Pattern Recognition 43 (2010) 557–568

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.de/locate/pr

U-curve: A branch-and-bound optimization algorithm for U-shaped cost functions on Boolean lattices applied to the feature selection problem ☆

Marcelo Ris [a,*], Junior Barrera [b,*], David C. Martins Jr. [a,**]

[a] Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo-SP, Brazil
[b] Faculdade de Filosofia Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto-SP, Brazil

ARTICLE INFO

Article history:
Received 30 October 2008
Received in revised form
7 July 2009
Accepted 16 August 2009

Keywords:
Boolean lattice
Branch-and-bound algorithm
U-shaped curve
Feature selection
Subset search
Optimal search

ABSTRACT

This paper presents the formulation of a combinatorial optimization problem with the following characteristics: (i) the search space is the power set of a finite set structured as a Boolean lattice; (ii) the cost function forms a U-shaped curve when applied to any lattice chain. This formulation applies for feature selection in the context of pattern recognition. The known approaches for this problem are branch-and-bound algorithms and heuristics that explore partially the search space. Branch-and-bound algorithms are equivalent to the full search, while heuristics are not. This paper presents a branch-and-bound algorithm that differs from the others known by exploring the lattice structure and the U-shaped chain curves of the search space. The main contribution of this paper is the architecture of this algorithm that is based on the representation and exploration of the search space by new lattice properties proven here. Several experiments, with well known public data, indicate the superiority of the proposed method to the sequential floating forward selection (SFFS), which is a popular heuristic that gives good results in very short computational time. In all experiments, the proposed method got better or equal results in similar or even smaller computational time.

1. Introduction

A combinatorial optimization algorithm chooses the object of minimum cost over a finite collection of objects, called search space, according to a given cost function. The simplest architecture for this algorithm, called full search, access each object of the search space, but it does not work for huge spaces. In this case, what is possible is to access some objects and choose the one of minimum cost, based on the observed measures. Heuristics and branch-and-bound are two families of algorithms of this kind. A heuristic algorithm does not have formal guaranty of finding the minimum cost object, while a branch-and-bound algorithm has mathematical properties that guarantee to find it.

Here, it is studied a combinatorial optimization problem such that the search space is composed of all subsets of a finite set with n points (i.e., a search space with 2^n objects), organized as a Boolean lattice, and the cost function has a U-shape in any chain of the search space or, equivalently, the cost function has a U-shape in any maximal chain of the search space.

This structure is found in some applied problems such as feature selection in pattern recognition Duda et al. [5], Jain et al. [7] and W-operator window design in mathematical morphology [8]. In these problems, a minimum subset of features, that is sufficient to represent the objects, should be chosen from a set of n features. In W-operator design, the features are points of a finite rectangle of Z^2 called window. The U-shaped functions are formed by error estimation of the classifiers or of the operators designed or by some measures, as the entropy, on the corresponding estimated join distribution. This is a well known phenomenon in pattern recognition: for a fixed amount of training data, the increasing number of features considered in the classifier design induces the reduction of the classifier error by increasing the separation between classes until the available data become too small to cover the classifier domain and the consequent increase of the estimation error induces the increase of the classifier error. Some known approaches for this problem are heuristics. A relatively well succeeded heuristic algorithm is the sequential floating forward selection (SFFS) [11], which gives good results in relatively small computational time.

There is a myriad of branch-and-bound algorithms in the literature that are based on monotonicity of the cost-function [6,10,14,15]. For a detailed review of branch-and-bound algorithms, refer to Somol and Pudil [13]. If the real distribution of the joint probability between the patterns and their classes were known, larger dimensionality would imply in smaller

☆ Funded by: BZG.
* Corresponding authors.
** Principal corresponding author.
E-mail addresses: mris@ime.usp.br (M. Ris), jb@ime.usp.br (J. Barrera), davidjr@ime.usp.br (D.C. Martins Jr.).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.08.018

classification errors. However, in practice, these distributions are unknown and should be estimated. A problem with the adoption of monotonic cost-functions is that they do not take into account the estimation errors committed when many features are considered (“curse of dimensionality” also known as “U-curve problem” or “peaking phenomena” [7]).

This paper presents a branch-and-bound algorithm that differs from the others known by exploring the lattice structure and the U-shaped chain curves of the search space.

Some experiments were performed to compare the SFFS to the U-curve approach. Results obtained from applications such as W-operator window design, genetic network architecture identification and eight UCI repository data sets show encouraging results, since the U-curve algorithm beats (i.e., finds a node with smaller cost than the one found by SFFS) the SFFS results in smaller computational time for 27 out of 38 data sets tested. For all data sets, the U-curve algorithm gives a result equal or better than SFFS, since the first covers the complete search space.

Though the results obtained with the application of the method developed to pattern recognition problems are exciting, the great contribution of this paper is the discovery of some lattice algebra properties that lead to a new data structure for the search space representation, that is particularly adequate for updates after up-down lattice interval cuts (i.e., cuts by couples of intervals [0,X] and [X,W]). Classical tree based search space representations do not have this property. For example, if the Depth First Search were adopted to represent the Boolean lattice only cuts in one direction could be performed.

Following this Introduction, Section 2 presents the formalization of the problem studied. Section 3 describes structurally the branch-and-bound algorithm designed. Section 4 presents the mathematical properties that support the algorithm steps. Section 5 presents some experimental results comparing U-curve to SFFS. Finally, Conclusion discusses the contributions of this paper and proposes some next steps of this research.

2. The Boolean U-curve optimization problem

Let W be a finite subset, $\mathcal{P}(W)$ be the collection of all subsets of W, $\subseteq$ be the usual inclusion relation on sets and, $|W|$ denote the cardinality of W. The search space is composed by $2^{|W|}$ objects organized in a Boolean lattice.

The partially ordered set $(\mathcal{P}(W), \subseteq)$ is a complete Boolean lattice of degree $|W|$ such that: the smallest and largest elements are, respectively, $\emptyset$ and W; the sum and product are, respectively, the usual union and intersection on sets and the complement of a set X in $\mathcal{P}(W)$ is its complement in relation to W, denoted by X^c.

Subsets of W will be represented by strings of zeros and ones, with 0 meaning that the point does not belong to the subset and 1 meaning that it does. For example, if $W=\{(-1,0),(0,0),(+1,0)\}$, the subset $\{(-1,0),(0,0)\}$ will be represented by 110. In an abuse of language, $X=110$ means that X is the set represented by 110.

A chain $\mathcal{A}$ is a collection $\{A_1,A_2,\ldots,A_k\}\subseteq\mathcal{X}\subseteq\mathcal{P}(W)$ such that $A_1\subseteq A_2\subseteq\cdots\subseteq A_k$. A chain $\mathcal{M}\subseteq\mathcal{X}$ is maximal in $\mathcal{X}$ if there is no other chain $\mathcal{C}\subseteq\mathcal{X}$ such that $\mathcal{C}$ contains properly $\mathcal{M}$.

Let c be a cost function defined from $\mathcal{P}(W)$ to $\mathbb{R}$. We say that c is decomposable in U-shaped curves if, for every maximal chain $\mathcal{M}\subseteq\mathcal{P}(W)$, the restriction of c to $\mathcal{M}$ is a U-shaped curve, i.e., for every $A,X,B\in\mathcal{M}$, $A\subseteq X\subseteq B\Rightarrow\max(c(A),c(B))\geq c(X)$.

Fig. 1 shows a complete Boolean lattice $\mathcal{L}$ of degree 4 with a cost function c decomposable in U-shaped curves. In this figure, it is emphasized a maximal chain in $\mathcal{L}$ and its cost function. Fig. 2 presents the curve of the same cost function restricted to some maximal chains in $\mathcal{L}$ and in $\mathcal{X}\subseteq\mathcal{L}$. Note the U-shape of the curves in Fig. 2.

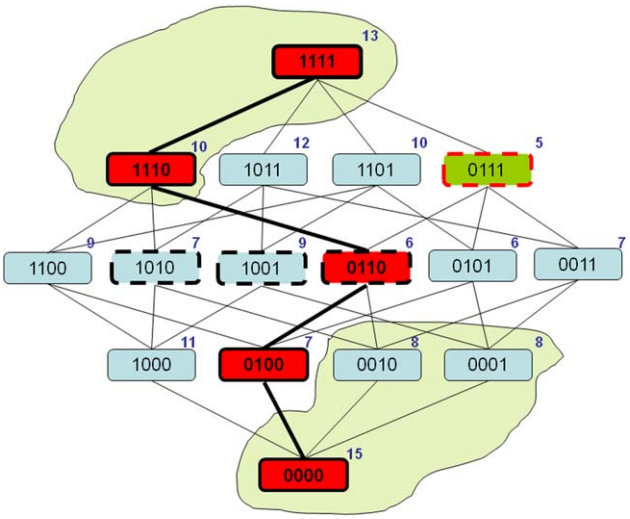

Fig. 1. A complete Boolean lattice $\mathcal{L}$ of degree 4 and the cost function decomposable in U-shaped curves. $\mathcal{X}=\mathcal{L}-\{0000,0010,0001,1110,1111\}$ is a poset obtained from $\mathcal{L}$. A maximal chain in $\mathcal{L}$ is emphasized. The element 0111 is the global minimum element and 0101 is the local minimum element in the maximal chain.

Our problem is to find the element (or elements) of minimum cost in a Boolean lattice of degree $|W|$. The full search in this space is an exponential problem, since this space is composed by $2^{|W|}$ elements. Thus, for moderately large $|W|$, the full search becomes unfeasible.

3. The U-curve algorithm

The U-shaped format of the restriction of the cost function to any maximal chain is the key to develop a branch-and-bound algorithm, the *U-curve algorithm*, to deal with the hard combinatorial problem of finding subsets of minimum cost.

Let A and B be elements of the Boolean lattice $\mathcal{L}$. An *interval* $[A,B]$ of $\mathcal{L}$ is the subset of $\mathcal{L}$ given by $[A,B]=\{X\in\mathcal{L}:A\subseteq X\subseteq B\}$. The elements A and B are called, respectively, the left and right extremities of $[A,B]$. Intervals are very important for characterizing decompositions in Boolean lattices [2,4].

Let R be an element of $\mathcal{L}$. In this paper, intervals of the type $[\emptyset,R]$ and $[R,W]$ are called, respectively, lower and upper *intervals*. The right extremity of a lower interval and the left extremity of an upper interval are called, respectively, lower and upper *restrictions*. Let $\mathcal{R}_L$ and $\mathcal{R}_U$ denote, respectively, collections of lower and upper intervals. The search space will be the *poset* $\mathcal{X}(\mathcal{R}_L,\mathcal{R}_U)$ obtained by eliminating the collections of lower and upper restrictions from $\mathcal{L}$, i.e., $\mathcal{X}(\mathcal{R}_L,\mathcal{R}_U)=\mathcal{L}-\cup\{[\emptyset,R]:R\in\mathcal{R}_L\}-\cup\{[R,W]:R\in\mathcal{R}_U\}$. In cases in which only the lower or the upper intervals are eliminated, the resulting search space is denoted, respectively, by $\mathcal{X}(\mathcal{R}_L)$ and $\mathcal{X}(\mathcal{R}_U)$ and given, respectively, by $\mathcal{X}(\mathcal{R}_L)=\mathcal{L}-\cup\{[\emptyset,R]:R\in\mathcal{R}_L\}$ and $\mathcal{X}(\mathcal{R}_U)=\mathcal{L}-\cup\{[R,W]:R\in\mathcal{R}_U\}$.

The search space is explored by an iterative algorithm that, at each iteration, explores a small subset of $\mathcal{X}(\mathcal{R}_L,\mathcal{R}_U)$, computes a local minimum, updates the list of minimum elements found and extends both restriction sets, eliminating the region just explored. The algorithm is initiated with three empty lists: minimum elements, lower and upper restrictions. It is executed until the whole space is explored, i.e., until $\mathcal{X}(\mathcal{R}_L,\mathcal{R}_U)$ becomes empty. The subset of $\mathcal{X}(\mathcal{R}_L,\mathcal{R}_U)$ eliminated at each iteration is defined from the exploration of a chain, which may be done in down-up or up-down direction. Algorithm 1 describes this process. The direction selection procedure (line 5) can use a random or an adaptative