Instance Selection

Huan Liu
Arizona State University, USA

Lei Yu
Arizona State University, USA

INTRODUCTION

The amounts of data become increasingly large in recent years as the capacity of digital data storage worldwide has significantly increased. As the size of data grows, the demand for data reduction increases for effective data mining. Instance selection is one of the effective means to data reduction. This article introduces basic concepts of instance selection, its context, necessity and functionality. It briefly reviews the state-of-the-art methods for instance selection.

Selection is a necessity in the world surrounding us. It stems from the sheer fact of limited resources. No exception for data mining. Many factors give rise to data selection: data is not purely collected for data mining or for one particular application; there are missing data, redundant data, and errors during collection and storage; and data can be too overwhelming to handle. Instance selection is one effective approach to data selection. It is a process of choosing a subset of data to achieve the original purpose of a data mining application. The ideal outcome of instance selection is a model independent, minimum sample of data that can accomplish tasks with little or no performance deterioration.

BACKGROUND AND MOTIVATION

When we are able to gather as much data as we wish, a natural question is "how do we efficiently use it to our advantage?" Raw data is rarely of direct use and manual analysis simply cannot keep pace with the fast accumulation of massive data. Knowledge discovery and data mining (KDD), an emerging field comprising disciplines such as databases, statistics, machine learning, comes to the rescue. KDD aims to turn raw data into nuggets and create special edges in this ever competitive world for science discovery and business intelligence. The KDD process is defined (Fayyad *et al.*, 1996) as *the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data*. It includes data selection, preprocessing, data mining, interpretation and evaluation. The first two processes (data selection and preprocessing) play a pivotal role in successful data mining (Han and Kamber, 2001). Facing the mounting challenges of enormous amounts of data, much of the current research concerns itself with scaling up data mining algorithms (Provost and Kolluri, 1999). Researchers have also worked on scaling down the data - an alternative to the algorithm scaling-up. The major issue of scaling down data is to select the relevant data and then present it to a data mining algorithm. This line of work is in parallel with the work on algorithm scaling-up and the combination of the two is a two-edged sword in mining nuggets from massive data.

In data mining, data is stored in a *flat file* and described by terms called *attributes* or *features*. Each line in the file consists of attribute-values and forms an *instance*, also named as a *record, tuple*, or *data point* in a multi-dimensional space defined by the attributes. Data reduction can be achieved in many ways (Liu and Motoda, 1998; Blum and Langley, 1997; Liu and Motoda, 2001). By selecting features, we reduce the number of columns in a data set; by discretizing feature-values, we reduce the number of possible values of features; and by selecting instances, we reduce the number of rows in a data set. We focus on instance selection here.

Instance selection reduces data and enables a data mining algorithm to function and work effectively with huge data. The data can include almost everything related to a domain (recall that data is not solely collected for data mining), but one application is normally about using one aspect of the domain. It is natural and sensible to focus on the relevant part of the data for the application so that search is more focused and mining is more efficient. It is often required to clean data before mining. By selecting relevant instances, we can

usually remove irrelevant, noise, and redundant data. The high quality data will lead to high quality results and reduced costs for data mining.

MAJOR LINES OF RESEARCH AND DEVELOPMENT

A spontaneous response to the challenge of instance selection is, without fail, some form of sampling. Although it is an important part of instance selection, there are other approaches that do not rely on sampling, but resort to search or take advantage of data mining algorithms. In the following, we start with sampling methods, and proceed to other instance selection methods associated with data mining tasks such as classification and clustering.

Sampling Methods

Sampling methods are useful tools for instance selection (Gu, Hu, and Liu, 2001).

$\binom{N}{n}$ *Simple random sampling* is a method of selecting n instances out of the N such that every one of the distinct samples has an equal chance of being drawn. If an instance that has been drawn is removed from the data set for all subsequent draws, the method is called random sampling without replacement. Random sampling with replacement is entirely feasible: at any draw, all N instances of the data set are given an equal chance of being drawn, no matter how often they have already been drawn.

Stratified random sampling The data set of N instances is first divided into subsets of N_1, N_2,…, N_l instances, respectively. These subsets are non-overlapping, and together they comprise the whole data set (i.e., $N_1+N_2,\ldots,+N_l=N$). The subsets are called strata. When the strata have been determined, a sample is drawn from each stratum, the drawings being made independently in different strata. If a simple random sample is taken in each stratum, the whole procedure is described as stratified random sampling. It is often used in applications that we wish to divide a heterogeneous data set into subsets, each of which is internally homogeneous.

Adaptive sampling refers to a sampling procedure that selects instances depending on results obtained from the sample. The primary purpose of adaptive sampling is to take advantage of data characteristics in order to obtain more precise estimates. It takes advantage of the result of preliminary mining for more effective sampling and vise versa.

Selective sampling is another way of exploiting data characteristics to obtain more precise estimates in sampling. All instances are first divided into partitions according to some homogeneity criterion, and then random sampling is performed to select instances from each partition. Since instances in each partition are more similar to each other than instances in other partitions, the resulting sample is more representative than a randomly generated one. Recent methods can be found in (Liu, Motoda, and Yu, 2002) in which samples selected from partitions based on data variance result in better performance than samples selected from random sampling.

Methods for Labeled Data

One key data mining application is classification – predicting the class of an unseen instance. The data for this type of application is usually labeled with class values. Instance selection in the context of classification has been attempted by researchers according to the classifiers being built. We include below five types of selected instances.

Critical points are the points that matter the most to a classifier. The issue was originated from the learning method of Nearest Neighbor (NN) (Cover and Thomas, 1991). NN usually does not learn during the training phase. Only when it is required to classify a new sample does NN search the data to find the nearest neighbor for the new sample and use the class label of the nearest neighbor to predict the class label of the new sample. During this phase, NN could be very slow if the data is large and be extremely sensitive to noise. Therefore, many suggestions have been made to keep only the critical points so that noisy ones are removed as well as the data set is reduced. Examples can be found in (Yu *et al*., 2001) and (Zeng, Xing, and Zhou, 2003) in which critical data points are selected to improve the performance of collaborative filtering.

Boundary points are the instances that lie on borders between classes. Support vector machines (SVM) provide a principled way of finding these points through minimizing structural risk (Burges, 1998). Using a non-linear function φ to map data points to a high-dimensional feature space, a non-linearly separable