them? How would unsupervised learning occur in those situations? Should that additional data be thrown away? Answering these questions is certainly outside the scope of the course requirements and hence this review.

Acknowledgements

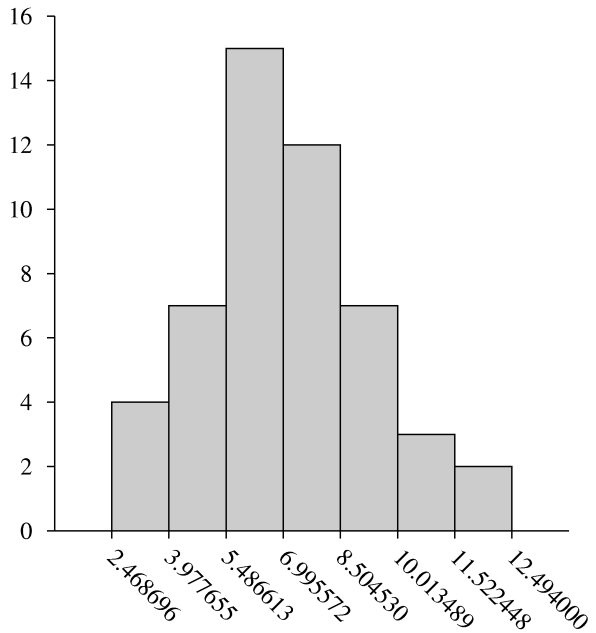
The author would like to thank Professor John Oommen from the University of Carleton for providing a productive learning environment. The author certainly appreciates Julio Valdés, Bob Orchard and Fazel Famili from the NRC's Institute for Information Technology (NRC-IIT) for their continued support and encouragement. In addition, the author would like to thank David G. Goodenough from the Pacific Forestry Center (PFC) and adjunct at the University of Victoria and Andrew Dyk (PFC) for their help in 1999 when learning about pattern recognition applied to the field of digital remote sensing.

REFERENCES

[1] A. J. Barton. Analysis of a Landsat 7 image of Ashdod, Israel. Technical report, University of Victoria, Victoria, British Columbia, December 1999.



(a) Distances from $\hat{\mu}_1$ for 8-dimensional points classified as ω_1. $card\,(\omega_1) = 50$, $range\,(distances) = [2.85, 11.39]$ and Bin width (1.2849) calculated using Sturge's Rule.



(b) Distances from $\hat{\mu}_2$ for 8-dimensional points classified as ω_2. $card\,(\omega_2) = 50$, $range\,(distances) = [2.47, 12.49]$ and Bin width (1.5089) calculated using Sturge's Rule

Figure 1: An unsupervised classification procedure was implemented (Isodata) and 8-dimensional points were classified into one of 2 classes (artificially generated data... see text for μ_1, Σ_1, μ_2 and Σ_2).

Final Project report for directed studies course CSC490 entitled Introduction to Pattern Recognition.

[2] A. J. Barton. Modelling variability in the leader algorithm family: A testable model and implementation. Technical Report NRC 47429. ERB-1119., National Research Council Canada, Institute for Information Technology, December 2004.

[3] A. J. Barton. Parametric and non-parametric classifiers applied to the supervised classification of proteomic data. Technical Report NRC 48805. ERB-1142., National Research Council Canada, Institute for Information Technology, December 2006.

[4] A. J. Barton. Simple self-adjusting data structure use: An empirical investigation of the unsupervised construction of conceptual units from mass spectrometry data. Technical Report NRC 48728. ERB 1139., National Research Council Canada, Institute for Information Technology, April 2006.

[5] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[6] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley New York, 1973.

[7] M. Friedman and A. Kandel. *Introduction to Pattern Recognition: Statistical, Structural, Neural and Fuzzy Logic Approaches*, volume 32 of *Series in Machine Perception and Artificial Intelligence*. World Scientific, New Jersey, 1999.

[8] J. A. Hartigan. *Clustering Algorithms*. Wiley Series in probability and mathematical statistics. John Wiley and Sons, Inc., 1975.

[9] J. Tou and R. Gonzalez. *Pattern Recognition Principles*. Don Mills, Ontario, Addison-Wesley, 1974.

[10] J. Valdés and A. Barton. Virtual reality visual data mining via neural networks obtained from multi-objective evolutionary optimization: Application to geophysical prospecting. In *2006 IEEE International Joint Conference on Neural Networks (IJCNN 2006)*, number NRC 48504, Vancouver, British Columbia, Canada, July 2006.