

Attribute Classification Using Feature Analysis

Felix Naumann, Ching-Tien Ho, Xuqing Tian, Laura Haas, and Nimrod Megiddo
IBM Almaden Research Center, San Jose, CA 95120
{felix,ho,laura,megiddo}@almaden.ibm.com, tianxq@acm.org

The basis of many systems that integrate data from multiple sources is a set of correspondences between source schemata and a target schema. Correspondences express a relationship between sets of source attributes, possibly from multiple sources, and a set of target attributes. Clio is an integration tool that assists users in defining value correspondences between attributes [1].

In real life scenarios there may be many sources and the source relations may have many attributes. Users can get lost and might miss or be unable to find some correspondences. Also, in many real life schemata the attribute names reveal little or nothing about the semantics of the data values. Only the data values in the attribute columns can convey the semantic meaning of the attribute. Our work relieves users of the problems of too many attributes and meaningless attribute names, by automatically *suggesting* correspondences between source and target attributes. For each attribute, we analyze the data values and derive a set of features. The overall feature set forms the characteristic signature of an attribute. There are more likely to be correspondences between attributes with similar signatures than between others. Our results show that a properly chosen small set of domain-independent features can mostly capture structural information of attributes.

Non-numerical attributes. Features for non-numerical attributes include the presence of characters, such as the @-symbol or a space character, in the data field. Also, we examine aggregate features, such as the presence of any upper case character (see table below). An attribute signature vector stores the average number of occurrences (as a fraction) of the Boolean features for all of its values.

Singletons	Aggregates
{a}, ..., {z},	{@,-, ..., /, \}, {0,...,9}
{A}, ..., {Z},	{a,e,i,o,u}, {A,E,I,O,U}
{0}, ..., {9},	{a,...,z}, {A,...,Z}
{@}, ..., {\}	{a,..., z,A,...,Z}
	{a,..., z,A,...,Z,0,...,9}

We describe the problem of finding corresponding attributes as a classification problem, and use the Naïve Bayes classifier to decide upon most likely matches. Several different domains served to test the features and Naïve Bayes classifier: a set of three bibliography databases, data from three real estate Web sites, a commer-

cial semiconductor manufacturers database collecting, and an insurance database. The experiments showed an expected strong dependency between column size and accuracy. For all domains we observed satisfying misclassification rates below 5% for > 250 training data and > 16 test data values.

Numerical attributes. For numerical attributes we use a non-Boolean feature set: To best model the range and distribution of the values of an attribute, we choose 18 quantile features for our implementation—the 10%, 20%, to 90% quantiles of each data column, and the 10%, 20%, to 90% quantiles of each data column after removing all data of value zero.

To decide upon most likely correspondences, we introduce a new quantile-based classification method. As before, we generate one signature for each numerical attribute from the training column. We then generate the signature for the numerical test attribute to be classified. We evaluate the quantile-based classification methods by applying them to two real world databases and a synthetic data set. The first database is a collection of numerical attributes taken from a large biochemical database; the second is an insurance database. Finally, we generated several data columns with synthetic data and only slightly differing distributions. We achieved on average misclassification rates below 5% for > 64 training and test data values.

Summary. The techniques described above are successfully implemented as used in Clio. Currently, Clio has four deployment modes of the attribute matching techniques: correspondence discovery between source and target attributes; foreign-key dependency discovery between source attributes; rapid table matching, to quickly map all attributes of a table; and relation hiding, to reduce the user’s view of very large schemata to only the relevant tables.

References

- [1] M.A. Hernández, R.J. Miller, L.M. Haas, L. Yan, C.T. Ho, and X. Tian. Clio: A semi-automatic tool for schema mapping. In *Proceedings of the ACM SIGMOD Conference*, 2001.