Learning, attribute-value

Hendrik Blockeel

Department of Computer Science, Katholieke Universiteit Leuven, Belgium &
Leiden Institute of Advanced Computer Science, Universiteit Leiden, The Netherlands
hendrik.blockeel@cs.kuleuven.be

Synonyms

Standard setting (in the context of machine learning, data mining, statistics)

Definition

Attribute-value learning is a setting for [machine learning](#) where the data to be analyzed are given in a tabular format: each row represents an individual, each column represents a property of the individual (also referred to as attribute or variable), and the entries in the table are the values of these attributes for each individual. In attribute-value learning, each individual is described using exactly the same attributes (only the values for these attributes differ from one individual to another), and these descriptions are independent of each other (the description of one individual does not refer to other individuals). This sets it apart from [relational learning](#).

Characteristics

In statistics, machine learning and data mining, the attribute-value learning setting (sometimes called the *standard setting*, e.g., Breiman et al., 1984) is by far the most common one. Most methods and algorithms make the assumption that the data consist of a set S of independent individuals, each of which is described using the same variables. Consider for instance [linear regression](#): a set of observations of the variables $x_1, x_2, \dots, x_k$, and y are given, and the task is to construct a model of the form $y = a_1 x_1 + \dots + a_k x_k$, which can be used to predict y from $x_1, x_2, \dots, x_k$. There is an implicit assumption here that the variables $x_1, x_2, \dots, x_k$ and y are well-defined for each individual, and that no other information is to be taken into account. The same assumption is made by other statistical or machine learning techniques, such as [decision tree](#) induction, [rule based methods](#), [nearest neighbor methods](#), [support vector machines](#), [artificial neural networks](#), [Bayesian networks](#), [principal components analysis](#), [multi-dimensional scaling](#), etc.

In the attribute-value setting, the data to be analyzed can always be represented as a single table, where columns are attributes (or variables), rows are instances, and the values in each row describe the instance by listing its values for the attributes. An example of such a dataset is the Iris dataset (Fisher, 1936), available at the UCI data set repository (Frank & Asuncion, 2010):

Sepal length	Sepal Width	Petal length	Petal width	Class
--------------	-------------	--------------	-------------	-------

5.1	3.5	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
7.0	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.3	3.3	6.0	2.5	Virginica
7.1	3.0	5.9	2.1	Virginica

An example of a classification rule that is consistent with the above examples is:

if Sepal length < 6 then Setosa else if Petal length < 5 then Versicolor else Virginica

The classification rule's decision is based entirely on the values of a few well-defined attributes.

While this setting is very natural, it is not for all learning problems the most natural setting. Consider QSAR (quantitative structure-activity relationship) modeling: one needs to predict the value of a variable y from a description of the structure of a molecule. A molecule's structure could be described in multiple ways, for instance, as a graph. There does not exist a natural and finite set of attributes, though, such that each molecule is uniquely determined by listing the values for these attributes. Molecules have properties such as molecular weight, the number of carbon atoms in the molecule, etc., which are straightforward attributes that can be used to describe the molecule, but the molecular structure is not uniquely determined by these attributes. As a result, an attribute-value representation of a molecule is necessarily an incomplete summary of all the information that is in the molecular structure. The summary can be made more informative by including more attributes, but it cannot be made complete, as the following discussion shows.

An illustration of the limitations of attribute-value learning

To illustrate the problem of representing complex objects in the attribute-value framework, consider the following problem. Assume we need to classify drawings, where a drawing can contain a number of simple objects. For instance, one drawing could have a blue disc and a red square, and be classified as "positive"; another drawing has two red discs and is classified as "negative". If we know that drawings never contain more than two objects, and it is sufficient to characterize the objects by their shape and color, then the following may seem an appropriate attribute-value representation of the above two examples:

Shape_obj1	Color_obj1	Shape_obj2	Color_obj2	Class
Disc	Blue	Square	Red	Positive
Disc	Red	Disc	Red	Negative

A rule that is consistent with these data is

if Shape_obj2 = Square then positive else negative

The problem with the above representation and with the rule is that the attributes are not well-defined. Unless a natural order is defined on the objects, we cannot know which is object 1 and which is object 2. An equally valid representation of the above data would be one where the red square is called object 1 and the blue disc object 2. However, for that representation, the above rule would make an incorrect prediction. Since there is no natural way to decide which is object 1 and object 2, the rule is really meaningless. It has meaning at the level of one particular representation of the drawings, but not at the level of the drawings themselves.

Instead of the above attributes, one could define alternative attributes such as “contains_a_blue_disc”, “contains_a_square”, “contains_a_red_square”, etc. These attributes do have a well-defined meaning for each drawing. But many such attributes can be defined, and no matter how many we include, there may always be relevant properties that we miss. Assume that we include all attributes of the form “contains_a_<color>_<shape>”, with each combination of color and shape present, then the condition “contains at least two discs” still cannot be expressed.

This example illustrates an inherent problem with attribute-value representations, which is that it may be difficult to define a finite set of well-defined attributes such that each possibly relevant condition can be expressed using these attributes.

Besides the attribute-value learning setting, there is the so-called [relational learning](#) setting. In this setting, a more flexible data representation language is used, and learners learn directly from information described in that language. Relational learning is more powerful in the sense that it can handle a larger variety of input data, but is more limited with respect to the learning algorithms that are available, and the extent to which these algorithms are optimized for scalability and performance.

Due to the limited availability of efficient relational learners, relational data is sometimes transformed (typically not without loss of information) into the attribute-value format; this process is called *propositionalization*. This approach has the advantage that powerful and efficient machine learning tools can be used, but only on a summary of the data.

Attribute-value learning methods

A wide range of learning algorithms and methods is available for attribute value learning. These include [decision tree](#) learners, [rule based methods](#), [nearest neighbor methods](#), [support vector machines](#), [artificial neural networks](#), [Bayesian networks](#), [regression analysis](#), [principal components analysis](#), [multi-dimensional scaling](#), ... These methods are described in more detail elsewhere in this volume. Most of these descriptions implicitly assume that the data are in the attribute-value format. Similarly, most practical tools for data mining or machine learning (such as [Weka](#)) implement only the attribute-value setting (sometimes limited extensions of it). Also many machine learning *tasks* ([classification](#), [regression](#), [clustering](#), [pattern mining](#), ...) are typically defined in the context of attribute-value learning, though their definitions can usually be extended towards the relational learning setting.

Cross-references

Classification
Regression analysis
Decision Tree
Rule based methods
Nearest neighbor methods
Support Vector Machines
Artificial Neural Networks
Bayesian networks
Artificial Intelligence and Machine Learning
Multidimensional scaling
Principal components analysis
Clustering
Pattern mining
Weka
Learning, Relational

References

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Chapman and Hall/CRC, Boca Raton, Florida, USA.

Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.

Fisher, R.A. (1936). The use of multiple measurements in taxonomic problems. Annual Eugenics, 7, Part II, 179-188.