Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 2, p. 55–56.
http://urlib.net/dpi.inpe.br/ismm@80/2007/06.06.15.12

A branch-and-bound optimization algorithm for U-shaped cost functions on Boolean lattices

Marcelo Ris and Junior Barrera

Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, Brazil
{mris,jb}@ime.usp.br

1. Introduction

A combinatorial optimization algorithm chooses the object of minimum cost in a finite collection of objects, called search space, according to a given cost function. The simplest architecture for this algorithm, called full search, accesses 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 algorithms of this kind. Heuristics does not have formal guaranty of finding the minimum cost object, while branch-and-bound has Mathematical properties that guaranty to find it.

Here, we study a combinatorial optimization problem such that the search space is composed of 2^n objects, organized as a Boolean lattice, and the cost function has a U-shape in a maximal chain of the search space.

This structure is found in some applied problems such as feature selection in pattern recognition and W-operator window design in mathematical morphology. In these problems, a minimum subset of features that is enough to represent the lattice objects should be chosen from a set of n features. In W-operator design the features are points of a rectangle of $\mathbb{Z}^2$, called window. The U-shaped functions are formed by error estimation of the classifiers or W-operators designed. This is a well known phenomena in pattern recognition: for a fixed amount of training data, the increase of features considered in the classifier design induces the diminishment of the classifier error by increasing the separation between classes, until the available data becomes too small to cover the classifier domain and the increase of the estimation error induces the increase of the classifier error. The known approaches for this problem are heuristics. Some relatively well succeeded heuristics are SFS and SFFS [5].

We developed a branch-and-bound solution (the *U-curve algorithm*) that uses the Boolean lattice structure and the U-shaped curves to explore a subset of the search space that is equivalent to the full search. Sophisticated Boolean lattice properties were discovered and applied to design an adequate data structure to represent and update the unexplored part of the search space.

2. The U-curve optimization problem

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

The partially ordered set $(\mathscr{P}(W), \subseteq)$ is a complete Boolean lattice of degree $|W|$ such that: the least and greatest 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 $\mathscr{P}(W)$ is its complement in relation to W, denoted X^c.

We will also represent subsets of W by strings of zeros and ones, with 0 meaning that the point does not belong to the subset and 1 meaning that it does.

A chain contained in $\mathcal{X} \subseteq \mathscr{P}(W)$ is a collection $\mathcal{A} = \{A_1, A_2, , A_k\} \subseteq \mathcal{X}$ such that $A_1 \subseteq A_2 \subseteq \ \subseteq A_k$.

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

Figure 1 shows a complete Boolean lattice $\mathcal{L}$ of degree 4 and 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.

There are a lot of functions describing U-shaped curves that can be used as the cost function [2]. We have used in our work the *mean conditional entropy* [4] but we can list other functions with the same feature: *MAE* (mean absolute error), *CoD* (Coeficient of Determination) [3] and *Bolstered Error* [1].

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 of $2^{|W|}$ elements. Thus, for huge spaces the full search is not feasible.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 2, p. 55–56.
http://urlib.net/dpi.inpe.br/ismm@80/2007/06.06.15.12

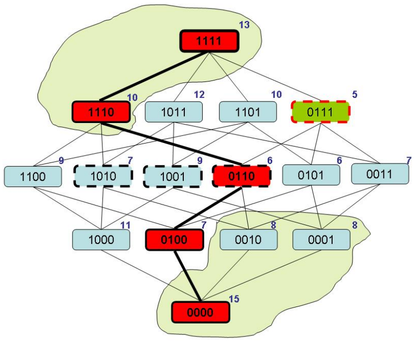

Figure 1. A search space in a Boolean lattice of degree 4. $\mathcal{X} = \mathcal{L} - \{0000, 0010, 0001, 1110, 1111\}$ is a constrained lattice obtained from $\mathcal{L}$.

2.1 The U-curve algorithm

The U-shaped format of the cost function is the key to develop a branch-and-bound algorithm, the *U-curve algorithm* to deal with this problem.

The algorithm consists of an iterative process and the search space is represented by the complete Boolean lattice $\mathcal{L}$. At each iteration, the search space will be constrained by some *restrictions*, that is, a subset of elements that can be discarded from the search process. The restrictions can be grouped in two lists: *upper restrictions* formed by intervals of the type $[R, W]$ and *lower restrictions* formed by intervals $[\emptyset, R]$.

The algorithm looks for *local minimum* elements (elements that has lower and upper adjacent elements with cost bigger than it) by constructing a chain over the constrained lattice. The *chain construction* is an iterated procedure that performs the following steps:

- Choose randomly the chain direction by a defined process: the *Direction Selection Process*. We are going to describe the up direction chain construction (down direction is a dual process)

- Finds a *minimal* element M contained in the current constrained space to be the first element in the chain. This process uses the restrictions lists to obtain this element.

- The procedure to obtain the minimal element is based only in the lower restriction list: if the element obtained is contained in the upper restrictions they can be discarded, the upper restriction updated with it and, a new iteration can begin. The restriction update process is a recursive process that adds an element to the list (when it is already covered by the list) and removes from the list the elements covered by the new element.

- The up direction chain construction process begins with M and flows by inserting an upper adjacent element E (selected randomly from the current constrained search space) until it finds the U-curve condition, that is, E has cost bigger than the last element M inserted.

- We can notice that M is the minimum element of the chain obtained, and let A and B be the elements of the chain adjacents to M with $A \subset M \subset B$ and, by construction, $c(A) \leq c(M) \leq C(B)$. We can prove that any elements C of the constrained search space with $C \subset A$ has cost bigger than A, and any element D of the constrained search space with $B \subset D$ has cost bigger than B. By using this characteristic, the lower and upper restrictions can be updated by A and B respectively.

- In order to prevent visiting the element M more than once, a recursive process called *minimum exhausting procedure* is executed. This process visits all the neighborhood elements of a given element M and turn all of them into *minimum exhausted* elements, that is, all adjacent elements of M in the resulting constrained lattice have cost bigger than M. During this process, the restrictions lists are updated by all the minimum exhausted elements found.

The algorithm stops when the search space is completely processed, that is, when the constrained lattice becomes empty.

References

[1] U. Braga-Neto and E. R. Dougherty, *Bolstered Error Estimation*, Pattern Recognition **37** (2004), 1267-1281.

[2] M. H. DeGroot, *Uncertainty, Information and Sequential Experiments*, Pattern Recognition Letters **33** (1962), no. 2, 404–419.

[3] E. R. Dougherty, S. Kim, and Y. Chen, *Coefficient of determination in nonlinear signal processing*, Signal Processing **80** (2000), 2219-2235.

[4] D. C. Martins Jr., R. M. Cesar Jr., and J. Barrera, *W-operator window design by minimization of mean conditional entropy*, Pattern Anal. Appl. **9** (2006), no. 2, 139–153.

[5] P. Pudil, J. Novovicova, and J. Kittler, *Floating Search Methods in Feature Selection*, Pattern Recognition Letters **15** (1994), no. 11, 1119–1125.