Chapter 46

DECOMPOSITION METHODOLOGY FOR KNOWLEDGE DISCOVERY AND DATA MINING

Oded Maimon
Department of Industrial Engineering
Tel-Aviv University
maimon@eng.tau.ac.il

Lior Rokach
Department of Industrial Engineering
Tel-Aviv University
liorr@eng.tau.ac.il

Abstract The idea of decomposition methodology is to break down a complex Data Mining task into several smaller, less complex and more manageable, sub-tasks that are solvable by using existing tools, then joining their solutions together in order to solve the original problem. In this chapter we provide an overview of decomposition methods in classification tasks with emphasis on elementary decomposition methods. We present the main properties that characterize various decomposition frameworks and the advantages of using these framework. Finally we discuss the uniqueness of decomposition methodology as opposed to other closely related fields, such as ensemble methods and distributed data mining.

Keywords: Decomposition, Mixture-of-Experts, Elementary Decomposition Methodology, Function Decomposition, Distributed Data Mining, Parallel Data Mining

1. Introduction

One of the explicit challenges in Data Mining is to develop methods that will be feasible for complicated real-world problems. In many disciplines, when a problem becomes more complex, there is a natural tendency to try to break it down into smaller, distinct but connected pieces. The concept of breaking

down a system into smaller pieces is generally referred to as *decomposition*. The purpose of decomposition methodology is to break down a complex problem into smaller, less complex and more manageable, sub-problems that are solvable by using existing tools, then joining them together to solve the initial problem. Decomposition methodology can be considered as an effective strategy for changing the representation of a classification problem. Indeed, Kusiak (2000) considers decomposition as the "most useful form of transformation of data sets".

The decomposition approach is frequently used in statistics, operations research and engineering. For instance, decomposition of time series is considered to be a practical way to improve forecasting. The usual decomposition into trend, cycle, seasonal and irregular components was motivated mainly by business analysts, who wanted to get a clearer picture of the state of the economy (Fisher, 1995). Although the operations research community has extensively studied decomposition methods to improve computational efficiency and robustness, identification of the partitioned problem model has largely remained an ad hoc task (He *et al*., 2000).

In engineering design, problem decomposition has received considerable attention as a means of reducing multidisciplinary design cycle time and of streamlining the design process by adequate arrangement of the tasks (Kusiak *et al*., 1991). Decomposition methods are also used in decision-making theory. A typical example is the AHP method (Saaty, 1993). In artificial intelligence finding a good decomposition is a major tactic, both for ensuring the transparent end-product and for avoiding a combinatorial explosion (Michie, 1995).

Research has shown that no single learning approach is clearly superior for all cases. In fact, the task of discovering regularities can be made easier and less time consuming by decomposition of the task. However, decomposition methodology has not attracted as much attention in the KDD and machine learning community (Buntine, 1996).

Although decomposition is a promising technique and presents an obviously natural direction to follow, there are hardly any works in the Data Mining literature that consider the subject directly. Instead, there are abundant practical attempts to apply decomposition methodology to specific, real life applications (Buntine, 1996). There are also many discussions on closely related problems, largely in the context of distributed and parallel learning (Zaki and Ho, 2000) or ensembles classifiers (see Chapter 45 in this volume). Nevertheless, there are a few important works that consider decomposition methodology directly. Various decomposition methods have been presented (Kusiak, 2000). There was also suggestion to decompose the exploratory data analysis process into 3 parts: *model search*, *pattern search*, and *attribute search* (Bhargava, 1999). However, in this case the notion of "decomposition" refers to the entire KDD process, while this chapter focuses on decomposition of the model search.

In the neural network community, several researchers have examined the decomposition methodology (Hansen, 2000). The *"mixture-of-experts"* (ME) method decomposes the input space, such that each expert examines a different part of the space (Nowlan and Hinton, 1991). However, the sub-spaces have soft "boundaries", namely sub-spaces are allowed to overlap. Figure 46.1 illustrates an n-expert structure. Each expert outputs the conditional probability of the target attribute given the input instance. A gating network is responsible for combining the various experts by assigning a weight to each network. These weights are not constant but are functions of the input instance x.

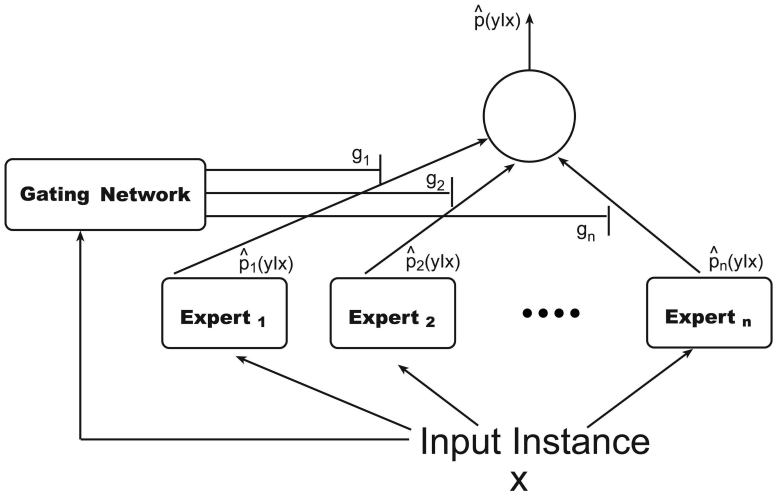

Figure 46.1. Illustration of n-Expert Structure.

An extension to the basic mixture of experts, known as hierarchical mixtures of experts (HME), has been proposed by Jordan and Jacobs (1994). This extension decomposes the space into sub-spaces, and then recursively decomposes each sub-space to sub-spaces.

Variation of the basic mixtures of experts methods have been developed to accommodate specific domain problems. A specialized modular network called the Meta-p_i network has been used to solve the vowel-speaker problem (Hampshire and Waibel, 1992; Peng *et al.*, 1995). There have been other extensions to the ME such as nonlinear gated experts for time-series (Weigend *et al.*, 1995); revised modular network for predicting the survival of AIDS patients (Ohno-Machado and Musen, 1997); and a new approach for combining multiple experts for improving handwritten numerals recognition (Rahman and Fairhurst, 1997).

However, none of these works presents a complete framework that considers the coexistence of different decomposition methods, namely: when we should prefer a specific method and whether it is possible to solve a given problem using a hybridization of several decomposition methods.

2. Decomposition Advantages

2.1 Increasing Classification Performance (Classification Accuracy)

Decomposition methods can improve the predictive accuracy of regular methods. In fact Sharkey (1999) argues that improving performance is the main motivation for decomposition. Although this might look surprising at first, it can be explained by the bias-variance tradeoff. Since decomposition methodology constructs several simpler sub-models instead a single complicated model, we might gain better performance by choosing the appropriate sub-models' complexities (i.e. finding the best bias-variance tradeoff). For instance, a single decision tree that attempts to model the entire instance space usually has high variance and small bias. On the other hand, Naïve Bayes can be seen as a composite of single-attribute decision trees (each one of these trees contains only one unique input attribute). The bias of the Naïve Bayes is large (as it can not represent a complicated classifier); on the other hand, its variance is small. Decomposition can potentially obtain a set of decision trees, such that each one of the trees is more complicated than a single-attribute tree (thus it can represent a more complicated classifier and it has lower bias than the Naïve Bayes) but not complicated enough to have high variance.

There are other justifications for the performance improvement of decomposition methods, such as the ability to exploit the specialized capabilities of each component, and consequently achieve results which would not be possible in a single model. An excellent example to the contributions of the decomposition methodology can be found in Baxt (1990). In this research, the main goal was to identify a certain clinical diagnosis. Decomposing the problem and building two neural networks significantly increased the correct classification rate.

2.2 Scalability to Large Databases

One of the explicit challenges for the KDD research community is to develop methods that facilitate the use of Data Mining algorithms for real-world databases. In the information age, data is automatically collected and therefore the database available for mining can be quite large, as a result of an increase in the number of records in the database and the number of fields/attributes in each record (high dimensionality).

There are many approaches for dealing with huge databases including: sampling methods; massively parallel processing; efficient storage methods; and dimension reduction. Decomposition methodology suggests an alternative way to deal with the aforementioned problems by reducing the volume of data to be processed at a time. Decomposition methods break the original problem into several sub-problems, each one with relatively small dimensionality. In this way, decomposition reduces training time and makes it possible to apply standard machine-learning algorithms to large databases (Sharkey, 1999).

2.3 Increasing Comprehensibility

Decomposition methods suggest a conceptual simplification of the original complex problem. Instead of getting a single and complicated model, decomposition methods create several sub-models, which are more comprehensible. This motivation has often been noted in the literature (Pratt *et al*., 1991; Hrycej, 1992; Sharkey, 1999). Smaller models are also more appropriate for user-driven Data Mining that is based on *visualization techniques*. Furthermore, if the decomposition structure is induced by automatic means, it can provide new insights about the explored domain.

2.4 Modularity

Modularity eases the maintenance of the classification model. Since new data is being collected all the time, it is essential once in a while to execute a rebuild process to the entire model. However, if the model is built from several sub-models, and the new data collected affects only part of the sub-models, a more simple re-building process may be sufficient. This justification has often been noted (Kusiak, 2000).

2.5 Suitability for Parallel Computation

If there are no dependencies between the various sub-components, then parallel techniques can be applied. By using parallel computation, the time needed to solve a mining problem can be shortened.

2.6 Flexibility in Techniques Selection

Decomposition methodology suggests the ability to use different inducers for individual sub-problems or even to use the same inducer but with a different setup. For instance, it is possible to use neural networks having different topologies (different number of hidden nodes). The researcher can exploit this freedom of choice to boost classifier performance.

The first three advantages are of particular importance in commercial and industrial Data Mining. However, as it will be demonstrated later, not all decomposition methods display the same advantages.

3. The Elementary Decomposition Methodology

Finding an optimal or quasi-optimal decomposition for a certain supervised learning problem might be hard or impossible. For that reason Rokach and Maimon (2002) proposed *elementary decomposition methodology*. The basic idea is to develop a meta-algorithm that recursively decomposes a classification problem using elementary decomposition methods. We use the term "elementary decomposition" to describe a type of simple decomposition that can be used to build up a more complicated decomposition. Given a certain problem, we first select the most appropriate elementary decomposition to that problem. A suitable decomposer then decomposes the problem, and finally a similar procedure is performed on each sub-problem. This approach agrees with the "no free lunch theorem", namely if one decomposition is better than another in some domains, then there are necessarily other domains in which this relationship is reversed.

For implementing this decomposition methodology, one might consider the following issues:

- What type of elementary decomposition methods exist for classification inducers?
- Which elementary decomposition type performs best for which problem? What factors should one take into account when choosing the appropriate decomposition type?
- Given an elementary type, how should we infer the best decomposition structure automatically?
- How should the sub-problems be re-composed to represent the original concept learning?
- How can we utilize prior knowledge for improving decomposing methodology?

Figure 46.2 suggests an answer to the first issue. This figure illustrates a novel approach for arranging the different elementary types of decomposition in supervised learning (Maimon and Rokach, 2002).

In *intermediate concept* decomposition, instead of inducing a single complicated classifier, several sub-problems with different and more simple concepts are defined. The intermediate concepts can be based on an aggregation of the

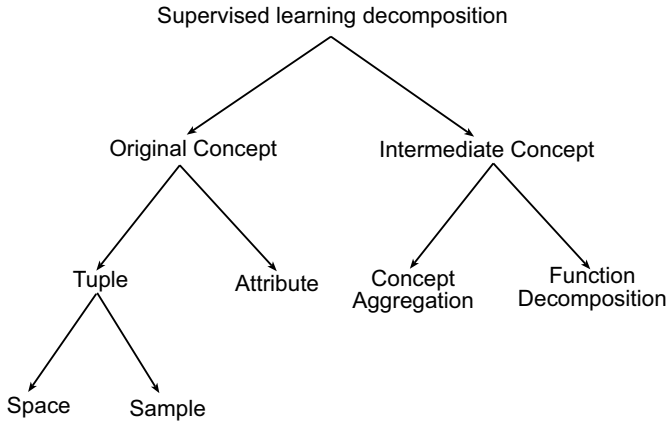

Figure 46.2. Elementary Decomposition Methods in Classification.

original concept's values (*concept aggregation*) or not (*function decomposition*).

Classical concept aggregation replaces the original target attribute with a function, such that the domain of the new target attribute is smaller than the original one.

Concept aggregation has been used to classify free text documents into predefined topics (Buntine, 1996). This paper suggests breaking the topics up into groups (co-topics). Instead of predicting the document's topic directly, the document is first classified into one of the co-topics. Another model is then used to predict the actual topic in that co-topic.

A general concept aggregation algorithm called *Error-Correcting Output Coding* (ECOC) which decomposes multi-class problems into multiple, two-class problems has been suggested by Dietterich and Bakiri (1995). A classifier is built for each possible binary partition of the classes. Experiments show that ECOC improves the accuracy of neural networks and decision trees on several multi-class problems from the UCI repository.

The idea to decompose a K class classification problems into K two class classification problems has been proposed by Anand *et al*. (1995). Each problem considers the discrimination of one class to the other classes. Lu and Ito (1999) extend the last method and propose a new method for manipulating the data based on the class relations among training data. By using this method, they divide a K class classification problem into a series of $K(K-1)/2$ two-class problems where each problem considers the discrimination of one class

to each one of the other classes. They have examined this idea using neural networks.

Fürnkranz (2002) studied the round-robin classification problem (pairwise classification), a technique for handling multi-class problems, in which one classifier is constructed for each pair of classes. Empirical study has showed that this method can potentially improve classification accuracy.

Function decomposition was originally developed in the Fifties and Sixties for designing switching circuits. It was even used as an evaluation mechanism for checker playing programs (Samuel, 1967). This approach was later improved by Biermann *et al*. (1982). Recently, the machine-learning community has adopted this approach. Michie (1995) used a manual decomposition of the problem and an expert-assisted selection of examples to construct rules for the concepts in the hierarchy. In comparison with standard decision tree induction techniques, structured induction exhibits about the same degree of classification accuracy with the increased transparency and lower complexity of the developed models. Zupan *et al*. (1998) presented a general-purpose function decomposition approach for machine-learning. According to this approach, attributes are transformed into new concepts in an iterative manner and create a hierarchy of concepts. Recently, Long (2003) has suggested using a different function decomposition known as bi-decomposition and shows it applicability in data mining.

Original Concept decomposition means dividing the original problem into several sub-problems by partitioning the training set into smaller training sets. A classifier is trained on each sub-sample seeking to solve the original problem. Note that this resembles ensemble methodology but with the following distinction: each inducer uses only a portion of the original training set and ignores the rest. After a classifier is constructed for each portion separately, the models are combined in some fashion, either at learning or classification time.

There are two obvious ways to break up the original dataset: tuple-oriented or attribute (feature) oriented. Tuple decomposition by itself can be divided into two different types: sample and space. In sample decomposition (also known as partitioning), the goal is to partition the training set into several sample sets, such that each sub-learning task considers the entire space.

In space decomposition, on the other hand, the original instance space is divided into several sub-spaces. Each sub-space is considered independently and the total model is a (possibly soft) union of such simpler models.

Space decomposition also includes the divide and conquer approaches such as mixtures of experts, local linear regression, CART/MARS, adaptive subspace models, etc., (Johansen and Foss, 1992; Jordan and Jacobs, 1994; Ramamurti and Ghosh, 1999; Holmstrom *et al.*, 1997).

Feature set decomposition (also known as attribute set decomposition) generalizes the task of feature selection which is extensively used in Data Mining. Feature selection aims to provide a representative set of features from which a classifier is constructed. On the other hand, in feature set decomposition, the original feature set is decomposed into several subsets. An inducer is trained upon the training data for each subset independently, and generates a classifier for each one. Subsequently, an unlabeled instance is classified by combining the classifications of all classifiers. This method potentially facilitates the creation of a classifier for high dimensionality data sets because each sub-classifier copes with only a projection of the original space.

In the literature there are several works that fit the feature set decomposition framework. However, in most of the papers the decomposition structure was obtained ad-hoc using prior knowledge. Moreover, as a result of a literature review, Ronco *et al*. (1996) have concluded that "*There exists no algorithm or method susceptible to perform a vertical self-decomposition without a-priori knowledge of the task!*". Bay (1999) presented a feature set decomposition algorithm known as MFS which combines multiple nearest neighbor classifiers, each using only a subset of random features. Experiments show MFS can improve the standard nearest neighbor classifiers. This procedure resembles the well-known bagging algorithm (Breiman, 1996). However, instead of sampling instances with replacement, it samples features without replacement.

Another feature set decomposition was proposed by Kusiak (2000). In this case, the features are grouped according to the attribute type: nominal value features, numeric value features and text value features. A similar approach was used by Gama (2000) for developing the linear-bayes classifier. The basic idea consists of aggregating the features into two subsets: the first subset containing only the nominal features and the second subset only the continuous features.

An approach for constructing an ensemble of classifiers using rough set theory was presented by Hu (2001). Although Hu's work refers to ensemble methodology and not decomposition methodology, it is still relevant for this case, especially as the declared goal was to construct an ensemble such that different classifiers use different attributes as much as possible. According to Hu, diversified classifiers lead to uncorrelated errors, which in turn improve classification accuracy. The method searches for a set of reducts, which include all the indispensable attributes. A reduct represents the minimal set of attributes which has the same classification power as the entire attribute set.

In another research, Tumer and Ghosh (1996) propose decomposing the feature set according to the target class. For each class, the features with low correlation relating to that class have been removed. This method has been applied on a feature set of 25 sonar signals where the target was to identify the meaning of the sound (whale, cracking ice, etc.). Cherkauer (1996) used fea-

ture set decomposition for radar volcanoes recognition. Cherkauer manually decomposed a feature set of 119 into 8 subsets. Features that are based on different image processing operations were grouped together. As a consequence, for each subset, four neural networks with different sizes were built. Chen *et al*. (1997) proposed a new combining framework for feature set decomposition and demonstrate its applicability in text-independent speaker identification. Jenkins and Yuhas (1993) manually decomposed the features set of a certain truck backer-upper problem and reported that this strategy has important advantages.

A paradigm, termed co-training, for learning with labeled and unlabeled data was proposed in Blum and Mitchell (1998). This paradigm can be considered as a feature set decomposition for classifying Web pages, which is useful when there is a large data sample, of which only a small part is labeled. In many applications, unlabeled examples are significantly easier to collect than labeled ones. This is especially true when the labeling process is time-consuming or expensive, such as in medical applications. According to the co-training paradigm, the input space is divided into two different views (i.e. two independent and redundant sets of features). For each view, Blum and Mitchell built a different classifier to classify unlabeled data. The newly labeled data of each classifier is then used to retrain the other classifier. Blum and Mitchell have shown, both empirically and theoretically, that unlabeled data can be used to augment labeled data.

More recently, Liao and Moody (2000) presented another option to a decomposition technique whereby all input features are initially grouped by using a hierarchical clustering algorithm based on pairwise mutual information, with statistically similar features assigned to the same group. As a consequence, several feature subsets are constructed by selecting one feature from each group. A neural network is subsequently constructed for each subset. All netwroks are then combined.

In the statistics literature, the most well-known decomposition algorithm is the MARS algorithm (Friedman, 1991). In this algorithm, a multiple regression function is approximated using linear splines and their tensor products. It has been shown that the algorithm performs an ANOVA decomposition, namely the regression function is represented as a grand total of several sums. The first sum is of all basic functions that involve only a single attribute. The second sum is of all basic functions that involve exactly two attributes, representing (if present) two-variable interactions. Similarly, the third sum represents (if present) the contributions from three-variable interactions, and so on.

Other works on feature set decomposition have been developed by extending the Naïve Bayes classifier. The Naïve Bayes classifier (Domingos and Pazzani, 1997) uses the Bayes' rule to compute the conditional probability of each pos-

sible class, assuming the input features are conditionally independent given the target feature. Due to the conditional independence assumption, this method is called "Naïve". Nevertheless, a variety of empirical researches show surprisingly that the Naïve Bayes classifier can perform quite well compared to other methods, even in domains where clear feature dependencies exist (Domingos and Pazzani, 1997). Furthermore, Naïve Bayes classifiers are also very simple and easy to understand (Kononenko, 1990).

Both Kononenko (1991) and Domingos and Pazzani (1997), suggested extending the Naïve Bayes classifier by finding the single best pair of features to join by considering all possible joins. Kononenko (1991) described the semi-Naïve Bayes classifier that uses a conditional independence test for joining features. Domingos and Pazzani (1997) used estimated accuracy (as determined by leave–one–out cross-validation on the training set). Friedman *et al*. (1997) have suggested the tree augmented Naïve Bayes classifier (TAN) which extends the Naïve Bayes, taking into account dependencies among input features. The selective Bayes Classifier (Langley and Sage, 1994) preprocesses data using a form of feature selection to delete redundant features. Meretakis and Wthrich (1999) introduced the large Bayes algorithm. This algorithm employs an *a-priori*-like frequent pattern-mining algorithm to discover frequent and interesting features in subsets of arbitrary size, together with their class probability estimation.

Recently Maimon and Rokach (2005) suggested a general framework that searches for helpful feature set decomposition structures. This framework nests many algorithms, two of which are tested empirically over a set of benchmark datasets. The first algorithm performs a serial search while using a new Vapnik-Chervonenkis dimension bound for multiple oblivious trees as an evaluating schema. The second algorithm performs a multi-search while using wrapper evaluating schema. This work indicates that feature set decomposition can increase the accuracy of decision trees.

It should be noted that some researchers prefer the terms "horizontal decomposition" and "vertical decomposition" for describing "space decomposition" and "attribute decomposition" respectively (Ronco *et al*., 1996).

4. The Decomposer's Characteristics

4.1 Overview

The following sub-sections present the main properties that characterize decomposers. These properties can be useful for differentiating between various decomposition frameworks.

4.2 The Structure Acquiring Method

This important property indicates how the decomposition structure is obtained:

- Manually (explicitly) based on an expert's knowledge in a specific domain (Blum and Mitchell, 1998; Michie, 1995). If the origin of the dataset is a relational database, then the schema's structure may imply the decomposition structure.
- Predefined due to some restrictions (as in the case of distributed Data Mining)
- Arbitrarily (Domingos, 1996; Chan and Stolfo, 1995) - The decomposition is performed without any profound thought. Usually, after setting the size of the subsets, members are randomly assigned to the different subsets.
- Induced without human interaction by a suitable algorithm (Zupan *et al.*, 1998).

Some may justifiably claim that searching for the best decomposition might be time-consuming, namely prolonging the Data Mining process. In order to avoid this disadvantage, the complexity of the decomposition algorithms should be kept as small as possible. However, even if this cannot be accomplished, there are still important advantages, such as better comprehensibility and better performance that makes decomposition worth the additional computational complexity.

Furthermore, it should be noted that in an ongoing Data Mining effort (like in a churning application) searching for the best decomposition structure might be performed in wider time buckets (for instance, once a year) than when training the classifiers (for instance once a week). Moreover, for acquiring decomposition structure, only a relatively small sample of the training set may be required. Consequently, the execution time of the decomposer will be relatively small compared to the time needed to train the classifiers.

Ronco *et al.* (1996) suggest a different categorization in which the first two categories are referred as "ad-hoc decomposition" and the last two categories as "self-decomposition".

Usually in real-life applications the decomposition is performed manually by incorporating business information into the modeling process. For instance Berry and Linoff (2000) provide a practical example in their book saying:

> It may be known that platinum cardholders behave differently from gold cardholders. Instead of having a Data Mining technique figure this out, give it the hint by building separate models for the platinum and gold cardholders.

Berry and Linoff (2000) state that decomposition can be also useful for handling missing data. In this case they do not refer to sporadic missing data but to the case where several attribute values are available for some tuples but not for all of them. For instance: "Historical data, such as billing information, is available only for customers who have been around for a sufficiently long time" or "Outside data, such as demographics, is available only for the subset of the customer base that matches"). In this case, one classifier can be trained for customers having all the information and a second classifier for the remaining customers.

4.3 The Mutually Exclusive Property

This property indicates whether the decomposition is mutually exclusive (*disjointed decomposition*) or partially overlapping (i.e. a certain value of a certain attribute in a certain tuple is utilized more than once). For instance, in the case of sample decomposition, "mutually exclusive" means that a certain tuple cannot belong to more than one subset (Domingos, 1996; Chan and Stolfo, 1995). Bay (1999), on the other hand, has used non-exclusive feature decomposition.

Similarly CART and MARS perform mutually exclusive decomposition of the input space, while HME allows sub-spaces to overlap.

Mutually exclusive decomposition can be deemed as a *pure* decomposition. While pure decomposition forms a restriction on the problem space, it has some important and helpful properties:

- A greater tendency in reduction of execution time than non-exclusive approaches. Since most learning algorithms have computational complexity that is greater than linear in the number of attributes or tuples, partitioning the problem dimensionality in a mutually exclusive manner means a decrease in computational complexity (Provost and Kolluri, 1997).

- Since mutual exclusiveness entails using smaller datasets, the models obtained for each sub-problem are smaller in size. Without the mutually exclusive restriction, each model can be as complicated as the model obtained for the original problem. Smaller models contribute to comprehensibility and ease in maintaining the solution.

- According to Bay (1999), mutually exclusive decomposition may help avoid some error correlation problems that characterize non-mutually exclusive decompositions. However, Sharkey (1999) argues that mutually exclusive training sets do not necessarily result in low error correlation. This point is true when each sub-problem is representative (i.e. represent the entire problem, as in sample decomposition).

- Reduced tendency to contradiction between sub-models. When a mutually exclusive restriction is unenforced, different models might generate contradictive classifications using the same input. Reducing inter-models contraindications help us to grasp the results and to combine the sub-models into one model. Ridgeway *et al.* (1999), for instance, claim that the resulting predictions of ensemble methods are usually inscrutable to end-users, mainly due to the complexity of the generated models, as well as the obstacles in transforming theses models into a single model. Moreover, since these methods do not attempt to use all relevant features, the researcher will not obtain a complete picture of which attribute actually affects the target attribute, especially when, in some cases, there are many relevant attributes.
- Since the mutually exclusive approach encourages smaller datasets, they are more feasible. Some Data Mining tools can process only limited dataset size (for instance when the program requires that the entire dataset will be stored in the main memory). The mutually exclusive approach can make certain that Data Mining tools are fairly scalable to large data sets (Chan and Stolfo, 1997; Provost and Kolluri, 1997).
- We claim that end-users can grasp mutually exclusive decomposition much easier than many other methods currently in use. For instance, boosting, which is a well-known ensemble method, distorts the original distribution of instance space, a fact that non-professional users find hard to grasp or understand.

4.4 The Inducer Usage

This property indicates the relation between the decomposer and the inducer used. Some decomposition implementations are "inducer-free", namely they do not use intrinsic inducers at all. Usually the decomposition procedure needs to choose the best decomposition structure among several structures that it considers. In order to measure the performance of a certain decomposition structure, there is a need to realize the structure by building a classifier for each component. However since "inducer-free" decomposition does not use any induction algorithm, it uses a frequency table of the Cartesian product of the feature values instead. Consider the following example. The training set consists of four binary input attributes (a_1, a_2, a_3, a_4) and one target attribute (y). Assume that an "inducer-free" decomposition procedure examines the following feature set decomposition: (a_1, a_3) and (a_2, a_4). In order to measure the classification performance of this structure, it is required to build two classifiers; one classifier for each subset. In the absence of an induction algorithm, two frequency tables are built; each table has $2^2 = 4$ entries representing the Cartesian product of the attributes in each subset. For each entry in the table,

we measure the frequency of the target attribute. Each one of the tables can be separately used to classify a new instance x: we search for the entry that corresponds to the instance x and select the target value with the highest frequency in that entry. This "inducer-free" strategy has been used in several places. For instance the extension of Naïve Bayes suggested by Domingos and Pazzani (1997), can be considered as a feature set decomposition with no intrinsic inducer. Zupan *et al*. (1998) have developed the function decomposition by using sparse frequency tables.

Other implementations are considered as an "inducer-dependent" type, namely these decomposition methods use intrinsic inducers, and they have been developed specifically for a certain inducer. They do not guarantee effectiveness in any other induction method. For instance, the work of Lu and Ito (1999) was developed specifically for neural networks.

The third type of decomposition method is the "inducer-independent" type. These implementations can be performed on any given inducer, however, the same inducer is used in all subsets. As opposed to the "inducer-free" implementation, which does not use any inducer for its execution, "inducer-independent" requires the use of an inducer. Nevertheless, it is not limited to a specific inducer like the "inducer-dependent".

The last type is the "inducer-chooser" type, which, given a set of inducers, the system uses the most appropriate inducer on each sub-problem.

4.5 Exhaustiveness

This property indicates whether all data elements should be used in the decomposition. For instance, an exhaustive feature set decomposition refers to the situation in which each feature participates in at least one subset.

4.6 Combiner Usage

This property specifies the relation between the decomposer and the combiner. Some decomposers are combiner-dependent. That is to say they have been developed specifically for a certain combination method like voting or Naïve Bayes. For additional combining methods see Chapter 45 in this volume. Other decomposers are combiner-independent; the combination method is provided as input to the framework. Potentially there could be decomposers that, given a set of combiners, would be capable of choosing the best combiner in the current case.

4.7 Sequentially or Concurrently

This property indicates whether the various sub-classifiers are built sequentially or concurrently. In sequential framework the outcome of a certain classifier may effect the creation of the next classifier. On the other hand, in con-

current framework each classifier is built independently and their results are combined in some fashion. Sharkey (1996) refers to this property as "The relationship between modules" and distinguishes between three different types: successive, cooperative and supervisory. Roughly speaking the "successive" refers to "sequential" while "cooperative" refers to "concurrent". The last type applies to the case in which one model controls the other model. Sharkey (1996) provides an example in which one neural network is used to tune another neural network.

The original problem in *intermediate concept decomposition* is usually converted to a sequential list of problems, where the last problem aims to solve the original one. On the other hand, in *original concept decomposition* the problem is usually divided into several sub-problems which exist on their own. Nevertheless, there are some exceptions. For instance, Quinlan (1993) proposed an original concept framework known as "windowing" that is considered to be sequential. For other examples the reader is referred to Chapter 45 in this volume.

Naturally there might be other important properties which can be used to differentiate a decomposition scheme. Table 46.1 summarizes the most relevant research performed on each decomposition type.

Table 46.1. Summary of Decomposition Methods in the Literature.

Paper	Decomposition Type	Mutually Exclusive	Structure Acquiring Method
(Anand *et al*., 1995)	Concept	No	Arbitrarily
(Buntine, 1996)	Concept	Yes	Manually
(Michie, 1995)	Function	Yes	Manually
(Zupan *et al*., 1998)	Function	Yes	Induced
(Ali and Pazzani, 1996)	Sample	No	Arbitrarily
(Domingos, 1996)	Sample	Yes	Arbitrarily
(Ramamurti and Ghosh, 1999)	Space	No	Induced
(Kohavi *et al*., 1997)	Space	Yes	Induced
(Bay, 1999)	Attribute	No	Arbitrarily
(Kusiak, 2000)	Attribute	Yes	Manually

5. The Relation to Other Methodologies

The main distinction between existing approaches, such as ensemble methods and distributed Data Mining to decomposition methodology, focuses on the following fact: the assumption that each model has access to a comparable quality of data is not valid in the decomposition approach (Tumer and Ghosh, 2000):

> A fundamental assumption in all the multi-classifier approaches is that the designer has access to the entire data set, which can be used in its entirety, resampled in a random (bagging) or weighted (boosting) way, or randomly partitioned and distributed. Thus, except for boosting situations, each classifier sees training data of comparable quality. If the individual classifiers are then appropriately chosen and trained properly, their performances will be (relatively) comparable in any region of the problem space. So gains from combining are derived from the diversity among classifiers rather that by compensating for weak members of the pool.

This assumption is clearly invalid for decomposition methodology, where classifiers may have significant variations in their overall performance. Furthermore when individual classifiers have substantially different performances over different parts of the input space, combining is still desirable (Tumer and Ghosh, 2000). Nevertheless neither simple combiners nor more sophisticated combiners are particularly well-suited for the type of problems that arise (Tumer and Ghosh, 2000):

> The simplicity of averaging the classifier outputs is appealing, but the prospect of one poor classifier corrupting the combiner makes this a risky choice. Weighted averaging of classifier outputs appears to provide some flexibility. Unfortunately, the weights are still assigned on a per classifier basis rather than a per tuple basis. If a classifier is accurate only in certain areas of the input space, this scheme fails to take advantage of the variable accuracy of the classifier in question. Using a combiner that provides different weights for different patterns can potentially solve this problem, but at a considerable cost.

The ensemble methodology is closely related to the decomposition methodology (see Chapter 45 in this volume). In both cases the final model is a composite of multiple models combined in some fashion. However, Sharkey (1996) distinguishes between these methodologies in the following way: the main idea of ensemble methodology is to combine a set of models, each of which solves the same original task. The purpose of ensemble methodology is to obtain a more accurate and reliable performance than when using a single model. On the other hand, the purpose of decomposition methodology is to break down a complex problem into several manageable problems, enabling each inducer to solve a different task. Therefore, in ensemble methodology, any model can provide a sufficient solution to the original task. On the other hand, in decomposition methodology, a combination of all models is mandatory for obtaining a reliable solution.

Distributed Data Mining (DDM) deals with mining data that might be inherently distributed among different, loosely coupled sites with slow connectivity, such as geographically distributed sites connected over the Internet (Kargupta and Chan, 2000). Usually DDM is categorized according to data distribution:

Homogeneous. In this case, the datasets in all the sites are built from the same common set of attributes. This state is equivalent to the sample decom-

position discussed above, when the decomposition structure is set by the environment.

Heterogeneous. In this case, the quality and quantity of data available to each site may vary substantially. Since each specific site may contain data for different attributes, leading to large discrepancies in their performance, integrating classification models derived from distinct and distributed databases is complex.

DDM can be useful also in the case of "mergers and acquisitions" of corporations. In such cases, since each company involved may have its own IT legacy systems, different sets of data are available.

In DDM the different sources are given, namely the instances are predecomposed. As a result, DDM is mainly focused on combining the various methods. Several researchers discuss ways of leveraging distributed techniques in knowledge discovery, such as data cleaning and preprocessing, transformation, and learning.

Prodromidis *et al.* (1999) proposed the JAM system a meta-learning approach for DDM. The meta-learning approach is about combining several models (describing several sets of data from several sources of data) into one high-level model. Guo and Sutiwaraphun (1998) describe a meta-learning concept know-as *knowledge probing*. In knowledge probing, supervised learning is organized into two stages. In the first stage, a set of base classifiers is constructed using the distributed data sets. In the second stage, the relationship between an attribute vector and the class predictions from all of the base classifiers is determined. Grossman *et al*. (1999) outline fundamental challenges for mining large-scale databases, one of them being the need to develop DDM algorithms.

A closely related field is *Parallel Data Mining* (PDM). PDM deals with mining data by using several tightly-coupled systems with fast interconnection, as in the case of a cluster of shared memory workstations (Zaki and Ho, 2000).

The main goal of PDM techniques is to scale-up the speed of the Data Mining on large datasets. It addresses the issue by using high performance, multiprocessor computers. The increasing availability of such computers calls for extensive development of data analysis algorithms that can scale up as we attempt to analyze data sets measured in terabytes on parallel machines with thousands of processors. This technology is particularly suitable for applications that typically deal with large amounts of data, e.g. company transaction data, scientific simulation and observation data. Another important example of PDM is the SPIDER project that uses shared-memory multiprocessors systems (SMPs) to accomplish PDM on distributed data sets (Zaki, 1999). Please refer to Chapter 48 for more information.

6. Summary

In this chapter we have reviewed the necessity of decomposition methodology in Data Mining and knowledge discovery. We have suggested an approach to categorize elementary decomposition methods. We also discussed the main characteristics of decomposition methods and showed its suitability to the current research in the literature.

References

Ali K. M., Pazzani M. J., Error Reduction through Learning Multiple Descriptions, Machine Learning, 24: 3, 173-202, 1996.

Anand R, Methrotra K, Mohan CK, Ranka S. Efficient classification for multiclass problems using modular neural networks. IEEE Trans Neural Networks, 6(1): 117-125, 1995.

Baxt, W. G., Use of an artificial neural network for data analysis in clinical decision making: The diagnosis of acute coronary occlusion. Neural Computation, 2(4):480-489, 1990.

Bay, S., Nearest neighbor classification from multiple feature subsets. Intelligent Data Analysis, 3(3): 191-209, 1999.

Bhargava H. K., Data Mining by Decomposition: Adaptive Search for Hypothesis Generation, INFORMS Journal on Computing Vol. 11, Iss. 3, pp. 239-47, 1999.

Biermann, A. W., Faireld, J., and Beres, T., 1982. Signature table systems and learning. IEEE Trans. Syst. Man Cybern., 12(5):635-648.

Blum A., and Mitchell T., Combining Labeled and Unlabeled Data with Co-Training. In Proc. of the 11th Annual Conference on Computational Learning Theory, pages 92-100, 1998.

Breiman L., Bagging predictors, Machine Learning, 24(2):123-140, 1996.

Buntine, W., "Graphical Models for Discovering Knowledge", in U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pp 59-82. AAAI/MIT Press, 1996.

Chan P.K. and Stolfo S.J, On the Accuracy of Meta-learning for Scalable Data Mining, J. Intelligent Information Systems, 8:5-28, 1997.

Chen K., Wang L. and Chi H., Methods of Combining Multiple Classifiers with Different Features and Their Applications to Text-Independent Speaker Identification, International Journal of Pattern Recognition and Artificial Intelligence, 11(3): 417-445, 1997.

Cherkauer, K.J., Human Expert-Level Performance on a Scientific Image Analysis Task by a System Using Combined Artificial Neural Networks. In Working Notes, Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms Workshop, Thirteenth National Conference on Artificial Intelligence. Portland, OR: AAAI Press, 1996.

Dietterich, T. G., and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 2:263-286, 1995.

Domingos, P., Using Partitioning to Speed Up Specific-to-General Rule Induction. In Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models, pp. 29-34, AAAI Press, 1996.

Domingos, P., & Pazzani, M., On the Optimality of the Naive Bayes Classifier under Zero-One Loss, Machine Learning, 29: 2, 103-130, 1997.

Fischer, B., "Decomposition of Time Series - Comparing Different Methods in Theory and Practice", Eurostat Working Paper, 1995.

Friedman, J. H., "Multivariate Adaptive Regression Splines", The Annual Of Statistics, 19, 1-141, 1991.

Friedman N., Geiger D., and Goldszmidt M., Bayesian Network Classifiers, Machine Learning 29: 2-3, 131-163, 1997.

Gama J., A Linear-Bayes Classifier. In C. Monard, editor, Advances on Artificial Intelligence – SBIA2000. LNAI 1952, pp 269-279, Springer Verlag, 2000

Grossman R., Kasif S., Moore R., Rocke D., and Ullman J., Data Mining research: Opportunities and challenges. Report of three NSF workshops on mining large, massive, and distributed data, 1999.

Guo Y. and Sutiwaraphun J., Knowledge probing in distributed Data Mining, in Proc. 4h Int. Conf. Knowledge Discovery Data Mining, pp 61-69, 1998.

Hansen J., Combining Predictors. Meta Machine Learning Methods and Bias, Variance & Ambiguity Decompositions. PhD dissertation. Aurhus University. 2000.

Hampshire, J. B., and Waibel, A. The meta-Pi network - building distributed knowledge representations for robust multisource pattern-recognition. Pattern Analyses and Machine Intelligence 14(7): 751-769, 1992.

He D. W., Strege B., Tolle H., and Kusiak A., Decomposition in Automatic Generation of Petri Nets for Manufacturing System Control and Scheduling, International Journal of Production Research, 38(6): 1437-1457, 2000.

Holmstrom, L., Koistinen, P., Laaksonen, J., and Oja, E., Neural and statistical classifiers - taxonomy and a case study. IEEE Trans. on Neural Networks, 8,:5–17, 1997.

Hrycej T., Modular Learning in Neural Networks. New York: Wiley, 1992.

Hu, X., Using Rough Sets Theory and Database Operations to Construct a Good Ensemble of Classifiers for Data Mining Applications. ICDM01. pp 233-240, 2001.

Jenkins R. and Yuhas, B. P. A simplified neural network solution through problem decomposition: The case of Truck backer-upper, IEEE Transactions on Neural Networks 4(4):718-722, 1993.

Johansen T. A. and Foss B. A., A narmax model representation for adaptive control based on local model -Modeling, Identification and Control, 13(1):25-39, 1992.

Jordan, M. I., and Jacobs, R. A., Hierarchical mixtures of experts and the EM algorithm. Neural Computation, 6, 181-214, 1994.

Kargupta, H. and Chan P., eds, Advances in Distributed and Parallel Knowledge Discovery , pp. 185-210, AAAI/MIT Press, 2000.

Kohavi R., Becker B., and Sommerfield D., Improving simple Bayes. In Proceedings of the European Conference on Machine Learning, 1997.

Kononenko, I., Comparison of inductive and Naive Bayes learning approaches to automatic knowledge acquisition. In B. Wielinga (Ed.), Current Trends in Knowledge Acquisition, Amsterdam, The Netherlands IOS Press, 1990.

Kononenko, I., SemiNaive Bayes classifier, Proceedings of the Sixth European Working Session on Learning, pp. 206-219, Porto, Portugal: SpringerVerlag, 1991.

Kusiak, A., Decomposition in Data Mining: An Industrial Case Study, IEEE Transactions on Electronics Packaging Manufacturing, Vol. 23, No. 4, pp. 345-353, 2000.

Kusiak, E. Szczerbicki, and K. Park, A Novel Approach to Decomposition of Design Specifications and Search for Solutions, International Journal of Production Research, 29(7): 1391-1406, 1991.

Langley, P. and Sage, S., Oblivious decision trees and abstract cases. in Working Notes of the AAAI-94 Workshop on Case-Based Reasoning, pp. 113-117, Seattle, WA: AAAI Press, 1994.

Liao Y., and Moody J., Constructing Heterogeneous Committees via Input Feature Grouping, in Advances in Neural Information Processing Systems, Vol.12, S.A. Solla, T.K. Leen and K.-R. Muller (eds.),MIT Press, 2000.

Long C., Bi-Decomposition of Function Sets Using Multi-Valued Logic, Eng. Doc. Dissertation, Technischen Universitat Bergakademie Freiberg 2003.

Lu B.L., Ito M., Task Decomposition and Module Combination Based on Class Relations: A Modular Neural Network for Pattern Classification, IEEE Trans. on Neural Networks, 10(5):1244-1256, 1999.

Maimon O. and Rokach L., "Improving supervised learning by feature decomposition", Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems, Lecture Notes in Computer Science, Springer, pp. 178-196, 2002.

Maimon O. and Rokach L., "Decomposition Methodology for Knowledge Discovery and Data Mining: Theory and Applications", World Scientific, 2005.

Meretakis, D. and Wthrich, B., Extending Nave Bayes Classifiers Using Long Itemsets, in Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, pp. 165-174, San Diego, USA, 1999.

Michie, D., Problem decomposition and the learning of skills, in Proceedings of the European Conference on Machine Learning, pp. 17-31, Springer-Verlag, 1995.

Nowlan S. J., and Hinton G. E. Evaluation of adaptive mixtures of competing experts. In Advances in Neural Information Processing Systems, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, Eds., vol. 3, pp. 774-780, Morgan Kaufmann Publishers Inc., 1991.

Ohno-Machado, L., and Musen, M. A. Modular neural networks for medical prognosis: Quantifying the benefits of combining neural networks for survival prediction. Connection Science 9, 1, 1997, 71-86.

Peng, F. and Jacobs R. A., and Tanner M. A., Bayesian Inference in Mixtures-of-Experts and Hierarchical Mixtures-of-Experts Models With an Application to Speech Recognition, Journal of the American Statistical Association, 1995.

Pratt, L. Y., Mostow, J., and Kamm C. A., Direct Transfer of Learned Information Among Neural Networks, in: Proceedings of the Ninth National Conference on Artificial Intelligence, Anaheim, CA, 584-589, 1991.

Provost, F.J. and Kolluri, V., A Survey of Methods for Scaling Up Inductive Learning Algorithms, Proc. 3rd International Conference on Knowledge Discovery and Data Mining, 1997.

Quinlan, J. R., C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, 1993.

Rahman, A. F. R., and Fairhurst, M. C. A new hybrid approach in combining multiple experts to recognize handwritten numerals. Pattern Recognition Letters, 18: 781-790,1997.

Ramamurti, V., and Ghosh, J., Structurally Adaptive Modular Networks for Non-Stationary Environments, IEEE Transactions on Neural Networks, 10 (1):152-160, 1999.

Ridgeway, G., Madigan, D., Richardson, T. and O'Kane, J., Interpretable Boosted Naive Bayes Classification, Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp 101-104, 1998.

Ronco, E., Gollee, H., and Gawthrop, P. J., Modular neural network and self-decomposition. CSC Research Report CSC-96012, Centre for Systems and Control, University of Glasgow, 1996.

Saaty, X., The analytic hierarchy process: A 1993 overview. Central European Journal for Operations Research and Economics, Vol. 2, No. 2, p. 119-137, 1993.

Samuel, A., Some studies in machine learning using the game of checkers II: Recent progress. IBM J. Res. Develop., 11:601-617, 1967.

Sharkey, A., On combining artificial neural nets, Connection Science, Vol. 8, pp.299-313, 1996.

Sharkey, A., Multi-Net Iystems, In Sharkey A. (Ed.) Combining Artificial Neural Networks: Ensemble and Modular Multi-Net Systems. pp. 1-30, Springer -Verlag, 1999.

Tumer, K. and Ghosh J., Error Correlation and Error Reduction in Ensemble Classifiers, Connection Science, Special issue on combining artificial neural networks: ensemble approaches, 8 (3-4): 385-404, 1996.

Tumer, K., and Ghosh J., Linear and Order Statistics Combiners for Pattern Classification, in Combining Articial Neural Nets, A. Sharkey (Ed.), pp. 127-162, Springer-Verlag, 1999.

Weigend, A. S., Mangeas, M., and Srivastava, A. N. Nonlinear gated experts for time-series - discovering regimes and avoiding overfitting. International Journal of Neural Systems 6(5):373-399, 1995.

Zaki, M. J., Ho C. T., and Agrawal, R., Scalable parallel classification for Data Mining on shared- memory multiprocessors, in Proc. IEEE Int. Conf. Data Eng., Sydney, Australia, WKDD99, pp. 198– 205, 1999.

Zaki, M. J., Ho C. T., Eds., Large- Scale Parallel Data Mining. New York: Springer- Verlag, 2000.

Zupan, B., Bohanec, M., Demsar J., and Bratko, I., Feature transformation by function decomposition, IEEE intelligent systems & their applications, 13: 38-43, 1998.