Object Prediction Using Naive Bayes Classifier

Komal
M.Tech. (Computer Science & Engineering)
Department of Computer Science and Applications, Kurukshetra University, Kurukshetra
thakrankomal07@gmail.com

Abstract: Data mining is a very powerful technology with great potential to help companies focus on the most important information or facts in the data they have collected about the behavior of their customers. It discovers information within the data that queries can't effectively reveal. Data mining with granular computing make a flexible and scalable analysis of targeted data object. Granular Computing provides a conceptual framework for studying many issues in data mining. This paper examines those issues, including mining data object, related knowledge representation and object prediction using Naïve Bayes classifier.

Keywords: Naïve bayes classifier, objects dataset, object prediction, Pattern prediction.

1. INTRODUCTION

Data Mining consists of extracting knowledge automatically from large amounts of data. This knowledge can be relationships between variables (association rules), or groups of items that would be similar (clusters). It can also be classifiers, e.g. functions mapping data items to classes given their features. This paper deals with the latter. A feature is a stored attribute of an item. This could the size of a person or the email content. The class is the value of a specified attribute that is to be predicted. The possible numbers of classes are finite and predetermined.

This paper is based on the supervised learning approach. Two successive steps involving each a data set are introduced: the training phase and the classification phase of data set. The first phase includes building a classifier with items whose classes are known. These elements are the training set. Then, the function is used on a second set of items to predict class of items. Consider for instance spam detection. A classifier is first trained with a small set of emails manually labeled legitimate" or "spam". "Then, it can predict the class of emails it has never met before.

The problem studied in this paper is the following: how should a classification algorithm be implemented in a Database Management System (DBMS)? Indeed, a traditional way to manage classification is to store the datasets in a DBMS and run the algorithms with an external application. The studied data sets are entirely copied to the application memory space. This allows immediate treatments, as the calculations are based on fast main memory languages such as C, C++ or Java. However, performing the classification directly in the database also presents advantages. Firstly, DBMSs balance well with large datasets by nature. Secondly, in-base classification tools avoid developing superfluous or redundant functions. Indeed, many operations required by data mining tasks such as sorting or counting are already offered by the database query language. Finally, such functionalities would enhance the database user productivity: a developer should not create his own ad-hoc solution each time he needs classification tools. This paper proposes two classification algorithms for the functional and object oriented DBMS.

Many methods have been introduced to perform classification. This work is based on Naïve Bayes, which shows high accuracy despite its simplicity [1]. Naïve Bayes is a generative technique. This means that it produces probability distributions for each class of items to be predicted. Consider for instance a training set based on some individuals whose genders and sizes are known. If the classes are based on the genders, two distributions will be built: one represents the gender "male", the other "female". Classifying an individual whose size is known and whose gender is unknown means comparing the probabilities that it is a male or a female given his size. This is performed thanks to Bayes' theorem. Naïve Bayes is more a generic technique than a particular algorithm. Indeed, many variant have been introduced.

2. LITRATURE SURVEY

Knowledge discovery is a process of extraction of useful information from a database [2]. According to , the past and current research in this field can be categorized in 1) investigating new methods to discover knowledge and 2) integrating and scaling these methods[3]. This work focuses on the last aspect.

"First generation" of data mining tools depends on a less or more loose connection between the DBMS and mining system: a front end developed with any language embeds SQL select statements and copies their results in main memory. Then, one stake for database research is to optimize the data retrieving operations and permit fast in-base treatments, in order to tighten the connection between the front end and the DBMS [4].

Efficient use of a query language is non-trivial, and could bring up good performance and usability enhancements. For instance, multiples passes through data using various orders may involve SQL sorting and grouping operations. This can be done with database tuning techniques, such as smart indexing, parallel query execution or main memory evaluation.

Therefore data mining applications should be "SQL aware". Tightening this connection is also one of the reason of object oriented databases (OOD), Turing complete programming languages embedded in most systems, such as PL/SQL (Oracle), and user defined functions developed in another language. A tightly-coupled data mining methodology in which complete parts of the knowledge discovery operations are pushed into the database to avoid context switching costs and intermediate results storage in the application space [5]. The following section deasl with classification operations written directly in structured query language (SQL). Finally, in the software industry, Microsoft SQL Server 2000 initiated in-base data mining classification and rule discovery tools.

Definitely, the cost of memory transfers from the DBMS to the application may cancel all the benefits or profits of executing some operations such as sorting or counting directly in the database. SQL also endures from a lack of expressivity: some operations that could be recognized in one pass with a procedural language may require more with SQL. Therefore, the optimal way is to load the data in main memory with a select statement "once and for all". Handling of the ad-hoc nature of the tasks is a major issue in the field of data mining. Therefore, scaling efforts should not be applied on specific algorithms such as APriori or decision trees [6], but on their basic operations. Improvements for SQL are proposed. At first level, new basic primitives could be developed for operations such as batch or sampling aggregation (multiple aggregations over the same data). The generalization of the CUBE operator is a step in this direction [7]. On a higher level, data mining primitives could be embedded, such as support for association rules [8].

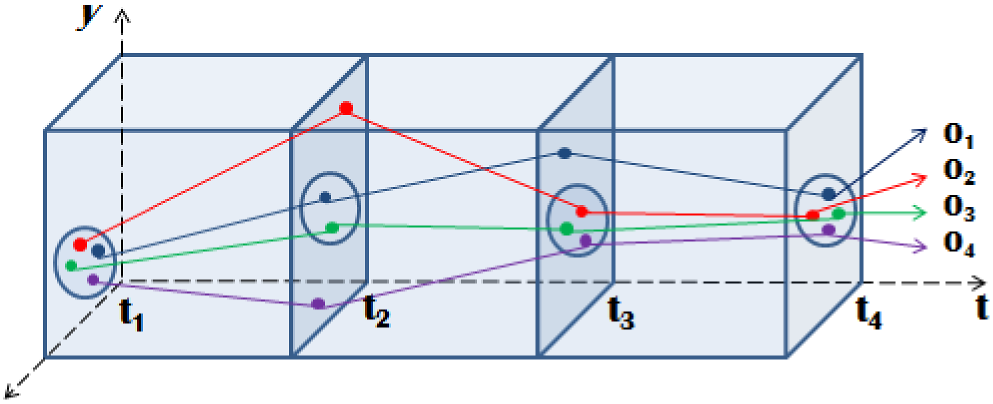

Figure 1: Classification of objects

A long term vision of these principles is exposed in [9]:"Knowledge Discovery Data Management system" is introduced. SQL could be generalized to create, store and manipulate "KDD objects": rules, classifiers, clusters, probabilistic formulas, …. etc. The KDD queries should be optimized and support a closure principle: the result of a query can itself be queried. In this perspective, Data Mining Query Languages have been introduced these past years. Among many others, MSQL and DMQL are representative of this effort [10].

3. PROPOSED SYSTEM ALGORITHM

The classification task is to identify the sex of a person. There are two classes $C = Female$ or $Male$. The data set can be decomposed in two subsets:

Consider for instance the following data set describing the features of 5 individuals:

Item	**Hair**	**Size**	**Sex**
X_1	Short	176	Male
X_2	Short	189	Male
X_3	Long	165	Female
X_4	Short	175	Female
Y	Short	174	?

The items which classes are known *$(X1, Male)$, $(X2, Male)$, $(X3, Female)$, $(X4, Female)$+. They constitute the training (or learning) set.

An item $Y = (Short, 174)$ which classes is unknown. It is a test item.The goal of supervised learning is to infer a classifier from the training set and apply it on the test item to predict its class.The training set will be referred to as *$(Xi, c$i)+i∈,1,p- with Xi = (x1, x2, x3, … , xn) and $ci \in C$. Each component will be referred to as attribute, or feature, taking its value in a space defined by the classification problem (either continuous or discrete). The test item will be represented by Y = (y1, y2, y3, … , yn). The classifier returns its class cY Supervised learning is a wide field of computer science and applied mathematics [11]. Among many others, NN, support vector machines, decision trees and nearest neighbor algorithms have been well established techniques. This paper is based on Naïve Bayes (NB). The following reasons justify this choice:

- NB-based techniques are usually very simple. These techniques are based on basic numeric operations, which makes them well suited to a DBMS implementation
- NB can deal with any kind of data (continuous or discrete inputs)
- NB is known to be robust to noise (in the data or in the distribution estimation) and high dimensionality data [2].

Presentation of Naïve Bayes

Xk is the random variable representing the kth feature of an item. C is the random variable describing its class. For readability's sake, $(Xk = ak)$ will be abbreviated as (ak), ak being a constant expression. Similarly, $(C = c)$ will be abbreviated as (c)

Procedure

With Naïve Bayes, classifying an data item $(y1, y2, y3, \ldots, yn)$ consists in computing $P(ci| y1 \wedge y2 \wedge \ldots \wedge yn)$ for each class ci. The class giving the highest score will be selected. However, this probability can generally not be calculated as such. Naïve Bayes classification based on the assumption that each attribute is conditionally independent to every other attributes,:

(1) $(ak \mid c \wedge al) = (ak \mid \ )$ with $k \neq l$ and ak, al , c constant expressions.Under this assumption, $(ci| y1 \wedge y2 \wedge \ldots \wedge yn) \approx (ci)(y1|ci)P(y2|ci) \ldots P(yn|ci)$ for each class ci. This simplification is fundamental.

Therefore, learning with Naïve Bayes consists in:

- Estimating the prior distributions of the classes, e.g. the probability of occurrence of each class P □ $(C = c)$
- Approximating the distributions of the features given each class P □ $(Xk = ak|ci)$ (in the example, the distribution of sizes for males is one of these). The choice of the distribution approximation method depends on the task. For instance, a Normal distribution could be fitted over numerical data. Counting the occurrences of the values of Xk in a frequency histogram is often a good solution for categorical values.

Example

With the previous example, five distributions will be inferred from the learning data:

- The prior distribution *P* □ (Sex). This distribution is easily estimated by counting the number of items in each class: 0.5 for each gender
- The conditional probability distributions P □ (Hair|Sex = Male) and P □ (Hair|Sex = Female). As Male is nominal, these distributions can also be approximated by counting p(Hair|Sex = Female) = 1/2 = 0.5 as one female out of two female has short hair in the training set and p(Hair|Sex = Male) = 2/2 = 1 as every male has short hair.
- P (Hair|Sex = Male) and P □ (Hair|Sex = Female). If the attribute “Size” is supposed to be continuous, counting the occurrence frequency of each distinct value does not make sense. Instead, a continuous distribution is fitted above the feature values for the training items of each class. In this case, it seems reasonable to approximate the distribution of sizes inside each class by a Gaussian distribution. It could have been another distribution: this is an alternative based on prior knowledge. To achieve this, the mean and standard deviation of the sizes are computed separately for the female and male items.

4. CONCLUSION

Knowledge discovery provides the technology to analyze mass volume of data and detect hidden patterns in data to convert raw data into valuable information. This paper mainly focused on the object prediction using Naïve Bayesian classification. Naïve Bayesian classification algorithm can also be used in Bug Fix Time Prediction Model and try to improve the software quality. It estimates the time and effort needed during bug triaging. The issues and applications are applicable to banking industries, manufacture industries, retail industries and so on.The future work focuses on improving the performance of prediction model.

REFRENCES

[1] John F. Roddick and Myra Spiliopoulou, "A Bibliography of Temporal, Spatial and Spatio-Temporal Data Mining Research", SIGKDD Explorations. 1999 ACM SIGKDD, Volume 1, Issue 1 – p 34, June 1999.

[2] Mu-Chen Chen, Long-Sheng Chen, Chun-Chin Hsu, Wei-Rong Zeng, "An information granulation based data mining approach for classifying imbalanced data.", Information Sciences 178 3214–3227, 2008.

[3] R. H. Güting, V. A. Teixeira, and Z. Ding, “Modeling and querying objects in networks,” VLDB, pp. 165-190, 2006.

[4] V. Ferreira, and M. Véras, “Uma implementação da Álgebra para Pontos Móveis usando Postgresql (in Portuguese),” CONNEPI2010, 2010.

[5] C. Junghans, and M. Gertz, “Modeling and Prediction of Moving Region Trajectories,” ACM 978, 2010.

[6] R. H. Guting, M. H. Bohlen, M. Erwig et al., “A Foundation for Representing and Querying data Objects,” Language, pp. 1-37, 2000.

[7] D. S. Cotrim, and J. Campos, “Representação das Características do Movimento de Objetos Móveis em Mapas Estáticos (in Portuguese),” Symposium A Quarterly Journal In Modern Foreign Literatures, 2007.

[8]"BerlinMOD Site," [Online]. Available: http://dna.fernuni-hagen.de/secondo/BerlinMOD/BerlinMOD.html/,2011.

[9] "SECONDO Site," [Online]. Available: http://dna.fernuni-hagen.de/Secondo.html/,2011.

[10] C. Düntgen, T. Behr, and R. Hartmut Güting, "Assessing Representations for data Object," INFORMATIK BERICHTE, 2010.

[11] F. Giannotti, and P. D., “Mobility, Data Mining and Privacy: A Vision of Convergence”, Berlin Heidelberg: Springer-Verlag, 2008.

.