

# Pattern Recognition Technique based on Adaptive Fuzzy k- mediod Clustering using Domain Knowledge

Divya Jain<sup>1</sup> & Vipin Tyagi<sup>2</sup>

<sup>1</sup>Department of Mathematics

<sup>2</sup>Department of Computer Science

Jaypee University of Engineering and Technology,  
A. B. Road Raghogarh, Guna

## ABSTRACT

In the real world problems various pattern recognition technologies process huge amount of pattern to discover relevant knowledge. These techniques are computationally expensive. Additional knowledge also known as domain or background knowledge can help us in reducing the search as well as to optimize the hypotheses by decreasing the size of the search area. In the present paper we discuss the processes of domain knowledge, in effectively discovering knowledge. On the reduced search area we apply the dynamic fuzzy K-mediod technique to clusters these patterns in various clusters, the system is made adaptive to these dynamic changes. This technique finds many applications in various fields, like medical sciences, fraud detection in bank customer etc.

**Keywords:** Databases, knowledge discovery, domain knowledge, hypothesis, optimization.

## 1. INTRODUCTION

Knowledge discovery is defined as the nontrivial extraction of inherent, previously unknown, and potentially useful information from data. Various pattern recognition technologies process huge amount of patterns to discover relevant knowledge. Various organizations have initiated to widen or occupy tools to discover knowledge from databases. For example, banks are analyzing data to find better rules for credit assessment. In a "hard" clustering algorithm, each pattern vector must be assigned to a single cluster. This "all or none" membership restrictions are not a realistic one, since many pattern vectors may have the distinctiveness of several classes. It is more natural to allocate to each pattern a vector set of memberships, one for each class. The proposition of this is that, the class boundaries are "fuzzy" rather than "hard".

## 2. FUZZY CLUSTERING

The notion of fuzzy sets, first put forth by Zadeh [14, 15], is an attempt to modify the basic conception of a space-that is, the set on which the given problem is defined. By introducing the concept of a fuzzy set i.e., an unsharply defined set, a different perspective is provided for certain problems in systems analysis, including pattern recognition. One of the significant

difficulties in development of a methodical approach to pattern recognition is that the phenomena of concern are modeled by equations which contain functions and operators which may appear simple and natural, but which yield some solutions which could be regarded as pathological. The problem from our view point is to differentiate between classes in manner which is simple and easy to visualize. In doing so, we restrict the solutions in an unknown way. The use of fuzzy sets by Zimmermann, H.-J.[16] is an attempt to reorganize this problem. Pattern classification problems have provided impulsion for the development of fuzzy set theory. Recently, fuzzy sets have provided a theoretical basis for cluster analysis with the introduction of fuzzy clustering. The first fuzzy clustering algorithm was developed in 1969 by Ruspini [10,11], Boutleux, E., Dubuisson, B[3] .and used by several others. Following this, Dunn[5] developed the first fuzzy extension of the least-squares approach to clustering and this was generalized by Bezdek[2]. Several problems in medical diagnosis have been attacked using fuzzy clustering algorithms. It appears that medical diagnosis may be an especially fruitful area of application for fuzzy clustering, since biological systems are extremely complex and the boundaries between distinct medical diagnostic classes are not sharply defined.

Another problem is that the set of all partitions resulting from "hard" clustering algorithm is extremely large, making an exhaustive search extremely complicated and expensive. Fuzzy clustering will generally lead to more computational tractability. Another advantage of fuzzy clustering is that troublesome or outlying members of the data set are more easily recognized than with hard clustering, since the degree of membership is continuous rather than "all-or-none." Bezdek[2] and Dunn[5] have noted the relationship of fuzzy clustering to estimating mixture distributions, but retained the Euclidean metric.

Sometimes, huge amount of pattern sample makes the process computationally very expensive. The vastness of the data compels to use the techniques of focusing on specific portion of the patterns, which requires some additional information about the form of patterns and conditions applied on it. This information, known

as domain or background knowledge, can be defined as any information that is not unambiguously presented in the patterns, can often help us and restrict the search for interesting knowledge .i.e. in particular, we look at the use of domain knowledge to reduce the investigation as well as to optimize the hypotheses which represent the interesting knowledge to be discovered.

A knowledge discovery in pattern recognition system must be able to represent and appropriately use domain knowledge in combination with the application of discovery algorithms. Domain knowledge assists knowledge discovery by focusing search. However, we should be careful in using domain knowledge to narrow the search in a pattern recognition system in order to avoid blocking the discovery of unexpected knowledge. Domain knowledge has been used in different aspects of the knowledge discovery in few systems. For example, Meta-Dendral uses domain knowledge (knowledge of chemistry) heavily for both hypothesis (representing the knowledge to be discovered) generation and testing. Prospector uses its domain (geological) knowledge in the same areas as Meta-Dendral.

Although the use of domain knowledge in knowledge discovery has been mentioned by researchers[6,7] the literature does not have any detailed discussions regarding the use of domain knowledge in different aspects of the knowledge discovery.

In particular, we look at the use of domain knowledge to

- (i) reduce the size of the databases
- (ii) optimize the hypotheses, which represent the interesting knowledge to be discovered
- (iii) optimize queries used to prove the hypotheses.

### **3. DOMAIN KNOWLEDGE**

With reference to M. Mehdi Owrang.0., [9] Domain or background knowledge can be defined as any information that is not explicitly presented in the system. In a medical diagnosis, for example, the knowledge "male patients can not be pregnant" or "male patients do not get breast cancer" is considered to be domain knowledge since it is not contained in the database directly. Similarly, in a business database, the domain knowledge "customers with high-income are good credit risks" may be useful even though it is not always true. Other types of knowledge like inter-field knowledge (e.g., experience and salary being positively correlated) and inter-instance. Domain knowledge originates from many sources. A data dictionary is the most basic form of domain knowledge. Typical information in the data dictionary includes the types of attributes, size of attributes, name of attributes, meaning of each attribute, format, constraints, domain of attribute, usage statistics, access control, mapping definitions, etc. Additional information about the specific analysis objectives may come from the domain expert (although it may be

generated automatically from the database. and can assume many forms).

Limiting the number of fields alone may not sufficiently reduce the size of the pattern set, in which case a subset of records must be selected. In the second approach, we can apply the discovery algorithms to a random sample of pattern. However, the rules discovered in a sample can be invalid on the full pattern set. Finally, in the third approach (taken by M. Mehdi Owrang 0[9], additional information, called domain knowledge, can be used to guide and constrain the search for interesting knowledge. The search time can be minimized by reducing knowledge can be related to pattern, but they are more related toward the semantic of the domain the size of the pattern recognition systems (eliminating irrelevant attributes as well as records), and optimizing the hypothesis that represents the knowledge to be discovered, and optimizing queries that are used to process the pattern to prove the hypothesis.

### **4. USING DOMAIN KNOWLEDGE TO REDUCE PATTERN RECOGNITION SIZE AREA**

Domain knowledge can be used to reduce the size of the pattern recognition system that is being searched for discovery by eliminating pattern records that are not needed for discovery. Consider a medical database in which simple domain knowledge could be "male patients can not be pregnant". If the knowledge to be discovered is "whether drug X has effects on pregnant patients", domain knowledge can be used to reduce the size of search area by eliminating the records for male patients from consideration. Other domain knowledge, for example, "female patients below 12 years or above 65 years can not be pregnant", can be applied to further reduce the size of the search area. Too much reliance on domain knowledge, however, may unduly constraint the knowledge discovery and may block unexpected discovery by leaving portions of the search area unexplored.

Consider the following hospital data file: Assume that the knowledge to be discovered is "the effects of drug X on patients with heart disease" and domain Drug X has these effects on people over 20 with heart disease and these effects on people under 20 with heart disease. Excluding this domain knowledge during discovery may help to classify the patterns more efficiently. For example, our pattern may support that drug X has different effects on people under 20 and over 20. However, due to the elimination of part of the search area (records for patients under (20), the discovery scheme just can't find enough pattern to support this.

In another example, if we use domain knowledge "male patients do not get breast cancer" for the hypothesis "effects of drug X on patients with breast cancer", we may never discover that male patients can have breast cancer. There are several things that we

can do to improve the effective use of domain knowledge in knowledge discovery and to avoid blocking the unexpected discovery.

First, the domain expert can assign a confidence factor to each domain knowledge and uses it only if the confidence factor is greater than a specified threshold. The assignment of a confidence factor to domain knowledge depends on how close the domain knowledge is to the established facts. The domain expert needs to define a mechanism to calculate the confidence factor of a domain knowledge that is derived from the given domain knowledge.

Second, rarely is discovered knowledge true across the entire pattern. It is important to represent and convey the degree of certainty to decide how much confidence the system or user should put into a discovery. Certainty involves several factors including the integrity of the pattern, the size of the sample on which the discovery is performed, and perhaps the degree of support from available domain knowledge.

Therefore, if the size of the search area is reduced too drastically after using some domain knowledge, then we may consider using fewer domain knowledge, or none of them, in order to avoid blocking unexpected discovery results. Otherwise, knowledge is discovered that does not have a high enough confidence factor to be considered interesting.

## 5. FUZZY K- MEDIOD ALGORITHM

Corresponding to Tyagi and Jain [12]

$$V(t) = \{v_1(t), \dots, v_c(t)\}.$$

where,

$$v_i(t) = \frac{\sum_{k=1}^N (au_{ik}^m + bt_{ik}^\eta) x_k(t)}{\sum_{k=1}^N (au_{ik}^m + bt_{ik}^\eta)} \quad (5.1)$$

in order to approximate the data structure at time instant  $t$  taking into account the history of dynamic development of feature vectors.

It should be stressed that fuzzy partition matrix  $U(t) = [u_{ik}(t)]$  and possibilistic function matrix  $T(t) = t_{ik}(t)$ , as well as cluster prototypes  $V(t)$ , evolve dynamically as new observations become available.

The Fuzzy K- medioids algorithm partitions a collection of  $n$  pattern points into  $K$  fuzzy clusters (where  $c < n$ ), and simultaneously seeking the best possible locations of these clusters. This method uses distance concept in  $n$ -dimensional Euclidean space to determine the geometric closeness of pattern points by assigning them to various clusters or classes.

The mathematical notations used in developing fuzzy K- medioids algorithm:

$x_k = k^{th}$  Pattern point (possibly  $m$  dimensional vector and  $k = 1, 2, \dots, c$ )

$v_i =$  The centre of the  $i^{th}$  fuzzy cluster ( $i = 1, 2, \dots, c$ ).

$$d_{ik} = \|x_k - v_i\|_2 = \left[ \sum_{j=1}^m (x_{kj} - v_{ij})^2 \right]^{1/2}$$

is the distance between  $x_k$  and  $v_i$ .

The partitioned clusters are typically defined by the membership matrix of order  $(c \times n)$ , where the elements of the matrix are  $\mu_{ik}$ . Here  $\mu_{ik}$  specifies the degree of membership of  $k^{th}$  data point in  $i^{th}$  cluster and its value lies between  $[0, 1]$ . i.e.

$$M = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \dots & \dots & \dots & \dots \\ \mu_{c1} & \mu_{c2} & \dots & \mu_{cn} \end{bmatrix} \quad (5.1)$$

where  $\sum_{i=1}^c \mu_{ik} = 1$ .

Here the objective function is

$$\min J(M, v_1, v_2, \dots, v_c) = \sum_{i=1}^c \sum_{k=1}^n (\mu_{ik})^q (d_{ik})^2 \quad (5.2)$$

where  $q \in [1, \infty)$  is weighting exponent parameter and it controls the extend of membership sharing between fuzzy clusters. When  $q = 1$ , fuzzy K- medioids algorithm tends to a hard K means algorithm. In general greater is the value of  $q$ , the fuzzier are the membership assignments of the clusters. Usually  $q = 2$  is the preferable choice.

$$\mu_{ik} = \left[ \sum_{j=1}^c \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{q-1}} \right]^{-1} \quad (5.3)$$

Once the centers of the fuzzy clusters have been determined, each of the “ $n$ ” data points can be allocated to the cluster with the closest cluster center.

In the real world, most of objects can be treated as one kind of patterns in general. Although patterns are ubiquitous and of varied forms, they can be generally divided into two categories: static and dynamic patterns. In nature, static patterns tend to unchanged

dramatically upon generation whilst dynamic patterns always tend to change in a wide range. As a prominent characteristic, the variability of dynamic patterns in the same class becomes gradually larger, which leads to tremendous difficulties in discriminating between dynamic patterns belonging to different classes.

In addition, dynamic patterns often convey mixing information, which is often hardly separable, so that the direct use of the mixing information can result in the inadequate performance for a specific task due to the interference of irrelevant information. The characteristics of dynamic patterns give rise to several challenging problems as follows: how to generate a parsimonious yet robust representation of dynamic patterns that tolerates their intra and interclass variability, how to highlight the relevant information and simultaneously suppress affection of any irrelevant information, how to make use of intrinsic contextual information for dynamic pattern analysis and discovery, and how to create an appropriate learning model for dynamic pattern recognition to yield the consistently good generalization without being affected by the constant change of dynamic patterns.

Let  $X(t) = \{x_1(t), \dots, x_N(t)\}$  be a set of dynamic objects,  $t = 1, \dots, t_p$  is given as a dynamic sequence of observations. The time interval of observations can in general be unlimited  $t \in [1, \infty)$ . Each object correspond to an  $M$ -dimensional trail in the feature space, this contains a history of dynamic development of each feature. Even though process or system variables observed can be continuous in nature, measurements of these variables are usually carried out discretely with a certain sampling rate. Therefore supposing that dynamic objects are observed at discrete time instants, a trajectory can be given by a discrete vector-valued function of the form:

$$x_k(t) = [x_k(t_1), x_k(t_2), \dots, x_k(t_p)]^T, \quad (5.4)$$

$k = 1, \dots, N,$

where  $p$  is the number of observations in a trajectory and  $x_j(t_p)$ , is an observation of a feature vector at

time instant  $t_p$ . Substituting an  $M$ -dimensional feature vector into the components of this function, a matrix representation of a trajectory of a dynamic object is obtained:

$$x_k(t) = \begin{bmatrix} x_{k1}(t_1) & x_{k2}(t_1) & \dots & x_{kM}(t_1) \\ x_{k1}(t_2) & x_{k2}(t_2) & \dots & x_{kM}(t_2) \\ \dots & \dots & \dots & \dots \\ x_{k1}(t_p) & x_{k2}(t_p) & \dots & x_{kM}(t_p) \end{bmatrix} \quad (5.5)$$

where columns correspond to trajectories of single features and rows correspond to feature vectors at a certain time instant. A trajectory which explicitly contains time as an additional feature so that single

features are time-dependent can be called time series. In general, trajectories can describe a dependence of features from another variable implicitly related to time.

In order to approximate the data structure at time instant  $t$  taking into account the history of dynamic development of feature vectors. It should be stressed that fuzzy partition matrix  $U(t) = [u_{ik}(t)]$  as well as cluster prototypes  $V(t)$ , evolve dynamically as new observations become available.

The dynamic development of cluster prototypes can be represented by the following model:

$$V(t) = \alpha(t)\Gamma_1[V(t-1)] + (1-\alpha(t))\Gamma_2[V(t-1)],$$

$$\alpha(t) \in [0,1]$$

Where  $\Gamma_1$  a transformation due to swift changes in the cluster is structure (formation, merging, splitting or destruction of clusters) and  $\Gamma_2$  is a transformation due to steady changes in the cluster structure. Transformation  $\Gamma_1$  consists of two further transformations: a transformation  $\Gamma_3$  regarding a change in the number of clusters and a transformation  $\Gamma_4$  regarding a change in the locations of the cluster prototypes. According to the four types of swift changes considered here, a change in the number of clusters can be modeled by a linear function:

$$\Gamma_3[c(t)] = c(t-1) + \beta(t), \beta(t) \in I$$

Transformation  $\Gamma_4$  is a non-linear function depending on a set of unknown parameters. Transformation  $\Gamma_2$  is obtained from the recursive equation for calculating the cluster prototypes. Refer to Chih-Cheng Hung, Wenping Liu and Bor-Chen Kuo[4] the system is made adaptive to the various dynamic changes.

## 6. CLUSTER VALIDITY MEASURES FOR DYNAMIC CLASSIFIERS

Validity measures are applied in a dynamic classifier design, to control the process of adaptation of a classifier to dynamic changes in the cluster structure. The motive behind it is to compare the two partitions: before and after an adaptation of the classifier.

It is assumed that the optimal number of clusters is determined by the monitoring procedure and the task of the validity measure is to confirm the new partition obtained. In order to choose a suitable validity measure it is reasonable to take into account the types

of dynamic changes that can appear in the cluster structure and lead to the modification of the classifier.

The cluster validity measure should satisfy the following requirements:

- (i) Clear separation in the resulting clusters
- (ii) High Density around cluster centers
- (iii) Minimum volume of clusters

A large number of validity measures exist in the literature. The following are the various fuzzy validity measures:

Gath and Geva [8] proposed the validity measure as:

If  $F_i$  is the fuzzy covariance matrix of cluster  $i$  defined as:

$$F_i = \frac{\sum_{j=1}^N (\mu_{ij}) \|x_j - v_i\|_A^2}{\sum_{j=1}^N (\mu_{ij})} \quad (6.1)$$

then the hyper volume is calculated by

$$V_{HV} = \frac{\sum_{i=1}^c (n_i)^{good}}{\sum_{i=1}^c (h_i)} \quad (6.2)$$

where  $(n_i)^{good}$  is the sum of good objects in cluster  $i$  defined as those objects whose distance to the cluster center does not exceed the standard deviation of features for this cluster and the number of objects is determined by a sum of their degrees of membership to cluster  $i$ .

The average density is calculated by

$$V_{HV} = \frac{1}{c} \sum_{i=1}^c \frac{(n_i)^{good}}{(h_i)} \quad (6.3)$$

The validity criteria proposed by Xie and Beni [13] is based on compactness and fuzzy separation of fuzzy clusters as:

$$\pi_i = \frac{\sum_{j=1}^N (\mu_{ij})^2 \|x_j - v_i\|_A^2}{n_i}; i = 1, 2, \dots, c \quad (6.4)$$

$$S_i = \sum_{r=1}^c \|v_i - v_r\|_A^2; i = 1, 2, \dots, c \quad (6.5)$$

where  $n_i = \sum_{j=1}^N (\mu_{ij})$  is the fuzzy cardinality of cluster  $i$  and  $A$  is an arbitrary symmetric positive definite matrix.

The validity index for cluster  $i$  is given by the ratio between its fuzzy separation and its compactness and the total validity measure is obtained by

$$SC_1 = \sum_{i=1}^c \frac{S_i}{\pi_i} = \sum_{i=1}^c \frac{n_i \sum_{r=1}^c \|v_i - v_r\|_A^2}{\sum_{j=1}^N (\mu_{ij})^2 \|x_j - v_i\|_A^2}; i = 1, 2, \dots, c \quad (6.6)$$

A larger value of  $SC_1$  indicates a better partition, i.e. a fuzzy partition characterized by well separated and compact fuzzy clusters.

Corresponding to the above validity measure we proposed a new possibilistic fuzzy validity measure based on separation and compactness as given by

$$\omega_i = \frac{\sum_{j=1}^N (a\mu_{ij} + bt_{ij})^2 \|x_j - v_i\|_A^2}{n_i}; i = 1, 2, \dots, c \quad (6.7)$$

$$\rho_i = \sum_{r=1}^c \|v_i - v_r\|_A^2; i = 1, 2, \dots, c \quad (6.8)$$

where  $n_i = \sum_{j=1}^N (a\mu_{ij} + bt_{ij})$  is the fuzzy cardinality of cluster  $i$  and  $A$  is an arbitrary symmetric positive definite matrix.

The validity index for cluster  $i$  is given by the ratio between its fuzzy separation and its compactness and the total validity measure is obtained by

$$SC_2 = \sum_{i=1}^c \frac{\omega_i}{\rho_i} = \sum_{i=1}^c \frac{n_i \sum_{r=1}^c \|v_i - v_r\|_A^2}{\sum_{j=1}^N (a\mu_{ij} + bt_{ik})^2 \|x_j - v_i\|_A^2}; i = 1, 2, \dots, c \quad (6.9)$$

Here larger value of  $SC_2$  indicates a better partition, i.e. a possibilistic fuzzy partition characterized by well separated and compact possibilistic fuzzy clusters, considering both fuzzy membership and typicality values.

## 7. CONCLUSION

Larger pattern sizes containing incomplete and inaccurate pattern, makes knowledge discovery to be more difficult. We have discussed the advantages of using domain knowledge to confine the search in discovering knowledge from huge pattern sizes. Domain knowledge can be used to reduce the search by reducing the size of the search area, reduce the size of the hypotheses by eliminating unnecessary conditions from the knowledge.

Suppose the discovered knowledge is: Drug X has such and such effects on people with heart disease. If we avoid using domain knowledge, the knowledge discovery system may find out a more reasonable result such as: hypotheses and remove unnecessary operations from a query that is used to process the pattern to prove (or disprove) the hypotheses. To avoid blocking unexpected discovery, we suggested to assign confidence factor to domain knowledge and use them when these confidence factors are high enough based on user's specification. In addition, we recommended to use domain knowledge when their use do not lead to a major reduction in the systems in order to avoid having few sample pattern for discovery or missing interesting pattern. Further the reduced domain size is subjected to fuzzy K- mediod clustering to cluster the static patterns. Usually the pattern types are dynamic in nature hence time constrains is also considered while clustering.

## 8. REFERENCES

- [1] Bajaj, R. K., Srivastava, A. and Hooda, D.S. (2009), "Fuzzy Clustering Algorithm for Testing the Convergence of Projected Per Capita GDP of BRIC and G6 Countries", *Inter. Journal of Mathematics and Applied Statistics*, 1, 1, 1-12.
- [2] Bezdek, J.C. (1981), "Pattern Recognition with Fuzzy Objective Function Algorithms". New York, Plenum.
- [3] Boutleux, E., Dubuisson, B. (1996), "Fuzzy Pattern Recognition to Characterize Evolutionary Complex Systems", Proc. IEEE Int. Conference on Fuzzy Systems, New Orleans, LA.
- [4] Chih-Cheng Hung, Wenping Liu and Bor-Chen Kuo(2007), "A New Adaptive Fuzzy Clustering Algorithm For Remotely Sensed Images" Beijing Municipal Education commission (No. KM20071000900), Beijing, China, and 863 Hi-tech Research and Development of China (No. 2006AA10Z232).
- [5] Dunn, J.C. (1974), "Well Separated Clusters and Optimal Fuzzy Partitions", *J. Cybern.*, 4,3, 95-104.
- [6] Fayyad, U.M.; Gregory Piatetsky-Shapiro, Padhraic Symth,"From Data Mining to Knowledge Discovery: An Overview", PP. 1-34, *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P. Symth, Editors, AAAI Pressme MIT Press, 1996.
- [7] Frawley, William J., Gregory Piatetsky-Shapiro, and Christopher J. Matheus, "Knowledge Discovery in Databases: An Overview", *AI Magazine*, fall 1992, Vol. 14, No. 3, PP. 57-70.
- [8] Gath, I. and Geva, B. (1989), "Unsupervised Optimal Fuzzy Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 8, 773-781.
- [9] M. Mehdi Owrang 0( 1997), "Optimization of Knowledge Discovery Process Using Domain Knowledge" *IEEE*, 428-433.
- [10] Ruspini, E.H. (1969), "A New Approach to Clustering", *Information Control*, 15, 1, 22-32.
- [11] Ruspini, E.H. (1970), "Numerical Methods for Fuzzy Clustering", *Information Sciences*, 2, 319-350.
- [12] Tyagi ,V.,and Jain,D., (2009), "Pattern Recognition Using Adaptive Dynamic Possibilistic Fuzzy Technique", *International Systems of Fuzzy Systems and Rough Systems, Serials Publication*, Volume2, 45-52.
- [13] Xie, X. L., Beni, G. (1991), "A Validity Measure for Fuzzy Clustering", *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 13, 8, 841-847.
- [14] Zadeh, L.A. (1965), "Fuzzy Sets", *Information and Control*, 8, 338-353.
- [15] Zadeh, L.A. (1984), "Making Computers Think Like People", *IEEE. Spectrum*, 8, 26-32.
- [16] Zimmermann, H.-J. (1996), "Fuzzy Set Theory - and its Applications" Third edition, Boston, Dordrecht, London.