IBIMA Publishing
Communications of the IBIMA
http://www.ibimapublishing.com/journals/CIBIMA/cibima.html
Vol. 2011 (2011), Article ID 124938, 8 pages
DOI: 10.5171/2011.124938

Distinctive Feature Extraction for Fast and Reliable Classification in Complex Systems

Seyed Shahrestani

School of Computing and Mathematics, University of Western Sydney, Australia

Abstract

In this work, an approach for establishment of class membership in complex systems is reported. The classification is based on adaptive recognition facilitating the discovery of pattern features that make them distinct from objects belonging to different classes. By viewing a pattern as a representation of extracts of information regarding various features of an object, most traditional recognition methods tend to achieve categorization by identifying the resemblances amongst the class members. In this work, a different view of classification is presented. The classification is based on identification of distinctive features of patterns. It argues that the basic functioning of the established methods also implies that the members of different classes have different values for some or all of such features expressing the objects under consideration. That is, the categorization can also be based on recognition of dissimilarities and distinctions between the objects fitting in different classes. Our proposed approach in based on identifying such charactering dissimilarities, which will then form the distinctive features of patterns and objects. In other words, objects are classified as members of a particular class if they possess some features, which make them distinguished from other objects present in the universe of objects. The proposed approach and its language work in a general manner. Consequently, the corresponding codes can be developed and utilized as a general adaptive pattern recognition scheme. The generality of the approach proposed in this work, makes it applicable to many classification and pattern recognition problems encountered in complex systems.

Keywords: **-** Adaptive Recognition, Distinctive Features, Knowledge Base, Negative Recognition

Introduction

In essence, a pattern can be considered as an extract of information regarding various characteristics or features of an object, state of a system, and the like. The pattern of an object with n features under consideration, is normally represented as an n-dimensional vector, $\mathbf{p}_X$. Classification can then be regarded as the act of partitioning the feature space into K_1 regions or classes, and identification of necessary and sufficient conditions that describe membership criteria for each class, C_X. Many methods and surveys of these methods for such adaptive pattern recognitions do exist, for example see classical titles such as, Lee (1988), Therrein (1989), and Pao (1989).

To improve recognition rate and to reduce misclassification error, particularly when dealing with situations that the number of classes is large, or cannot be pre-determined, some researchers have suggested clustering. For example, based on their similarities, the patterns are mapped into a generalized indicator vector. In turn, this is then combined with a standard search tree method for categorization and recognition purposes as discussed for instance by Wilkes (1993) and more recently by Vaucher (2008) or

Wang (2010). Another proposed method explained in Basri (1993), is to find a pattern prototype – a typical example of some classes – and use that for establishing the category of a new pattern, before comparing it with all other examples of that category to recover its specific identity.

Other researchers have suggested the use of some distance metric to computes a similarity index between the patterns first, before merging close objects with each other. Some examples of such approaches are those discussed in Kurosawa (2008) and Jin, et al (2010). The index is then used to compute the center of the group comprising of the merged objects. Subsequently, the centers are taken as the representative pattern for each group. The process is carried out in a recursive fashion on the representative patterns, until the number of centers stabilizes. The ensuing hierarchy and centers are then used for efficient recognition of newly encountered patterns. These techniques and their modified versions are further explained in Commike (1991) and Todd (2007).

Clearly, it will be beneficial to have classification processes that can easily manage addition of new features and categories, with capabilities to process information in parallel, able to adjust themselves based on their previous experience or misclassifications. Such classifiers are able to adapt themselves to new feature spaces, able to partition such spaces adaptively. Adaptive pattern recognition approaches can be successfully utilized in many areas of industrial and research significance. In particular, they can be applied to areas such as the development of knowledge-based and expert systems, fault diagnosis and management of complex systems. This is a mature but ongoing field of research with good explanations available in Mourot (1993), Zhao (2005), and Shahrestani (2009). Some implementations of expert systems for specific purposes can be found in Zheng (2006) and Long (2008).

Generally speaking, systems relying on heuristic rules are considered to be fragile. More specifically, for new situation falling outside the rules, they are unable to function and fresh rules have to be generated. Thus, a very large knowledge base must be created and stored for retrieval purposes. As discussed in Shahrestani (1995), in general, heuristic rules are hard to come up with and are always incomplete. The rules are usually inconsistent – no two experts come up with the same set.

In this work, which is an extension of author's previous works, discussed for instance in Shahrestani (1994) and Shahrestani (2005), objects are classified as members of a particular class if they possess some features which make them distinguished from other objects present in the universe of objects. The paper will also show that by making use of the distinctive features and their corresponding values, classification of all patterns, even for complex systems, can be accomplished. These are further discussed in the remainder of this paper. The proposed approach along with selection of distinctive features is discussed in the following section. This is followed by a discussion on the advantages and limits of the proposed negative recognition approach. The last section presents the conclusions.

Adaptive Classification and Recognition

The need for the utilization of artificially intelligent approaches in classification and categorization is already well established. In general, the classification process involves a combination of issues relating to amount of available information and their complexity. In this context various new requirements need to be met by categorization solutions. Some of these requirements are mentioned in this section, while some possible enabling approaches for complying with them are discussed in later parts.

The interest in building machines and systems with human-like capabilities has lead to considerable research activity and results. Important features of human capabilities that researchers are interested in implementing in artificial systems include learning, adaptability, selforganization, cognition (and recognition), reasoning, planning, decision making, action, and the like. All of which are related to intelligence. These research activities form the core of Artificial Intelligence (AI), for instance see Wilson (2007). The overwhelming amount of information that is available in most modern system environments requires new approaches to managing problems in such systems.

Many problems that may not have been traditionally resolved through classification techniques can benefit from categorization to narrow them down to more manageable sub-problems. AI techniques can help in both the categorization process. As well as resolving the resulting sub-problem. Pattern recognition is the ability to perceive structure in some data; it is one of the aspects common to all AI methods. The raw input data is pre-processed to form a pattern. A pattern is an extract of information regarding various characteristics or features of an object, state of a system, etc. Patterns either implicitly or explicitly contain names and values of features, and if they exist, relationships among features. The entire act of recognition can be carried out in two steps. In the first step a particular manifestation of an object is described in terms of suitably selected features. The second step, which is much easier than the first one, is to define and implement an unambiguous mapping of these features into class--membership space.

Patterns whose feature values are real numbers can be viewed as vectors in *n*-dimensional space, where *n* is the number of features in each pattern. With this representation, each pattern corresponds to a point in the n-dimensional metric feature space. In such a space, distance between two points indicates similarities (or differences) of the corresponding two patterns. Partitioning the feature space by any of the many available methods; e.g. maximum likelihood, K-nearest neighbors, decision surfaces and discriminate functions then carry out the actual classification.

Negative Recognition and Pattern Classification

The majority of pattern recognition methods start the classification them on the basis of similarities among objects or patterns representing them. Undoubtedly, patterns representing the same class of objects should have characteristics and at least some feature values in common. But reasonably, patterns describing members of distinct classes must have different values for one more of these features, as further discussed in author's previous work, Shahrestani (1994). In other words, objects can be categorized to be members of a particular class if they have some distinctive features that distinguish them from other objects under consideration. Consequently, the categorization process can be carried out through negative recognition that is on the basis of differences. The approach proposed in this work, is based on identifying features that are distinctive for some objects in the universe of objects being considered. To do this, objects which have some distinctive features, differences with all other objects, are put in one group.

To achieve the classification, the first step in implementing negative recognition, is identification of characteristics that distinguish each class from other classes within the training set. Such characteristics correspond to the distinctive features of the objects or patterns. In other words, objects can be classified as members of a particular class if they have some *distinctive features* making them distinguished from other objects present in the universe of objects. Consequently, it is also reasonable to start categorization on the basis of differences, or through *negative recognition*. That is, place objects or classes which have some evident differences, or distinctive features, from all other objects or classes, into one group. It should be noted that a feature that may be distinctive for a class, among a particular set of classes, is not necessarily distinctive in another set which also includes that particular class.

The main objective of this approach is to find distinctive features, or differences of each class with other classes present in the training set. If these are not evident for all classes in the set, smaller subsets are formed, so that exploration of distinctive features is facilitated. It should be noted that a feature that may be distinctive for a class, among a particular set of classes, is not necessarily distinctive in another set which also includes that particular class. Consider α_i to be the set of K_i classes under consideration, or the universe of discourse. In this universe, let the l^{th} pattern $\mathbf{p}_{al}$ representing class $C_a (\in \alpha_i)$ have n features: $x^l_{a1}, x^l_{a2}, \ldots, x^l_{an}$, whose corresponding values are: $v^l_{a1}, v^l_{a2}, \ldots, v^l_{an}$

Now, if for class C_a, represented by q_a patterns, there are m distinctive features, these distinctive features are the ones whose values are the same in all patterns belonging to the same class, and their values are different in patterns representing any other class. That is:

For j=1, 2, ..., n when the j^{th} feature is distinct:

$$x^{(dist)}_{ak} = x_{aj}, \text{ k=1,2, ..., m.} \tag{1}$$

Furthermore, these features can be found by considering all patterns representing classes in α_i and using:

For all t =1, 2, ..., q_a v^t_{aj} does not change;

and for any l =1,2,...,q_a and for each r=1,2,...,K_i where $r \neq a$:

$$v^l_{aj} \neq v^s_{rj} \text{ for all } s=1,2,\ldots,q_r \tag{2}$$

If conditions (2) are satisfied then the j^{th} feature is distinct for class C_a. Based on the identification of distinctive features an n dimensional vector, $\mathbf{m}^{i,a}$ is defined as:

$$\mathbf{m}^i_a = \{m_1, m_2, \ldots, m_n\}$$

where for j=1, 2, ...,n:

m_j=1 if the j^{th} feature is distinct, and
m_j=0 otherwise. (3)

We refer to this vector as the mask vector. Any single pattern $\mathbf{p}_{al}$, along with the mask vector $\mathbf{m}^{i,a}$, can be used to establish the necessary and sufficient conditions for pattern recognition and ascertaining if an object is a member of class C_a or not. The index i of the mask vector, is to signify that the set α_i has been used in finding it, and it will be called *mask type*. By making use of the masks and the values of the features, signified by them, class identification of new patterns can be carried out. During supervised sessions, which can be for machine knowledge base expansion, if the class of a new pattern is unidentified or misclassified (by machine), the new pattern can be added to the training set, to initiate masks updating. It will be shown that if the grouping of classes is not to be altered by this addition, to update the masks, only a fast and partial recalculation is necessary.

For the small number of trivial cases when class C_c is only one present in the set α_i, it is obvious that all the existing features in the pattern representing this single class must be considered as the distinct ones. In other words, for such a case, if representing class C_c, requires q_c patterns $\mathbf{p}_c1$, $\mathbf{p}_c$ 2, ..., $\mathbf{p}_{cq}$,, the first part of the conditions in (2) are sufficient to find the features of interest. Using those conditions, simply mean the resulting mask vector will be found as

$$\mathbf{m}^i_c = \{1,1,\ldots,1\}. \tag{4}$$

Now, if α_i consists of two classes C_a and C_b, with each represented by a number of patterns, it is easy to see that using (2) and

(3), the identification of mask vectors amount to the following

$$C_a = C_b \Rightarrow \mathbf{m}^{i,a} = \mathbf{m}^{i,b} = \mathbf{0}. \qquad (5)$$

$$\mathbf{m}^{i,a} = \mathbf{m}^{i,b} \neq \mathbf{0} \Rightarrow C_a \neq C_b. \qquad (6)$$

As far as the implementation is concerned, these mean that each pattern is contrasted with the rest of the patterns that form the training set. This way, by using relation (2), the distinctive features of that pattern can be identified. These features will then be used to form the mask vectors of type one, which in turn are utilized for further class identification purposes through relation (3). A mask of type one corresponds to the most general case. It implies that the identified features by the mask are distinctive of this particular class among *all* classes present in the training set. This gives rise to the following classification rule.

Any pattern is in the same class as the pattern in the training set, if they have the same value for any feature distinguished by the corresponding mask.

Consequently, the necessary and sufficient set of conditions that can differentiate and express all the members of any given class, can be actually described by any pattern in that class provided its mask vector can be found.

If each and every pattern cannot be associated with a mask of type one, higher type masks will need to be established. To find type two masks, first the patterns representing classes with type one masks will be eliminated from the training set. After that the same process as the one for finding masks of type one is repeated on this new smaller training set. The classification rule will then be as follows.

If the pattern is not a member of any class with a type one mask, then it is in the same class as the pattern in the training set, if they have the same value for any feature distinguished by the mask (of type two).

The whole process is repeated until classification rules for all classes present in the training set have been found. As described previously, this amounts to establishing a mask of some type for all of classes in the original training set.

Implementation Considerations

The proposed approach and its language work in a general manner. Consequently, the corresponding codes can be developed and utilized as a general adaptive pattern recognition scheme. This has clear advantages, as the suitability and performance of the approach and the related algorithms can be tested using previously evaluated data, and then be used in many other domains. The proposed approach appears to be applicable to many classification problems encountered in complex systems that may be solved by artificial intelligence techniques. This approach does not depend on heuristics that may result in partial rules. Consequently, for a broad range of cases, only the firing of a single rule will set the necessary and sufficient conditions for establishing class membership of an object.

Only two rather straightforward routines for input-output manipulations are required for any new situation. The inputs file manager will normally get the data output from another system. Depending on the nature of the problem, the inputs file handling system may also need to do some data management itself. It will then do the appropriate formatting of the features, to get the patterns into their required format. It will then write them into an intermediary file, which will in turn be read by the actual classification. As the proposed approach easily furnishes for it, basic machine procedures form the core decision making routines. This results in a fast and reliable code. The output file manager gets the result from the intermediary file, and simply converts it into a suitable form for the required action – print out, corrective action, and so forth.

Although the proposed methodology is rather straight forward, but many different approaches can be taken for its implementation as a code. Finding proper values and types for the masks is the core of the approach described in this work. So, the emphasis of the code should be on finding these values reliably and quickly. For example given a new pattern, there *will* be some mask changes if the pattern cannot be identified with previously found masks. But even so, the masks *might* retain their previous types. Consequently, execution time may be reduced, if the new masks can be found within the previous grouping of classes. This will be important, for cases where the number of classes and/or patterns is very large, as with general cases in complex systems.

Relation (2) shows that the mask for the class that the new pattern is a member of will change. As the new pattern has not been recognized, some of the features must have different values from the previous ones. On the other hand, provided the mask types are not altered, using the same relation is rather trivial. That is, to find the new mask, it is just necessary to eliminate from the mask vector, the features whose values have changed between the new and any previously encountered pattern in this class. For other classes the masks might change by virtue of the second part of relation (2). Again any pattern in any particular class along with the new pattern can be used for that test, and any feature whose value is the same can be eliminated from the previous mask to arrive at the new mask for this class. Of course if any of the mask vectors become null in the process, all masks must be recalculated. As the mask vectors becoming null implies that some mask types, and grouping of classes, are required to be modified, the complete recalculation of masks becomes necessary.

The proposed approach and the code based on it can act in a general way to recognition of patterns adaptively. The required knowledge base is compact, the language used is quite general, no heuristic approach is necessary, and the code is fast and reliable. Only two rather simple routines for I/O manipulations are needed for any new job. The input-file manager gets the data output of another system. It may also do some data manipulation itself, depending on the nature of the problem. It will then do the proper packing of the features, to get the patterns into their required format, and writes them into an intermediary file, which will in turn be read by the actual code. As the proposed approach conveniently allows it, basic machine operations are the main decision making routines, which results in a fast code. The output file manager gets the result from another intermediary file, and simply converts it into a suitable form for the required action – print out, corrective action, and so forth.

Training Results and Discussions

If the training set, the first one or the ones after some time of code implementation, contains large number of noisy patterns in a given class, it may happen that the mask type for one of the similar classes gets improper assignment. Similar classes will have patterns which will resemble each other very closely. The dynamic status of the system may contain valuable information, and they can easily be implemented in this approach, if the need arises. Generally speaking, inclusion of such information, will lead to longer patterns.

An important issue to consider in categorization relates to the fact that in many situations, it is possible that different objects have similar representative patterns. In a very noisy environment the representation of objects, may even result in patterns that are not evidently distinct. In such situations, for the case of similar but distinct patterns, a possible solution may consist of taking only one pattern from such a class, and keep those other rare patterns, unidentifiable with the resulting mask, in the knowledge base and recognize their class membership by a table look up.

Another solution may be to combine these classes and associate a mask (and a class number) with both of them, and to identify them by a table look up. When the patterns are not distinct, actually the same pattern is representing two classes, so even a supervisor will need some other information to identify the class membership of the pattern.

The approach will actually remove one of these two classes from the set α_i to find a mask of type i, for the other class. Then along with other classes whose masks are found, this class is also eliminated from the set of classes under consideration, and a mask for the other class is found.

This mask will be of a type greater than i. In cases like this, all of the classification rules should be fired. In other words, conflicts should be allowed to rise. Conflicts may arise when the (new) pattern can be identified as the member of two classes with two different mask types. As a single distinctive feature is enough to establish class membership, for distinct patterns, conflicts can be resolved by other features which are distinct between the conflicting classes.

Conclusions

In this work, a fast and reliable method for categorization of patterns that may be encountered in complex systems is described. The proposed method and consequently the code that is based on it can act as comprehensive techniques for object classification and adaptive pattern recognition. The proposed approach and the scheme based on it can act as a general decision support system for classification in complex systems.

The required knowledge base is compact and the language used is quite general. The approach proposed in this work, has the clear advantage of requiring only a single rule for achieving the required recognition and hence establishing the class membership of a given pattern. That is because, in contrast to most classification systems, which depend on heuristic rules, this approach does not resort to any heuristic rule.

References

Basri, R. (1993). "Recognition by Prototypes," Proceedings of IEEE International Conf. on Computer Vision and Pattern Recognition, 1993, 161–167.

Commike, Y. & Hull, J. (1991). "Syntactic Pattern Classification by Branch and Bound Search," Proceedings of IEEE International Conf. on Computer Vision and Pattern Recognition, 1991, 432–437.

Jin; C., Pok, G., Kim, H., Cha, E. & Ryu, K. (2010). "Classification of Closed Frequent Patterns Improved by Feature Space Transformation," Proceedings IEEE 10th International Conference on Computer and Information Technology (CIT). 29 June- 1 July 2010, Bradford, 1306 – 1311.

Kurosawa, Y. (2008). "Incremental learning for feature extraction filter mask used in similar pattern classification," Proceedings IEEE International Joint Conference on Neural Networks, 1-8 June 2008, Hong Kong, 497 – 504.

Lee, Y. (1988). Evolution, Learning and Cognition, *World Scientific Publishing Co*.

Long, H. & Wang, X. (2008). "Aircraft fuel system diagnostic fault detection through expert system", Proceedings IEEE Sixth World Congress on Intelligent Control and Automation, WCICA2008, 7104-7107.

Mourot, G., Bouschiri, S. & Ragot, J. (1993). "Pattern Recognition for Diagnosis of Technological Systems: A Review," Proceedings of IEEE International Conf. on Systems, Man and Cybernetics, vol. 5, 1993, 275–281.

Pao, Y. (1989) Adaptive Pattern Recognition and Neural Networks, *Addison-Wesl*ey.

Shahrestani, S., Yee, H. & Ypsilantis, J. (1995) "Adaptive Recognition by Specialized Grouping of Classes," Proceedings 4th IEEE Conference on Control Applications, 1995, Albany, NY, 1995, 637-642.

Shahrestani, S. (2005). "Exploiting Structure and Concepts for Enhanced Management of Information and Networked Systems," WSEAS Trans. on Computers, 7 (4). 769-777.

Shahrestani, S. (2009). "Complex System Diagnosis through Adaptive Recognition," Proceedings 11th International Conf. on Mathematical Methods, Computational Techniques and Intelligent Systems, MAMECTIS09, 2009, Canary Islands, Spain, 133-138.

Therrein, C. (1989). Decision Estimation and Classification, John Wiley and Sons.

Ting Wang, T. & Wang, W. (2010). "Pattern classification with ordered features using mRMR and neural networks," Proceedings IEEE International Conference on Information Networking and Automation (ICINA). 18-19 Oct. 2010, Kunming, China, 128-131.

Todd, M., McArthur, S., McDonald, J. & Shaw, S. (2007). "A Semiautomatic Approach to Deriving Turbine Generator Diagnostic Knowledge," *IEEE Trans*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, 37 (5). 979-992.

Vaucher, S., Sahraoui, H. & Vaucher, J. (2008). "Discovering New Change Patterns in Object-Oriented Systems," Proceedings IEEE International Conf. Reverse Engineering, 2008, 37-41.

Wilkes, D. & Tsotsos, J. (1993). "Efficient Serial Associative Memory," Proceedings of IEEE International Conf. on Computer Vision and Pattern Recognition, 1993, 701–702.

Wilson, A., McNamara, L. & Wilson, G. (2007). "Information integration for complex systems," *Reliability Engineering & System Safety*, 92 (1). 121-130.

Zhao, W., Bai, X., Wang, W. & Ding, J. (2005). "A Novel Alarm Processing and Fault Diagnosis Expert System Based on BNF Rules," Proceedings IEEE International Transmission and Distribution Conf. and Exhibition, 2005, 1-6.

Zheng, X., Wang, Z. & Qian, F. (2006). "An Expert System for Real-time Fault Diagnosis and Its Application in PTA Process," Proceedings IEEE Sixth World Congress on Intelligent Control and Automation, WCICA 2006, 5623-5627.