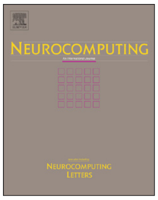
Neurocomputing 138 (2014) 371–382

Contents lists available at ScienceDirect

Neurocomputing

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



Bandit-based local feature subset selection

Mohammad-Hassan Zokaei Ashtiani [a,*], Majid Nili Ahmadabadi [a,b], Babak Nadjar Araabi [a,b]

[a] *Robotics and AI Laboratory, Control and Intelligent Processing Center of Excellence, School of ECE, Faculty of Engineering, University of Tehran, Tehran, Iran*
[b] *School of Cognitive Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran*

ARTICLE INFO

Article history:
Received 17 September 2012
Received in revised form
17 January 2014
Accepted 1 February 2014
Communicated by T. Heskes
Available online 17 February 2014

Keywords:
Local feature subset selection
Monte Carlo tree search
Reinforcement learning

ABSTRACT

In this work we propose a method for local feature subset selection, where we simultaneously partition the sample space into localities and select features for them. The partitions and the corresponding local features are represented using a novel notion of feature tree. The problem of finding an appropriate feature tree is then formulated as a reinforcement learning problem. A value-based Monte Carlo tree search with the corresponding credit assignment policy is devised to learn near-optimal feature trees. Furthermore, the Monte Carlo tree search is enhanced in a way to be applicable for large numbers of actions (i.e., features). This objective is achieved by taking into account a bandit-based explorative policy while having a soft exploitive estimation policy. The results for synthetic datasets show that when local features are present in data, the proposed method can outperform other feature selection methods. Furthermore, the results for microarray classification show that the method can obtain results comparable to the state of the art, using a simple KNN classifier.

1. Introduction

Feature selection has become one of the main challenges of machine learning, especially with the introduction of applications with a huge number of features. Filtering out irrelevant/redundant features can reduce computational costs and enhance the accuracy of classification methods. Furthermore, finding relevant subsets of features can be a goal per se. For example, in microarray classification, discovering the set of genes (i.e., features) responsible for a disease or functionality is of a great importance.

Many methods have been proposed for feature selection. However, most of them tend to select global features, i.e., they select a single subset of features for classification in the whole sample space.[1] Nevertheless, there might be cases in which different subsets of features are the most informative ones for classification in different parts of the sample space. For example, important features in diagnosis of a disease in men and women may be different. This could be the case in microarray classification too: expression levels of different genes (i.e., different features) may be informative for classification of different instances.

Consequently, extracting local features in different localities of the sample space can be useful. An idea to do this is to partition the sample space into different localities, and to select features for each of them. Nonetheless, partitioning the sample space into appropriate localities is a challenge because it is coupled with the problem of feature selection in each locality. In other words, creating a locality is useful only if there exists a distinct set of appropriate features for that locality. As a result, in order to have an efficient method for local feature subset selection, feature selection and locality formation should be done simultaneously.

Furthermore, local feature selection increases the risk of over-fitting. The reason is that in this setting features are selected locally using a limited number samples. Therefore, a noise feature would have a higher chance to be selected compared to the global feature selection case; especially when the number of features is large. A natural way to avoid over-fitting is to prevent localities from getting too small by incorporating a cross-validation process. The other way is to penalize the variation of the selected subset of features along the sample space through regularization. These constraints will decrease the risk of over-fitting, based on the reasonable assumption that neighbor localities are more likely to share features.

In this work we propose a method for local feature subset selection. First of all, local features are represented by the novel notion of feature tree. This unified model enables us to represent the localities and their corresponding features. Moreover, selection of common features for neighbor localities is encouraged in this model, in order to reduce variations of the selected features along the sample space.

[*] Correspondence to: David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1. Tel.: +1 519 888 4567x33602.
E-mail addresses: zokaei@ut.ac.ir (M.-H. Zokaei Ashtiani), mnili@ut.ac.ir (M. Nili Ahmadabadi), araabi@ut.ac.ir (B. Nadjar Araabi).

[1] Note that here our focus is on the feature subset selection methods; nevertheless, there are many local dimensionality reduction methods which find a mapping, rather than an explicit subset of features.

http://dx.doi.org/10.1016/j.neucom.2014.02.001
0925-2312

Afterwards, a top-down approach is used to find an appropriate feature tree. The idea is to split the sample space into localities and select features for each of them simultaneously. The problem is formalized as a reinforcement learning problem. This formulation enables us to take a robust and none-greedy approach to approximate the solution of the coupled problem of splitting the sample space and local feature selection.

One important benefit of utilizing reinforcement learning framework is the possibility of using well-founded decision making methods (e.g., bandit-based planning) to guide the search. In contrast to heuristic search methods, bandit-based planning helps us to have a systematic way of guiding the search in the large space of possible feature trees. Furthermore, we propose an enhanced version of Monte-Carlo tree search in order to be able to deal with relatively large search spaces (i.e., datasets with thousands of features).

In the next section the related works on feature subset selection are reviewed. Afterwards, the proposed method is introduced. The experimental results are given in Section 4. Finally, discussions and conclusions are presented in Section 5.

2. Related work

Traditionally, feature selection methods for supervised learning are divided into three general groups; namely Filter, Embedded and Wrapper approaches. Despite the fact that many methods have been developed for feature subset selection, only a few of them select local features. It is noteworthy that we limit our discussion to feature subset selection methods. Nevertheless, there are other dimensionality reduction methods (e.g., manifold learning methods) that find a local mapping from a high dimensional feature space to a low dimensional space. A good example of these methods – which is both supervised and local – is Local Fisher Discriminant Analysis [1]. These methods do not aim at finding an explicit subset of features for classification. However, our focus is on applications where selection of an explicit subset of features is needed.

Filter approaches to feature selection [2] aim at filtering out irrelevant features and serve as a preprocessor to other machine learning methods. The most common way to evaluate features is to compare them individually to the target function based on a correlation measure. Therefore, they are generally unable to employ complex dependencies among multiple features. In addition, the properties of the utilized classification method are not exploited in these methods. Nevertheless, filter methods are computationally efficient and can be used when the number of features is very large. However, they are global and therefore, a single set of features is selected for the whole input space.

Embedded methods incorporate feature selection in the learning process, resulting in a unified optimization problem. An important group of these methods use regularization (penalty) terms in order to limit the number of utilized features. The well-known least absolute shrinkage and selection operator (lasso) [3] popularized this idea, and other formulations in different settings have been investigated [4–6]. However, the output of these methods is usually a global set of features. Decision tree is another example of an embedded method which performs local feature selection for construction of an appropriate tree.

Wrapper methods [7] employ a more direct way of feature subset selection by searching in the power set of features. The quality of each subset of features is assessed using an evaluation function which reflects the generalization error of the learner. This separation between the optimization method and the evaluation function makes the wrapper methods virtually applicable for all learning machines. However, wrapper methods can be computationally very expensive, as they explore the power set of features and do not utilize specific characteristics of a classifier. Domingos [8] adopted a wrapper approach for local feature selection around each sample, using a greedy sequential feature selection.

Moreover, different search strategies have been proposed for feature selection in order to manage search complexity while reducing the probability of missing a good feature subset. Complete search methods (e.g., branch and bound [9] and beam search [10]) are not usually tractable even for medium-sized feature sets, because the number of feature subsets grows exponentially with the number of features. On the other hand, greedy search methods, such as forward selection and backward elimination, tend to perform a myopic search in the feature space to provide a tractable solution. Stochastic search methods – e.g., the genetic algorithm [11], LVW [12] and random subspace methods [13] – are somewhere between the two extremes, as they balance between tractability and optimality. In addition, Harandi et al. [14,15] proposed a guided stochastic search based on reinforcement learning in order to reduce the search complexity while achieving high performance. Recently, Gaudel and Sebag [16] have used Monte-Carlo tree-based search to find appropriate feature subsets. They have formulated feature selection as a reinforcement learning problem, and solved it approximately using the Upper Confidence Tree (UCT [17]) framework.

Local feature subset selection for unsupervised learning has a rich literature. In biclustering, sample points and features are clustered simultaneously. Therefore, distinct subspaces are selected for each cluster of the samples. The search space of this problem is exponential; therefore, in order to have a tractable solution, the problem is relaxed in various ways, including greedy search, divide-and-conquer, iterative clustering (which alternates between clustering the features and the samples), real-valued approximations of the spectral formulation, etc. (see [18,19] for a survey). Nevertheless, the process of feature selection in these methods is unsupervised.

As illustrated above, there has been limited work on the problem of supervised local feature subset selection. Although many dimensionality reduction methods are local, they do not obtain explicit subsets of features. Moreover, the formation of localities and selection of features is usually greedy in the few methods that perform local feature subset selection. In the next section we introduce our method for local feature subset selection.

3. The proposed method

The main idea of the proposed method is to formulate the problem of local feature subset selection as a sequential decision making problem in which we look for a series of good actions (e.g., splitting the input space into localities or selecting features for localities). The first step to achieve this goal is to represent local features appropriately. The novel notion of feature tree, introduced in Section 3.1, addresses this issue. In Section 3.2, we propose a measure for comparing the goodness of different feature trees. As a result, the problem of local feature subset selection is casted to the concrete problem of finding good feature trees.

After defining the optimization problem, the next step is to propose a method for finding good feature trees. In Section 3.3, we suggest a sequential decision making process to create feature trees. Subsequently, we will be able to formulate the optimization as a Reinforcement Learning (RL) problem in which we look for a near-optimal sequence of actions. Then, the selected sequence of actions can be used to reconstruct the feature tree.

However, the number of states and actions of this RL problem can be very large (e.g., exponential with respect to the number of features). Therefore, the usual RL methods fail when the number of features is more than a few hundreds. In Section 3.4, we propose a novel method for solving the large-scale RL problem