Pattern Recognition Letters 68 (2015) 56–62

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Optimal feature selection for nonlinear data using branch-and-bound in kernel space☆

Matthias Ring*, Bjoern M. Eskofier

Digital Sports Group, Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany

ARTICLE INFO

Article history:
Received 11 March 2014
Available online 24 August 2015

Keywords:
Feature selection
Kernel methods
Branch-and-bound

ABSTRACT

Branch-and-bound (B&B) feature selection finds optimal feature subsets without performing an exhaustive search. However, the classification accuracy achievable with optimal B&B feature subsets is often inferior compared to the accuracy achievable with other algorithms that guarantee optimality. We argue this is due to the existing criterion functions that define the optimal feature subset but may not conceive inherent nonlinear data structures. Therefore, we propose B&B feature selection in Reproducing Kernel Hilbert Space (B&B-RKHS). This algorithm employs two existing criterion functions (Bhattacharyya distance, Kullback–Leibler divergence) and one new criterion function (mean class distance), however, all computed in RKHS. This enables B&B-RKHS to conceive inherent nonlinear data structures. The algorithm was experimentally compared to the popular wrapper approach that has to use an exhaustive search to guarantee optimality. The classification accuracy achieved with both methods was comparable. However, runtime of B&B-RKHS was superior using the two existing criterion functions and even completely out of reach using the new criterion function (about 60 times faster on average). Therefore, this paper proposes an efficient algorithm if feature subsets that guarantee optimality have to be selected in data sets with inherent nonlinear structures.

1. Introduction

Algorithms for feature selection may be grouped into two categories: wrapper and filter approaches [1]. The wrapper approach evaluates a feature subset by estimating the classifier's performance on the feature subset. Feature selection is linked to and guided by the classifier. However, to find the optimal feature subset, the wrapper approach has to examine all possible subsets using an exhaustive search.

The filter approach evaluates a feature subset by computing a criterion function on the feature subset. Feature selection is independent of and detached from the classifier. To find the optimal feature subset, the filter approach can avoid an exhaustive search using the branch-and-bound (B&B) search method [2]. This advantage is facilitated by the usage of monotonic criterion functions.

However, despite this advantage over the wrapper approach, the classification accuracy of optimal B&B feature subsets—similar to feature subsets of other filter approaches—is often inferior compared to the classification accuracy achievable with the wrapper approach (Section 4; [1,3]).

We argue this is due to the criterion functions that have been employed in B&B feature selection so far. The criterion functions were the Bhattacharyya distance, Patrick–Fisher distance, Mahalanobis distance and divergence [2,4–10]. All these functions are defined as integrals over probability density functions [11, Chapter A.2.2]. In the context of B&B feature selection, it is assumed that the data follows a normal distribution, and in this case, the integrals simplify to closed-form terms. Since normal distributions are defined by mean and covariance, the closed-form terms are based on the estimated mean and covariance of the data. Consequently, B&B feature selection searches the optimal feature subset on basis of mean and covariance. Clearly, these two measures may not conceive inherent nonlinear structures that might be present in complex classification problems. See Fig. 1 for an illustration.

Therefore, we propose B&B feature selection in Reproducing Kernel Hilbert Space (RKHS). Kernel methods have been applied to various problems if data contained inherent nonlinear structures, e.g., classification, regression, or data analysis [12]. There are also algorithms for feature selection that work with kernel methods, e.g., [13–15]. The contribution of this paper is to show that B&B feature selection is also able to work with kernel methods. In detail, we show that B&B feature selection in RKHS (B&B-RKHS) finds optimal feature subsets that achieve classification accuracies comparable to the popular wrapper approach. However, B&B-RKHS finds these optimal feature subsets faster than the wrapper approach that has to employ an

☆ This paper has been recommended for acceptance by Prof. M.A. Girolami.

* Corresponding author. Tel.: +49 9131 8528980; fax: +49 9131 303811.
E-mail address: matthias.ring@cs.fau.de (M. Ring).

http://dx.doi.org/10.1016/j.patrec.2015.08.007
0167-8655/

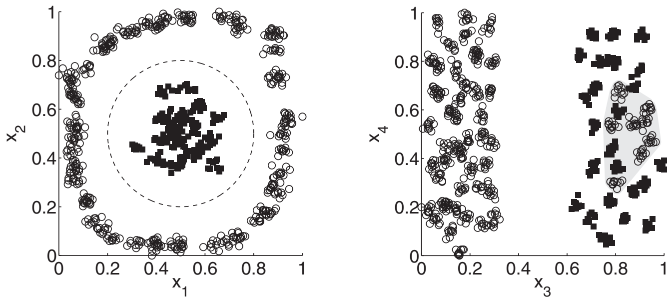

Fig. 1. A synthetic classification problem, consisting of two classes (○, ■) and four features ($x_1, \ldots, x_4$), is illustrated by projections onto the x_1–x_2-plane (left) and the x_3–x_4-plane (right). The classes are perfectly separable in the x_1–x_2-plane, e.g., using the dashed line as decision boundary. In contrast, in the x_3–x_4-plane, some patterns of the ○-class cannot be distinguished from patterns of the ■-class (gray-filled area). Nevertheless, if two out of the four features have to be selected, previous B&B algorithms favor features 3 and 4 over features 1 and 2. This is because, for example, the Bhattacharyya distance J_B is greater for features 3 and 4 ($J_B = 1.09$) than for features 1 and 2 ($J_B = 0.74$). The proposed B&B-RKHS algorithm favors features 1 and 2 over features 3 and 4. This is because the Bhattacharyya distance in RKHS J_B^{RKHS} (polynomial kernel, $b = 1,\ p = 3,\ r = 3$) is greater for features 1 and 2 ($J_B^{RKHS} = 3.2 \times 10^{-3}$) than for features 3 and 4 ($J_B^{RKHS} = 9.0 \times 10^{-6}$). Thus, B&B-RKHS enables the classifier to correctly classify all patterns, in contrast to previous B&B algorithms.

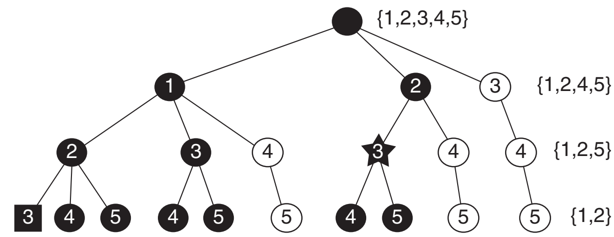

Fig. 2. The B&B search tree to select $d = 2$ out of $D = 5$ features. The root represents the full feature set $F' = \{1, 2, 3, 4, 5\}$. All other nodes represent subsets that are defined by removing the displayed feature from the subset of the parent node. The leaves represent all possible candidate subsets F of size d. Exemplarily, the feature subsets for nodes of the right-most branch are written next to the nodes. Note that the right-most child of every node (white-filled nodes) has only one successor. As an improvement, subsequent white-filled nodes can be replaced by a single node that represents the candidate subset at the end of the branch [4]. The star- and square-formed nodes describe another improvement. If the criterion function at the star-formed node is less than the current bound, monotonicity ensures that the criterion function at the square-formed node is also less than the current bound [5]. Examination of the square-formed node is not necessary.

exhaustive search to guarantee optimality. To the best of our knowledge, the application of B&B feature selection in RKHS has not been explored in the literature so far.

2. Methods

Branch-and-bound feature selection consists of two components that may be exchanged independently. The first component is the search method that defines the order in which feature subsets are examined. The second component is the criterion function that evaluates individual feature subsets.

The following sections describe these two components for B&B-RKHS. First, the search method is described (Section 2.1), and second, two methods are described to evaluate criterion functions in RKHS (Sections 2.2 and 2.3). Finally, this section is concluded with an algorithmic summary of all individual steps in B&B-RKHS (Section 2.4).

2.1. Search method

The proposed B&B-RKHS employs the currently most efficient, non-heuristic B&B search method. Note that there are also heuristic improvements for B&B search [2,7–10]. However, heuristics are not employed because of two reasons. First, the heuristic assumptions are not generally valid for every data set. Second, the efficiency of the heuristic assumptions may not be valid in combination with evaluation in RKHS. This would require detailed experiments which are out of the scope of this paper.

The following sections first introduce the notation and some preliminaries. Then, the three algorithms that together define the currently most efficient, non-heuristic B&B search method are described.

2.1.1. Notation and preliminaries

Given a feature set $F' = \{1, 2, \ldots, D\}$ of size D, B&B finds the optimal feature subset $F^\star \subset F'$ of size $d < D$. The optimal subset $F^\star$ is defined as

$$F^\star = \underset{F \subset F', |F| = d}{\operatorname{argmax}} J(F), \qquad (1)$$

where J is the criterion function. The prerequisite of B&B search is that the criterion function J must be monotonic. This means that J must guarantee for every pair of subsets F_i and F_j, with $F_i \subset F_j \subseteq F'$, that $J(F_i) \leq J(F_j)$ is fulfilled. As described in the next section, this very prerequisite of monotonicity enables B&B to find the optimal subset faster than an exhaustive search.

2.1.2. Branch-and-bound search

Narendra and Fukunaga [2] first applied B&B search to feature selection. They created a search tree in which the root represents the full feature set F' and the leaves represent all possible candidate subsets F of size d. Every node on a path from the root to a leaf represents a feature that is removed from the full feature set F' to obtain the candidate subset F. The search tree is constructed so that the right-most child of every node has only one successor. Hence, for fixed values of d and D, the tree has always the same structure. See Fig. 2 for an example or the work of Narendra and Fukunaga [2] for the formal tree construction algorithm.

To find the optimal subset $F^\star$, the tree is searched with a depth-first strategy from right to left. In Fig. 2, the first candidate subset F of size d is $\{1, 2\}$. The criterion function at this leaf gives the initial bound $B = J(\{1, 2\})$. Then, the depth-first search backtracks to the root and to the next child of the root, i.e., node $\{1, 3, 4, 5\}$. If the criterion function at this node is less than or equal to the current bound, i.e., $J(\{1, 3, 4, 5\}) \leq B$, the subtree below this node is discarded. Monotonicity of J guarantees that the criterion function at the leaves in this subtree is less than or equal to the current bound B. This means the criterion function at the candidate subsets $\{1, 3\}$, $\{1, 4\}$ and $\{1, 5\}$ cannot be better than the criterion function at the candidate subset $\{1, 2\}$. This very situation enables B&B to save examinations of candidate subsets that an exhaustive search would examine, namely $\{1, 3\}$, $\{1, 4\}$ and $\{1, 5\}$. On the other hand, if the criterion function at node $\{1, 3, 4, 5\}$ is greater than the current bound, i.e., $J(\{1, 3, 4, 5\}) > B$, the depth-first search continues. It continues either until the criterion function falls under the current bound at an inner node, or until a leaf is reached and the bound can be updated.

Yu and Yuan [4] extended the algorithm of Narendra and Fukunaga [2]. They replaced the right-most branch of every node (white-filled nodes in Fig. 2) by a single node that represents the candidate subset at the end of the branch. This trick saves unnecessary evaluations of the criterion function at inner nodes. In Fig. 2 for example, the three white-filled nodes in the right-most branch of the root can be replaced by a single node that represents the candidate subset $\{1, 2\}$.

Chen [5] further extended the algorithm of Yu and Yuan [4]. He stored subsets that fall below the current bound. If a later-examined subset F_i is a subset of a stored subset F_j, i.e., $F_i \subset F_j$, the subtree at F_i is discarded without examination. Monotonicity guarantees $B \geq J(F_j) \geq J(F_i)$. In Fig. 2 for example, assume that subset $\{1, 4, 5\}$, i.e., the star-formed node, falls below the current bound. The subset is stored and