Neurocomputing 123 (2014) 60–74

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Incremental filter and wrapper approaches for feature discretization

Artur J. Ferreira [a,c,*], Mário A.T. Figueiredo [b,c]

[a] Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal
[b] Instituto Superior Técnico, Lisboa, Portugal
[c] Instituto de Telecomunicações, Lisboa, Portugal

A R T I C L E I N F O

Article history:
Received 30 April 2012
Received in revised form
25 October 2012
Accepted 29 October 2012
Available online 8 April 2013

Keywords:
Feature discretization
Static discretization
Incremental discretization
Filter
Wrapper
Feature selection

A B S T R A C T

Discrete data representations are necessary, or at least convenient, in many machine learning problems. While feature selection (FS) techniques aim at finding relevant subsets of features, the goal of feature discretization (FD) is to find concise (quantized) data representations, adequate for the learning task at hand. In this paper, we propose two incremental methods for FD. The first method belongs to the *filter* family, in which the quality of the discretization is assessed by a (supervised or unsupervised) relevance criterion. The second method is a *wrapper*, where discretized features are assessed using a classifier. Both methods can be coupled with any static (unsupervised or supervised) discretization procedure and can be used to perform FS as pre-processing or post-processing stages. The proposed methods attain efficient representations suitable for binary and multi-class problems with different types of data, being competitive with existing methods. Moreover, using well-known FS methods with the features discretized by our techniques leads to better accuracy than with the features discretized by other methods or with the original features.

1. Introduction

Datasets with large numbers of features and (relatively) smaller numbers of instances are challenging for machine learning methods. In fact, it is often the case that many features are irrelevant or redundant [1,2] for the learning task at hand (*e.g.*, learning a classifier). Moreover, some features may have minor fluctuations which can be irrelevant, or even harmful, for the learning task. These effects may be specially critical with small training sets (with few instances), where these irrelevancies/redundancies are harder to detect. In these scenarios, the performance of machine learning and data mining tasks, in terms of both time and accuracy, can be improved by preprocessing data with a discretization step. Moreover, some learning algorithms require a discrete representation of the data.

To deal with the problems mentioned in the previous paragraph, several *feature selection* (FS) [3] and *feature discretization* (FD) [4] methods have been proposed. FS aims at reducing the number of features, often allowing the learning algorithms to obtain classifiers with better performance. FD techniques lead to feature representations containing enough information for the learning task at hand, while ignoring minor fluctuations. A byproduct of FS and FD is a reduction of the memory required to represent the data.

FD and FS are topics with a vast literature; see [4–8] for extensive reviews of FD methods and [3,9–12] for a comprehensive coverage and pointers to the literature on FS.

1.1. Our contribution

This paper proposes two incremental FD methods. One is a *filter* approach, since the quality of the discretization is assessed by an unsupervised or supervised relevance function. The other is a *wrapper*, since it requires using a classifier learning algorithm. Both methods yield a variable number of bits per feature and assess the performance of each feature as the discretization is carried out and can be coupled with any static (unsupervised or supervised) discretization procedure. In the supervised wrapper method, if the original representation is better than the discretized one, in terms of classification error, the original one is kept; this leads to a hybrid data representation containing both original and discretized features, with a variable number of bits.

1.2. Organization of the text

Section 2 reviews unsupervised and supervised FD and FS techniques. Section 3 presents the proposed FD methods, discusses the motivation behind these methods, and their extensions to include FS before or after the discretization stage. Section 4 reports a comprehensive experimental evaluation of our methods in comparison with other techniques. Finally, Section 5 presents some concluding remarks and future work directions.

* Corresponding author at: Instituto Superior de Engenharia de Lisboa, ADEETC - Gab. 16, 1959-007 Lisboa, Portugal. Tel.: +351 21 831 72 16.
E-mail addresses: arturj@cc.isel.ipl.pt, arturj@isel.pt (A.J. Ferreira).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.10.036

2. Background on feature discretization and selection

This section reviews some FD and FS techniques that have proven effective for many learning problems. This description is far from exhaustive, as FD and FS are two topics with a long research history. The reader is referred to [4–8], for reviews of FD methods and to [3,9–12] concerning FS.

2.1. Feature discretization

The main goals of FD techniques are to reduce the amount of required memory to represent the data and to obtain concise representations, ignoring minor fluctuations. As a consequence, FD usually leads to both an improvement in classification accuracy and reduction of training time, as compared to the use of the original features [4,5,7,13].

FD techniques can be categorized along the following five axes: supervised or unsupervised; static or dynamic; global or local; top-down or bottom-up; direct or incremental. Supervised methods use class labels, whereas unsupervised ones do not. Static methods perform a single discretization that passes over the data, treating each feature independently of the others, while dynamic methods discretize all features jointly, thus taking into account interdependencies. Global techniques discretize the entire feature space, whereas local ones are based on a decision mechanism, such as a tree. Regarding how the sequence of binary codes that represent the discrete feature values are constructed, the top-down versus bottom-up categorization refers to the splitting and merging actions, respectively. Finally, in direct FD methods, one decides *a priori* on the number of bits per feature, whereas incremental methods start with a coarse discretization pass for all features and subsequently allocate more bits to each feature, guided by some criterion.

In the FD literature, the quality of a discretization has been assessed by several indicators; the two most common are

- Complexity—given by the number of discretization intervals or, equivalently, the number of bits needed to represent them. The training time of a learning algorithm working on discrete data is usually proportional to the number of discretization intervals.
- Classification accuracy—an adequate FD technique is expected to lead to better classification accuracy, as compared to the use of the original features.

2.1.1. Unsupervised methods

This subsection reviews static unsupervised scalar FD techniques. In this context [4], the most common techniques are:

- *Equal-interval binning* (EIB), which performs uniform quantization with a given number of bits per feature;
- *Equal-frequency binning* (EFB) [14], which obtains a non-uniform quantizer where, for each feature, the number of occurrences in each interval is the same;
- *Proportional k-interval discretization* (PkID) [15], which adjusts the number and size of the discretization intervals to the number of training instances, thus seeking a trade-off between bias and variance of the class probability estimate of a naïve Bayes (NB) classifier [16,17].

EIB is simple and easy to implement, but very sensitive to outliers, thus it may lead to inadequate discrete representations. In EFB, the quantization intervals are smaller in regions where there are more occurrences of the feature values. EFB is thus less sensitive to outliers than EIB. In both the direct top-down EIB and EFB methods, one can specify *a priori* the number of discretization intervals. In contrast, the PkID method computes the number and size of discretized intervals proportionally to the number of training instances, seeking a trade-off between the granularity of the intervals and the expected accuracy of probability estimation. Given a numeric feature for which the number of observed instances is ν, it is discretized into $\sqrt{\nu}$ intervals, with $\sqrt{\nu}$ instances in each interval.

Although it is expected that supervised FD leads, in principle, to better classifiers than unsupervised FD, it has been found that unsupervised FD performs well in conjunction with several classifiers; in particular, EFB in conjunction with NB classification produces very good results [4]. It has also been found that combining EIB or EFB with *support vector machine* (SVM, [18,19]) classifiers lead to good results on microarray data [20]. Experimental results show that, in comparison to EIB and EFB, PkID boosts NB classifiers to a competitive classification performance for lower dimensional datasets, and better classification performance for higher dimensional ones [15].

2.1.2. Supervised methods

Information entropy minimization (IEM) [21], based on the *minimum description length* (MDL) principle [22], is one of the oldest and most used methods for supervised FD. The key idea of using MDL is that the most informative features to discretize are those that are the most compressible. The IEM method is based on the use of the entropy minimization heuristic for discretization of a continuous value into multiple intervals as well as on the idea of constructing small decision trees. IEM follows a top-down approach in the sense that it starts with one interval and splits intervals in the process of discretization.

The *IEM variant* (IEMV) proposed in [23] is also based on the MDL principle, using an entropy minimization heuristic to choose the discretization intervals. In fact, the authors propose a function based on the MDL principle, such that its value decreases as the number of different values for a feature increases. Experimental results show that both IEM and IEMV lead to better decision trees than the previous methods.

The ChiSquare method [24] is a simple and general algorithm that uses the χ^2 statistic to discretize numeric features. The empirical results demonstrate that ChiSquare is effective in FS and FD of numeric and ordinal features respectively.

The so-called Khiops discretization method [25] uses the χ^2 statistic to merge consecutive intervals in order to improve the global dependence measure. The method optimizes the χ^2 criterion globally on the whole discretization domain and does not require any stopping criterion. A theoretical study followed by experiments demonstrates the robustness and the good performance of the method. MODL is another FD method proposed by the same author [26], which builds an optimal Bayesian criterion, introducing the concept of space of discretization models and a prior distribution defined on this space.

An efficient FD algorithm for constructing *Bayesian belief networks* (BBN) was proposed in [27]. The partitioning minimizes the information loss, relative to the number of intervals used to represent each variable. Partitioning can be done only prior to BBN construction or extended for repartitioning during that construction.

The supervised, static, global, top-down, and incremental *class-attribute interdependence maximization* (CAIM) [28] algorithm aims at maximizing the class-attribute interdependence and to generate a (possibly) minimal number of discrete intervals. The experimental results in [28] show that discrete features (attributes) generated by CAIM usually have the lowest number of intervals and the highest class-attribute interdependency, when compared with six other state-of-the-art methods.