Surrogate-Assisted Monte-Carlo Tree Search in Facility Location and Beyond (Extended Abstract)

Saeid Amiri[1], Danial Dervovic[2], Parisa Zehtabi[3], Michael Cashmore[2]
[1]JPMorganChase AI Research, New York, USA
[2]JPMorganChase AI Research, Glasgow, UK
[3]JPMorganChase AI Research, London, UK
{saeid.amiri, danial.dervovic, parisa.zehtabi, michael.cashmore}@jpmorgan.com

Introduction

Combinatorial problems abound in industry. A persistent issue encountered using search-based solutions is that evaluating particular nodes may be expensive. As an example, organisations frequently adjust their facilities network by opening new branches in promising areas and closing branches in areas where they expect low profits, which may be formulated as a combinatorial search problem. We examine a particular class of facility location problems, where the objective is to minimize the loss of sales resulting from the removal of several retail stores. Estimating sales accurately is expensive and time-consuming. To overcome this challenge, we leverage Monte-Carlo Tree Search (MCTS) (Kocsis and Szepesvári 2006; Coulom 2006) assisted by a surrogate model that computes evaluations faster. Initial results suggest that MCTS supported by a fast surrogate function can generate solutions faster while maintaining a solution consistent with non-assisted MCTS.

Problem Statement

This problem is a class of facility location problem in which a fixed number of retail stores are going to be closed. There is a city network of $N \in \mathbb{N}$ stores. We seek to remove M stores ($M < N$), that result in minimum forgone sales of the network. Our decision variable is the vector $X \in \{0,1\}^N$ such that $X_j = 1$ if store j remains open and $X_j = 0$ if store j is closed. The objective is expressed as:

$$\text{Minimize } \sum_{j=1}^{N} F_m(\mathbb{1}_N, j) - \sum_{j=1}^{N} F_m(X, j)$$

$$\text{Subject to } \sum_{j=1}^{N} X_j = M$$

where $\mathbb{1}_N$ is a vector of 1s and size N. F_m is an evaluation function that estimates the sales of store j. The sales estimated per store depend not only upon the features of that store, but also on other stores in the network.

Surrogate-Assisted MCTS (SMCTS)

The surrogate assisted MCTS (SMCTS) framework works as follows.

Node representation: In our search tree, a node is identified by the set of candidate stores for removal according to the path from the root. The root node represents no store removal. Each node keeps duplicate attributes V'_s and N'_s for its value and the number of visits in case it goes through the re-evaluation step.

SMCTS has five components: *i.* **Selection:** We use UCB1 (Kocsis and Szepesvári 2006) to select the next node, specifically, $\text{argmax}\{v_s + C\sqrt{N_p/N_s}\}$, where v_s is the value of the node s, N_s the number of times node s is visited, and N_p is the number of visits to the parent node.[1]; *ii.* **Evaluation:** A node can be evaluated using functions F_m and F_s. F_m is the costly main evaluation function, while F_s is an approximate surrogate function that is faster to compute but less accurate compared to F_m; *iii.* **Backup:** The return generated by the main or surrogate evaluation function is backed up to update the values; *iv.* **Expansion:** A node is expanded to its children by removing any of the remaining stores from the network. The number of children expanded is equal to the remaining number of stores in the network; *v.* **Re-evaluation:** Two sibling nodes are re-evaluated if their values fall within each other's error bounds. The SMCTS algorithm is formalised in Algorithm 1. Algorithm 1 requires a surrogate function F_s with the error bound σ_s, (in our case, σ_s is the difference between the Root Mean Squared Error (RMSE) of the F_s and F_m) and a main evaluation function F_m.

The novelty of SMCTS is in the **re-evaluation** step where an occasional refinement of node values is done in order to reduce value errors. It is called when all the children of node s are visited an equal number of times. In that case, **re-evaluation** sorts the values of all children in the subtree (sharing same parent node). We name the values of two adjacent sorted nodes V_{s_i} and $V_{s_{i+1}}$. These values may not be accurate as they have been evaluated using F_s, therefore if $V_{s_{i+1}} - \sigma_s$ is less than $V_{s_i} + \sigma_s$, then these node values need to be updated with F_m.

Preliminary Results

Our evaluation studies the following hypotheses: *i.* The number of surrogate evaluations depends on the surrogate quality. The higher error it has, the more re-evaluation steps

[1]We use v'_s, N'_p, N'_s instead, after the node is reevaluated.

Algorithm 1: Surrogate-assisted MCTS

Input: Surrogate (F_s) and main (F_m) evaluation functions, action set A, root node s_0, error bound σ_s

while Computational budget **do**
  $s \leftarrow s_0$
  while $s.terminal$ is $False$ **do**
    $a \leftarrow Select(s, A)$
    $s \leftarrow s.children[a]$
    if $s.leaf$ is $True$ **then**
      $Expand(s)$
    end if
    $v \leftarrow Evaluate(s, F_s)$
    $Backup(s, v)$
    if $s.leaf$ is $False$ and $s.children$ equally visited **then**
      $Re\text{-}evaluate(s, F_m, \sigma_s)$
    end if
  end while
end while
return Node with the highest value

are needed. *ii.* With a judicious choice of a surrogate function, SMCTS maintains a solution consistent with unassisted MCTS.

Dataset: We use the Iowa Liquor Dataset [2] that contains the daily purchase information for each store in the state of Iowa. We preprocess the data by calculating the total sales at each store in a year and by defining some new features such as the number of stores within 0.5 miles proximity.

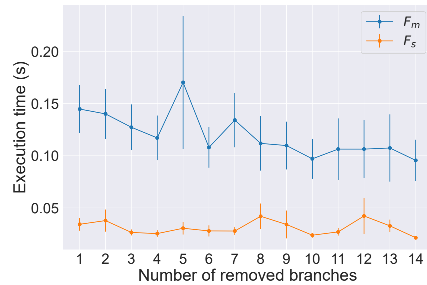
Evaluation Functions: Our main evaluation function F_m is an XGBoost regression model (Chen and Guestrin 2016) that estimates the sales amount for a store. Given a store removal, some features in the dataset need to be recalculated, resulting in a costly evaluation (Figure 1a). To create a surrogate function F_s, we use a subset of the features and train another XGBoost regression model. F_s is less accurate on sales estimation compared to F_m. In our case, F_s has a normalized RMSE of 0.27 and F_m has RMSE 0.16 (both on the test set). Figure 1b shows the ratio of surrogate evaluations to total evaluations. With the increase in the error of the surrogate, we observe an increase in re-evaluations. Figure 1c presents the consistency comparison of the two approaches for various store removals. We use the Sørensen–Dice coefficient[3] to measure the similarity of the results of the two methods. The values are the average of ten counties, randomly sampled from the dataset. We observe that in most cases, SMCTS output is consistent with MCTS.

Conclusion & Future Work

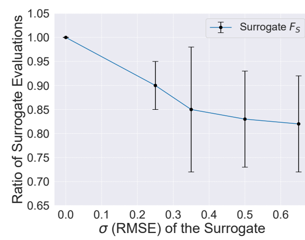
Here we propose MCTS with surrogate functions and apply it to a store closure problem in which the goal is to minimise the total sales loss of a store network. For future work, we will explore the applicability of SMCTS to other combinatorial search problems and different surrogate functions.

[2] https://data.iowa.gov/Sales-Distribution/Iowa-Liquor-Sales/m3tr-qhgy

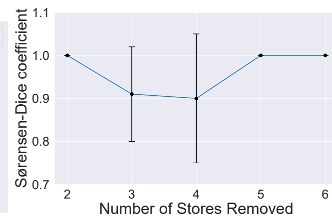
[3] with complete disagreement $\equiv 0$, complete agreement $\equiv 1$.



(a) Execution time of F_s and F_m.



(b) SMCTS surrogate errors.



(c) Agreement between SMCTS and MCTS.

Figure 1: SMCTS evaluation on the store closure problem and execution time comparison.

Disclaimer. This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates (“JP Morgan”), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. KDD ’16.

Coulom, R. 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In *CG*, 72–83. Springer.

Kocsis, L.; and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*. Springer.