Feature Selection, Extraction and Construction

Hiroshi Motoda

Inst. of Sci. & Indus. Res.
Osaka University
Ibaraki, Osaka, Japan 567-0047

Huan Liu

Dept. of CS & Eng.
Arizona State University
Tempe, AZ, USA 85287-5406

Abstract

Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion. Feature extraction/construction is a process through which a set of new features is created. They are used either in isolation or in combination. All attempt to improve performance such as estimated accuracy, visualization and comprehensibility of learned knowledge. Basic approaches to these three are reviewed giving pointers to references for further studies.

1 Introduction

Researchers and practitioners realize that an important part of data mining is pre-processing in which data is processed before it is presented to any learning, discovering, or visualizing algorithm [23, 11]. Feature extraction, selection and construction is one effective approach to data reduction among others such as instance selection [24], data selection [25]. The goal of feature extraction, selection and construction is three fold: 1) reducing the amount of data; 2) focusing on the relevant data; and 3) improving the quality of data and hence the performance of data mining algorithms, such as learning time, predictive accuracy. There could be two main approaches. One is to rely on data mining algorithms and the other is to conduct preprocessing before data mining. It seems natural to let data mining algorithms deal with data directly as the ultimate goal of data mining is to find hidden patterns from data. Indeed, many data mining methods attempt to select, extract, or construct features, however, both theoretical analyses and experimental studies indicate that many algorithms scale poorly in domains with large numbers of irrelevant and/or redundant features [17].

The other approach is to preprocess the data so that it is made suitable for data mining. Feature selection, extraction and construction are normally tasks of preprocessing, and are independent of data mining. First, it can be done once and used for all subsequent data mining tasks. Second, it usually employs a less expensive evaluation measure than using a data mining algorithm. Hence, it can handle larger sized data than data mining can. Third, it often works off-line. Therefore, if necessary, many different algorithms can be tried. However, in addition to their being mainly preprocessing tasks, there are other commonalities among them: 1) they try to achieve the same goal for data reduction, 2) they require some criteria to make sure that the resulted data allows data mining algorithm to accomplish nothing less, if not more, and 3) their effectiveness has to be measured in multiple aspects such as reduced amounts of data, relevance of the reduced data, mostly, if possible, their direct impact on data mining algorithms. Feature selection (FS), extraction (FE) and construction (FC) can be used in combination. In many cases, feature construction expands the number of features with newly constructed ones that are more expressive but they may include useless features. Feature selection can help automatically reduce those excessive features.

2 Feature Selection

2.1 Concept

Feature selection is a process that chooses a subset of M features from the original set of N features ($M \leq N$), so that the feature space is optimally reduced according to a certain criterion [3, 5]. According to [21], the role of feature selection in machine learning is 1) to reduce the dimensionality of feature space, 2) to speed up a learning algorithm, 3) to improve the predictive accuracy of a classification algorithm, and 4) to improve the comprehensibility of the learning results. Recent study about feature selection in unsupervised learning context shows that feature selection can also help to improve the performance of clustering algorithms with reduced feature space [31, 32, 7, 6, 14]. In general, feature selection is a search problem according to some evaluation criterion.

Feature subset generation One intuitive way is to generate subsets of features sequentially. If we start with an empty subset and gradually add one feature at a time, we adopt a scheme called *sequential forward selection*; if we start with a full set and remove one feature at a time, we have a scheme called *sequential backward selection*. We can also *randomly* generate a subset so that each possible subset (in total, 2^N, where N is the number of features) has an approximately equal chance to be generated. One extreme way is to *exhaustively* enumerate 2^N possible subsets.

Feature evaluation An optimal subset is always relative to a certain evaluation criterion (*i.e.* an optimal subset chosen using one evaluation criterion may not be the same as that using another evaluation criterion). Evaluation criteria can be broadly categorized into two groups based on their dependence on the learning algorithm applied on the selected feature subset. Typically, an independent criterion (i.e. filter) tries to evaluate the goodness of a feature or feature subset without the involvement of a learning algorithm in this process. Some of the independent criteria are distance measure, information measure, dependency measure, consistency measure [LM98b]. A dependent criterion (i.e. wrapper) tries to evaluate the goodness of a feature or feature subset by evaluating the performance of the learning algorithm applied on the selected subset. In other words, it is the same measure on the performance of the applied learning algorithm. For supervised learning, the primary goal of classification is to maximize predictive accuracy, therefore, predictive accuracy is generally accepted and widely used as the primary measure by researchers and practitioners. While for unsupervised learning, there exist a number of heuristic criteria for estimating the quality of clustering results, such as cluster compactness, scatter separability, and maximum likelihood. Recent reviews on developing dependent evaluation criteria for unsupervised feature selection based on these criteria can be found in [Tal99b, DB00, KSM00].

2.2 Algorithms

Many feature selection algorithms exist. Using the general model described earlier, we can regenerate these existing algorithms by having proper combinations for each component.

Exhaustive/complete approaches Focus [1, 2] applies an inconsistency measure and exhaustively evaluates subsets starting from subsets with one feature (i.e., sequential forward search); Branch-and-Bound [27, 30] evaluates estimated accuracy, and ABB [22] checks an inconsistency measure that is monotonic. Both start with a full feature set until the preset bound cannot be maintained.

Heuristic approaches SFS (sequential forward search) and SBS (sequential backward search) [30, 5, 3] can apply any of five measures. DTM [4] is the simplest version of a wrapper model - just learn a classifier once and use whatever features found in the classifier.

Nondeterministic approaches LVF [18] and LVW [19] randomly generate feature subsets but test them differently: LVF applies an inconsistency measure, LVW uses accuracy estimated by a classifier. Genetic Algorithms and Simulated Annealing are also used in feature selection [30, 13]. The former may produce multiple subsets, the latter produces a single subset.

Instance-based approaches Relief [15, 16] is a typical example for this category. There is no explicit procedure for feature subset generation. Many small data samples are sampled from the data. Features are weighted according to their roles in differentiating instances of different classes for a data sample. Features with higher weights can be selected.

3 Feature Extraction

3.1 Concepts

Feature extraction is a process that extracts a set of new features from the original features through some functional mapping [35]. Assuming there are n features (or attributes) $A_1, A_2, ..., A_n$, after feature extraction, we have another set of new features $B_1, B_2, ..., B_m (m < n), B_i = F_i(A_1, A_2, ..., A_n)$, and F_i is a mapping function. Intensive search is generally required in finding good transformations. The goal of feature extraction is to search for a minimum set of new features via some transformation according to some performance measure. The major research issues can therefore be summarized as follows.

Performance Measure It investigates what is the most suitable in evaluating extracted features. For a task of classification, the data has class labels and predictive accuracy might be used to determine what is a set of extracted features. When it is of clustering, the data does not have class labels and one has to resort to other measures such as inter-cluster/intra-cluster similarity, variance among data, etc.

Transformation It studies ways of mapping original attributes to new features. Different mappings can be employed to extract features. In general, the mappings can be categorized into linear or nonlinear transformations. One could categorize transformations along two dimensions: linear and labeled, linear and non-labeled, nonlinear and labeled, nonlinear and non-labeled. Many data mining techniques can be used in transformation such as EM, k-Means, k-Medoids, Multi-layer Perceptrons, etc [12].

Number of new features It surveys methods that determine the minimum number of new features. With our objective to create a minimum set of new features, the real question is how many new features can ensure that "the true nature" of the data remains after transformation.

One can take advantage of data characteristics as a critical constraint in selecting performance measure, number of new features, and transformation. In addition to with/without class labels, data attributes can be of various types: continuous, nominal, binary, mixed. Feature extraction can find its many usages: dimensionality reduction for further processing [23], visualization [8], compound features used to booster some data mining algorithms [20].

3.2 Algorithms

The functional mapping can be realized in several ways. We present here two exemplar algorithms to illustrate how they treat different aspects of feature extraction.

A feedforward neural networks approach A single hidden layer multilayer perceptron can be used to to extract new features [29]. The basic idea is to use the hidden units as newly extracted features. The predictive accuracy is estimated and used as the performance measure. This entails that data should be labeled with classes. The transformation from input units to hidden units is non-linear. Two algorithms are designed to construct a network with the minimum number of hidden units and the minimum of connections between the input and hidden layers: the network construction algorithm parsimoniously adds one more hidden unit to improve predictive accuracy; and the network pruning algorithm generously removes redundant connections between the input and hidden layers if predictive accuracy does not deteriorate.

Principal Component Analysis PCA is a classic technique in which the original n attributes are replaced by another set of m new features that are formed from linear combinations of the original attributes. The basic idea is straightforward: to form an m-dimensional projection ($1 \leq m \leq n-1$) by those linear combinations that maximize the sample variance subject to being uncorrelated with all these already selected linear combinations. Performance measure is sample variance; the number of new features, m, is determined by the m principal components that capture the amount of variance subject to a pre-determined threshold; and the transformation is linear combination. PCA does not require that data be labeled with classes. The search for m principal components can be rephrased to finding m eigenvectors associated with the m largest eigenvalues of the covariance matrix of a data set [12].

4 Feature Construction

4.1 Concepts

Feature construction is a process that discovers missing information about the relationships between features and augments the space of features by inferring or creating additional features [26, 34, 23]. Assuming there are n features $A_1, A_2, ..., A_n$, after feature construction, we may have additional m features $A_{n+1}, A_{n+2}, ..., A_{n+m}$. All new constructed features are defined in terms of original features, as such, no inherently new informed is added through feature construction. Feature construction attempts to increase the expressive power of the original features. Usually, the dimensionality of the new feature set is expanded and is bigger than that of the original feature set. Intuitively, there could be exponentially many combinations of original features, and not all combinations are necessary and useful. Feature construction aims to automatically transform the original representation space to a new one that can help better achieve data mining objectives: improved accuracy, easy comprehensibility, truthful clusters, revealing hidden patterns, etc. Therefore, the major research issues of feature construction are the following four.

How to construct new features Various approaches can be categorized into four groups: data-driven, hypothesis-driven, knowledge-based, and hybrid [34, 23]. The data-driven approach is to construct new features based on analysis of the available data by applying various operators. The hypothesis-driven approach is to construct new features based on the hypotheses generated previously. The knowledge-based is to construct new features applying existing knowledge and domain knowledge.

How to choose and design operators for feature construction There are many operators for combining features to form compound features [23]. Conjunction, disjunction and negation are commonly used constructive operators for nominal features. Other common operators are M-of-N and X-of-N [36], where M-of-N is true iff at least M out of N conditions are true, and X-of-N X iff X of N conditions are true; cartesian product [28] of two or more nominal features. For numerical features, simple algebraic operators such as equivalence, inequality, addition, subtraction, multiplication, division, maximum, minimum, average are often used to construct compound features.

How to use operators to construct new features efficiently It is impossible to exhaustively explore every possible operator. It is, thus, imperative to find intelligent methods that can avoid exhaustive search and heuristically try potentially useful operators. This line of research investigates the connections between data mining tasks, data characteristics, and operators that could be effective.

How to measure and select useful new features Not all constructed features are good ones. We have to be selective. One option is to handle the selection part by applying feature selection techniques to remove redundant and irrelevant features. When the number of features is very large, it is sensible to make decision while a new compound feature is generated to avoid too many features. This would require an effective measure to evaluate a new feature and provide an indicator. Researchers are investigating various measures that are not computationally expensive. Some examples are measures of consistency, distance as used in feature selection [21].

4.2 Algorithms

Feature construction can be realized in several ways. We show here two exemplar algorithms to illustrate how new features are constructed and built into an induction model. Many examples can be found in [23].

Greedy search for use in decision tree nodes A straightforward algorithm is to use a greedy search. In case of a decision tree induction, the algorithm generates at each decision node one new feature based on both original features and those already constructed and select the best one. To construct a new feature, the algorithm performs a greedy search in the instance space using a prespecified set of constructive operators. The search starts from an empty set. At each search step, it either adds one possible feature-value pair or deletes one possible feature-value pair in a systematic manner. An evaluation function that takes both class entropy and model complexity into account can be used. Optimal M-of-N and X-of-N can be found in this manner [36]. A variant of this which is useful for numeric attributes is to search for the best linear discriminant function [9] and its extension to functional trees [10].

Genetic algorithm for use in wrapper mode Genetic algorithms are adaptive search techniques for evolutional optimization. Each individual is evaluated based on its overall fitness to the given application domain. New individuals are constructed from their parents by two operators: mutation and crossover, as well as copy. An individual is represented by a variable-length nested list structure comprising a set of original and compound features. For continuous features, we can use a set of arithmetic operators such as +, , *, /. One good application is image classification (eye detection in pictures) [33] in which both feature construction and feature selection are interleaved and C4.5 is used to return a fitness value.

5 Conclusions

Feature extraction and construction are the variants of feature transformation through which a new set of features is created. Feature construction often expands the feature space, whereas feature extraction usually reduces the feature space. Feature transformation and feature selection are not two totally independent issues. They can be viewed as two sides of the representation problem. We can consider features as a representation language. In some cases where this language contains more features than necessary, feature selection helps simplify the language; in other cases where this language is not sufficient to describe the problem, feature construction help enrich the language by constructing compound features. The use of feature selection, extraction and construction depends on the purpose for simpler concept description or for better data mining task performance.

Despite of recent advancement of feature selection, extraction and construction, much work is needed to unify this currently still diversified field. Many types of data exist in practice. Boolean, nominal, and numerical types are popular, but others like structural, relational, temporal should also receive our equal attention in data mining applications of real world problems. Feature selection, extraction and construction are key techniques in answering the pressing needs for data mining. These techniques can help reduce data for mining or learning tasks and enable those mining algorithms, which were unable to mine, to mine.

Acknowledgements

This work was partially supported by the grant-in-aid for scientific research on priority area "Active Mining" funded by the Japanese Ministry of Education, Culture, Sports, Science and Technology for H. Motoda and by the National Science Foundation under Grant No. IIS-0127815 for H. Liu.

References

[1] H. Almuallim and T. Dietterich, "Learning with Many Irrelevant Features", *Proc. of the Ninth National Conference on Artificial Intelligence*, pp. 547–552, 1991

[2] H. Almuallim and T. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features", *Artificial Intelligence* 69, 1-2, pp. 279–305, 1994.

[3] A.L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning", *Artificial Intelligence*, 97, pp. 245–271, 1997.

[4] C. Cardie, "Using Decision Trees to Improve Case-Based Learning", *Proc of the Tenth International Conference on Machine Learning*, pp. 25–32, 1993.

[5] M. Dash and H. Liu, "Feature Selection Methods for Classifications", *Intelligent Data Analysis: An International Journal*, 1, 3, 1997. http://www-east.elsevier.com/ida/free.htm.

[6] M. Dash and H. Liu,"Feature Selection for Clustering", *Proc. of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, (PAKDD-2000)*, Springer Verlag, pp. 110–121, 2000.

[7] J. G. Dy and C. E. Brodley, "Feature Subset Selection and Order Identification for Unsupervised Learning. *Proc. of the Seventeenth International Conference on Machine Learning*, pp. 247–254, 2000.

[8] U. Fayyad, G.G. Grinstein, and A. Wierse, *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann Publishers, 2001.

[9] J. Gama and P. Brazdil, "Constructive Induction on Continuous Spaces", chapter 18, pp. 289–303. In [23], 1998. 2nd Printing, 2001.

[10] J. Gama, "Functional Trees", em Proc. of the Fourth International Conference on Discovery Science, pp. 58–73, 2001.

[11] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufman, 2001.

[12] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, A Bradford Book The MIT press, 2001.

[13] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19, 2, pp. 153–158, 1997.

[14] Y. Kim, W. Street, and F. Menczer, "Feature Selection for Unsupervised Learning via Evolutionary Search", *Proc. of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 365–369, 2000.

[15] K. Kira and L. Rendell, "The feature selection problem: Traditional methods and a new algorithm", *Proc. of the Tenth National Conference on Artificial Intelligence*, pp. 129–134, 1992.

[16] I. Kononenko, "Estimating attributes : Analysis and extension of RELIEF", *Proceedings of the European Conference on Machine Learning*, pp. 171–182, 1994.

[17] P. Langley, *Elements of Machine Learning*, Morgan Kaufmann, 1996.

[18] H. Liu, and R. Setiono, R, "A Probabilistic Approach to Feature Selection - A Filter Solution", *Proc. of the International Conference on Machine Learning (ICML-96)*, pp. 319–327, 1996.

[19] H. Liu and R. Setiono, "Feature Selection and Classification - A Probabilistic Wrapper Approach", *Proc. of the Ninth International Conference on Industrial and Engineering Applications of AI and ES*, pp. 419–424, 1996.

[20] H. Liu and R. Setiono, "Feature Transformation and Multivariate Decision Tree Induction", *Proc. of the First International Conference on Discovery Science (DS'98)*, Springer Verlag, pp. 279–290, 1998.

[21] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery Data Mining*, Boston: Kluwer Academic Publishers, 1998.

[22] H. Liu, H. Motoda and M. Dash, "A Monotonic Measure for Optimal Feature Selection", *Proc. of the European Conference on Machine Learning*, pp. 101–106, 1998

[23] H. Liu and H. Motoda, editors, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Boston: Kluwer Academic Publishers, 1998. 2nd Printing, 2001.

[24] H. Liu and H. Motoda, editors, *Instance Selection and Construction for Data Mining*, Boston: Kluwer Academic Publishers, 2001.

[25] H. Liu, H. Lu, and J. Yao, "Toward Multi-database Mining: Identifying Relevant Databases", *IEEE Transactions on Knowledge and Data Engineering*, 13, 4, pp. 541–553, 2001.

[26] C.J. Matheus, "The Need for Constructive Induction", *Proc. of the Eighth International Workshop on Machine Learning*, pp. 173–177, 1991.

[27] P. Narendra and K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection", *IEEE Trans. on Computer*, C-26, 9, pp. 917–922, 1977.

[28] M.J. Pazzani, "Constructive Induction of Cartesian Product Attributes", chapter 21, pp. 341–354. In [23], 1998. 2nd Printing, 2001.

[29] R. Setiono and H. Liu, "Feature Extraction via Neural Networks", chapter 12, pp. 191–204. In [23], 1998. 2nd Printing, 2001.

[30] W. Siedlecki and J. Sklansky, "On Automatic Feature Selection", *International Journal of Pattern Recognition and Artificial Intelligence*, 2, pp. 197–220, 1988

[31] L. Talavera,"Feature Selection as A Preprocessing Step for Hierarchical Clustering", *Proc. of the Sixteenth International Conference on Machine Learning*, pp. 389–397, 1999.

[32] L. Talavera, "Feature Selection as Retrospective Pruning in Hierarchical Clustering", *Proc. of the Third Symposium on Intelligent Data Analysis (IDA'99)*, pp. 75–86, 1999.

[33] H. Vafaie and K. De Jong, "Evolutionary Feature Space Transformation", pp. 307–323. In [23], 1998. 2nd Printing, 2001.

[34] J. Wnek and R.S. Michalski, "Hypothesis-Driven Constructive Induction in AQ17-HCI: A Method and Experiments", *Machine Learning*, 14, pp. 139–168, 1994.

[35] N. Wyse, R. Dubes, and A.K. Jain, "A critical evaluation of intrinsic dimensionality algorithms", In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, pp. 415–425. Morgan Kaufmann Publishers, Inc., 1980.

[36] Z. Zheng, "A Comparison of Constructing Different Types of New Features for Decision Tree Learning", chapter 15, pp. 239 – 255. In [23], 1998. 2nd Printing, 2001.