Semi-supervised SPO tree classifier based on the DPC framework

Zhou Liang, Liqiong Lu, Junjie Yang*, Weiming Hong, Dong-Meau Chang
School of Computer Science and Intelligence Education, Lingnan Normal University
524048, ZhanJiang, GuangDong, China

lianzhou@lingnan.edu.cn lulq@lingnan.edu.cn 34111677@qq.com hongwm@lingnan.edu.cn morganch@me.com

Abstract

Decision tree is a simple, effective and interpretable algorithm, which has been widely used in different machine learning applications. Recently, the decision tree algorithms were applied to the decision-making problems, one of which was based on the Smart Predict-then-Optimize (SPO) framework and named as the SPO tree algorithm. Compared with other decision tree algorithms, the SPO tree pays more attention on the "quality" of decision rather than minimizing the prediction error and provides better decision and lower model complexity. However, it remains a problem that how to apply the SPO tree to the classification task in semi-supervised learning scenario. To address such a problem, in this paper, the semi-supervised SPO tree classifier is proposed based on the density peak clustering (DPC) framework. The proposed method can utilize the information of labels, densities and distances from data. The experimental results show that, compared with other algorithms, the proposed method has a more robust classification performance in the semi-supervised learning scenario.

1. Introduction

Decision tree is a simple, effective and interpretable algorithm for the machine learning tasks [1–3]. Generally, the decision tree algorithm recursively selects the optimal feature and assigns the data samples to different nodes according to their values of the selected feature. Benefited from the interpretability, the decision tree has been widely used in many real-world applications, *e.g.*, commercial analysis, medical analysis and manufacturing.

In recent years, the decision tree algorithms have been applied to the decision-making problem [4, 5], which can be denoted as an optimization problem containing uncertain input parameters. Such a problem can be solved by a two-stages procedure: first, the values of parameters are predicted; second, an optimal decision-making model is learned based on the predicted parameters. This two-stages procedure is named as the Predict-then-Optimize (PO) framework. However, the optimal decision is not involved in the prediction of parameters. The Smart Predict-then-Optimize (SPO) framework [6] proposed the SPO loss for involving the decision variable and gained a better performance. But solving the SPO loss directly can be a difficult task [6, 7]. Until recently, the SPO tree algorithm [8] was proposed for directly solving the loss function.

Compared with the conventional decision tree framework, the SPO tree algorithm pays more attention on the "quality" of the decision rather than minimizing the prediction error on the input parameter, which can provide better decision and lower model complexity. However, the SPO tree algorithm is currently not able to deal with more complex and practical scenarios, for example, the semi-supervised learning problem [9–15]. Exploring the classification ability of SPO tree algorithm in the semi-supervised learning scenario remains a challenging problem.

To address this problem, based on the Density Peak Clustering (DPC) framework [16], we propose the semi-supervised SPO tree classifier algorithm in this paper. The proposed method retains the advantages of utilizing and directly solving the SPO loss function. Besides, without making specific prior assumption on the data distribution, it can mine the unsupervised information from data only by adding a single integer as input. The main contributions of this paper can be summarized as follows:

- First, we propose the strategy for converting the labels of data to the cost vectors for the two-edges shortest path decision problem, which makes the SPO tree algorithm is able to utilize the supervised information of data for classification;
- Second, based on the DPC framework, we propose the labeled delta distance and a simple, easy-to-use soft-margin strategy for encoding the unsupervised information of data into cost vectors;
- Third, we propose the semi-supervised SPO tree classifier algorithm named as SS-SPOTC and its ensemble

version SS-SPORF. And the experiments on different datasets are made for evaluating the effectiveness of the proposed methods in the semi-supervised classification scenario.

The rest of the paper is organized as follow. Sec. 2 reviews the related algorithms. In Sec. 3, the proposed method is presented. The experimental result is given in Sec. 4. And the conclusion is made in Sec. 5.

2. Related work

2.1. The SPO tree algorithm

The SPO framework is applied for solving the decision-making problems, for examples, estimating the click rates of users for the personalized advertising recommendation, and predicting the asset returns for the portfolio optimization problem. We denote the feasible decision region as $S \subseteq R^d$ and d is the dimension of decision space. The decision-making problem is defined as minimizing $z^*(c) = \min\limits_{w \in S}(c^T w)$, where $c \in R^d$ is the cost vector and $w \in R^d$ is the decision. We define $W^*(c) = \arg\min\limits_{w \in S}\{c^T w\}$ as the optimal decision set for $z^*(c)$, and $w^*(c)$ is the element in $W^*(c)$. Here we assume that the values of $w^*(c)$ and $z^*(c)$ can be calculated by a given cost c when S is specified in an optimization problem with some constraints, *e.g.*, conic, linear or integer constraints.

In the PO framework, the true cost vector is unknown when calculating $w^*(c)$. Thus, the estimation $\hat{c}$ is calculated before solving the optimal decision. The estimation procedure can be formulated as a training problem in the machine learning framework. We denote the training set with costs as $\{(x_1, c_1), (x_2, c_2), \cdots, (x_n, c_n)\}$, where $x \in R^p$ is the data sample with p features. Defining the cost prediction function as $f : R^p \to R^d$ and the lost function as $l(\cdot, \cdot) : R^d \times R^d \to R_+$, the true cost estimation is defined as calculating $\hat{c} = f(x)$ for minimizing $l(\hat{c}, c)$, where c denotes the true cost. Then the estimation of c can be formulated as the following empirical error minimization problem:

$$f^* = \arg\min_f \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), c_i) \tag{1}$$

And the mean square error (MSE) is utilized, shown as:

$$l_{MSE}(\hat{c}, c) = \|\hat{c} - c\|^2 \tag{2}$$

For the PO framework, Eq. (2) only considers the prediction error for cost. The SPO framework replaces it with the SPO loss by measuring the "quality" of decisions for c, which are made on the estimation $\hat{c}$, shown as:

$$l_{SPO}(\hat{c}, c) = \max_{w \in W^*(\hat{c})} \{c^T w\} - z^*(c) \tag{3}$$

Solving Eq. (3) can be a difficult task due to its non-convex and discontinuous characteristics. Recently the SPO tree algorithm was proposed for directly minimizing the loss. Here the recursive partitioning approach of SPO tree is introduced. We assume the training samples are assigned to different leaf nodes, which are denoted as $N_{1:L} = N_1, N_2 \cdots, N_L$. The objective function of SPO tree is defined as:

$$\min_{N_{1:L} \in T} \frac{1}{n} \sum_{l=1}^{L} (\min_{\hat{c}_l} \sum_{i \in N_l} l_{SPO}(\hat{c}_l, c_i)) \tag{4}$$

where $N_{1:L} \in T$ means that the generated model should be satisfied with the structure of decision tree, and $\hat{c}_l$ denotes the estimation cost on leaf node N_l. We denote the average cost of samples on leaf node N_l as $\overline{c_l} = \frac{1}{|N_l|} \sum_{i \in N_l} c_i$. The SPO tree algorithm guarantees that if $\overline{c_l}$ has an unique minimizer for the corresponding decision problem, then $\overline{c_l} = \arg\min\limits_{\hat{c}_l} \sum_{i \in N_l} l_{SPO}(\hat{c}_l, c_i)$. Thus the objective function can be reformulated as:

$$\min_{N_{1:L} \in T} \frac{1}{n} \sum_{l=1}^{L} \sum_{i \in N_l} (c_i^T w^*(\overline{c_l}) - z^*(c_i)) \tag{5}$$

And the SPO tree algorithm is summarized as Algorithm 1:

Algorithm 1 *The SPO tree algorithm*

```
Input: data with costs $D = \{(x_1, c_1), (x_2, c_2), \cdots, (x_n, c_n)\}$
Output: SPO decision tree
1: Initialize $D$ as the root node;
2: repeat
3:     Select a node as the current node;
4:     For the current node, select the optimal feature according to objective function 5 by solving a given decision problem;
5:     The data samples of current node are divided into new nodes according to their values of the selected feature;
6: until Reach the termination condition.
```

2.2. Semi-supervised learning and the DPC algorithm

Obtaining the labeled data can be a challenging task in many real world applications. Thus the semi-supervised learning scenario has gained broader prospects in recent years. It assumes that the training set consists of labeled data D_l and unlabeled data D_u. We denote the number of labeled data as n_l, the number of unlabeled data as n_u, and the total number of data as $n = n_l + n_u$, where $n_l \ll n_u$. The semi-supervised learning algorithms need to mine the unsupervised information, *e.g.*, the graph-based algorithms [11–13] can utilize the similarities between samples along with the labels. For the semi-supervised decision tree algorithms, an algorithm is proposed for incorporating the distributions of variables with labels in hybrid classification framework [14]. And a predictive clustering tree is extended for the semi-supervised framework and applied to multi-target regression [15]. However, such methods do not consider combining the SPO loss and DPC framework for the decision tree algorithm, which is able to utilize the information of labels, densities and distances from data.

The DPC methods [16–20] are designed for the unsupervised learning scenario, which utilize the following clustering definition: the cluster center is denoted as a sample with the highest density in its corresponding cluster, and the samples with higher densities should be far away from this center and not belong to the current cluster. The algorithm utilizes densities and delta distances for clustering. Here we introduce the density defined by the cut-off kernel for simplicity. For the data sample x, its density is defined as:

$$\rho_x = \sum_y \chi(d_{xy} - d_c) \tag{6}$$

where $\chi(x) = 1$ when $x < 0$ and $\chi(x) = 1$ otherwise. d_{xy} is the distance between x and y, and d_c is the cut-off value defining the range of neighborhoods. After calculating the density of x, the delta distance of x is defined as:

$$\delta_x = \min_{y:\rho_y > \rho_x} (d_{xy}) \tag{7}$$

δ_x measures the shortest distance between x and the samples with higher densities. For the data sample with the highest density, its delta distance is defined as $\delta_x = max_y(d_{xy})$. Afterward, the data samples can be regarded as three categories: cluster centers with anomalously large densities and delta distances, regular samples with large densities and small delta distances, and isolated samples with low densities and large delta distances. The algorithm first selects the cluster centers, and then propagates the cluster labels to the remaining.

3. Semi-supervised SPO tree classifier algorithm

3.1. SPO tree algorithm as a classifier

Given dataset $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ and $y \in \{0, 1\}$, the supervised classification scenario requires a classifier $f(x)$ should be trained from labeled data and used for predicting the labels of new samples. Hence, the SPO tree cannot produces an ideal classification output because it is designed for the dataset with costs. For utilizing the labeled data, the two-edges shortest path decision problem is involved.

As shown in Fig. 1, the shortest path problem tries to find a path with the lowest cost from node 1 to 4. We denote the cost as $c = (c_0, c_1, c_2, c_3)$ for the problem. Then it is considered as an optimization problem with constraining $w = (w_0, w_1, w_2, w_3)$ to $(1, 0, 0, 1)$ or $(0, 1, 1, 0)$, which can be easily solved by the optimization toolkit. It is observed that for a given cost c, w is unique for such a problem. It is corresponding to the classification output and meets the requirement of SPO tree algorithm.

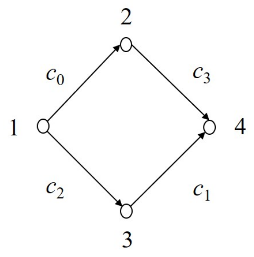

Figure 1. Two-edges shortest path problem.

Hence, to make the SPO tree as a classifier, the labels should be converted to the costs for two-edges shortest path problem. Specifically, for x with label y, we first set c_2, c_3 to 0, and set c_0, c_1 by the converting strategy descried as:

$$c_i = i * (1 - y) + (1 - i) * y \tag{8}$$

where $i \in \{0, 1\}$. This strategy works whether y is 0 or 1. For example, c_0, c_1 are set to 0 and 1 when $y = 0$, implying that the cost of predicting x to class 0 is 0 and to class 1 is 1. The SPO tree produces a classification result by solving the shortest path problem. The prediction of new sample depends on the decision w^l corresponding to its assigned leaf node l, i.e., the algorithm predicts sample to class 0 if $w^l = (1, 0, 0, 1)$ or to class 1 otherwise. The SPO tree classifier (SPOTC) algorithm is summarized as Algorithm 2. Based on the SPOTC, the random forest classifier can also be formed and named as SPO random forest (SPORF) algorithm.

Algorithm 2 *The SPOTC algorithm*

```
Input: data $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$
Output: SPO decision tree classifier
1: Initialize the cost vector $c = (0, 0, 0, 0)$ for each sample in $D$;
2: For each sample $x$, calculate $c_0, c_1$ by Eq. (8) according to its label $y$;
3: Initialize the converted data with costs $D'$ as the root node;
4: repeat
5:     Select a node as the current node;
6:     For the current node, select the optimal feature according to objective function 5 by solving the two-edges shortest path problem;
7:     The data samples of current node are divided into new nodes according to their values of the selected feature;
8: until Reach the termination condition.
```

Eq. (8) only considers utilizing the labeled data, which is actually a hard-margin strategy. Thus, SPOTC and SPORF algorithms are not able to handle the semi-supervised classification tasks. In next section, the soft-margin strategy is proposed for involving the unsupervised and supervised information simultaneously for the SPO tree algorithm.

3.2. Detailed specification for the proposed method

For the semi-supervised learning scenario, the similar data samples should have similar outputs. Therefore, we make the following assumptions for the proposed algorithm. First, different classes are separated and have their corresponding peaks that are far away from each other. Second, data samples with larger densities gain higher confidences. Based on such assumptions, the algorithm first distinguishes the data into peaks and regular samples, and then utilizes different converting strategies according to whether they are labeled.

In order to distinguish the peaks, the algorithm first calculates the density and delta distance for each sample. We adopt the nearest neighbor for defining the density because of its simplicity and robustness, which only requires a positive integer k as input. We denote NN_x as the set of k-nearest neighbors for x (x is not included), and the core distance of x is defined as the distance from x to its farthest neighbor in NN_x, shown as:

$$d_x^c = \max_{x' \in NN_x} (d_{xx'}) \tag{9}$$

Then the density of x is defined as:

$$\rho_x = exp\left(-\frac{1}{k} \sum_{x' \in NN_x} (d_{xx'}/d_{mc})^2\right) \tag{10}$$

where $d_{mc} = \frac{1}{n}\sum_x d_x^c$ is denoted as the average core distance of data. Afterwards the delta distances are calculated by Eq. (7). For the two-classes classification problem, the algorithm selects the first two samples with the largest densities and delta distances as peaks. Then the unlabeled peak is assigned the same label as its labeled nearest neighbor by the algorithm.

For the regular sample x, the labeled one is converted directly by utilizing Eq. (8). For the unlabeled one, the labeled delta distance is proposed for measuring the minimum distance between x and the labeled sample $\tilde{x}$ with higher density. For class i, the corresponding labeled delta distance of x is defined as:

$$\delta_x^i = \min_{(\tilde{x},\tilde{y}):\rho_{\tilde{x}}>\rho_x,\tilde{y}=i} d_{x\tilde{x}} \tag{11}$$

then the corresponding costs c_0, c_1 of x are calculated by the soft-margin constructing strategy descried as:

$$c_i = (1 - exp(-\delta_x^i)) \cdot (1 - \rho_x) \tag{12}$$

where $i \in \{0, 1\}$. For Eq. (12), c_i is set to 1 if the samples with higher densities than x do not contain the label i.

The reason of utilizing Eq. (11) and Eq. (12) is that, based on the assumptions, the labeled data with large density is considered as the data with high confidence, from which the label should be propagated to the data with low confidence. Eq. (11) tries to find the minimum distance between x and the labeled data with higher confidences in different classes. While Eq. (12) considers the unsupervised information of x and the supervised information of labeled data simultaneously. I.e., $1 - \rho_x$ implies the larger density of x, the lower global cost for making a decision on x. And $1 - exp(-\delta_x^i)$ implies the closer distance between x and the sample with higher confidence within class i, the lower cost for classifying x to class i.

Thus, the semi-supervised SPO tree classifier (SS-SPOTC) algorithm is summarized as Algorithm 3. And the semi-supervised SPO random forest (SS-SPORF) algorithm can be constructed by utilizing the SS-SPOTC as basic classifier.

Algorithm 3 *The SS-SPOTC algorithm*

```
Input: $k$, labeled data $D_l = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{n_l}, y_{n_l})\}$ and
       unlabeled data $D_u = \{x_{n_l+1}, x_{n_l+2}, \cdots, x_n\}$
Output: Semi-supervised SPO decision tree classifier
 1: Set $D = D_l \cup D_u$;
 2: Initialize the cost vector $c = (0, 0, 0, 0)$ for each sample in $D$;
 3: For each $x$, calculate $\rho_x$ and $\delta_x$ by using Eq. (10) and Eq. (7), and
    then calculate $\gamma_x = \rho_x \cdot \delta_x$;
 4: Select the first two samples with the largest $\gamma$ values as peaks
    $x_{p_1}, x_{p_2}$;
 5: if $x_{p_1} \in D_u$ or $x_{p_2} \in D_u$ then
 6:     Assign the label(s) to the peak(s) from its (their) labeled nearest
        neighbor;
 7: end if
 8: Calculate $c_0, c_1$ of $x_{p_1}, x_{p_2}$ by using the Eq. (8);
 9: Sort the remaining samples in a descending order according to their
    densities and set $j = 1$, and then the sorted set is denoted as
    $\{x_1^{'}, x_2^{'}, \cdots, x_{n-2}^{'}\}$
10: repeat
11:     if $x_j^{'} \in D_l$ then
12:         Calculate $c_0, c_1$ of $x_j^{'}$ by using Eq. (8) according to its label $y_j^{'}$;
13:     else
14:         Calculate $c_0, c_1$ of $x_j^{'}$ by using Eq. (11) and Eq. (12);
15:     end if
16:     $j = j + 1$;
17: until $j > n - 2$
18: Initialize the converted data with costs $D^{'}$ as the root node;
19: repeat
20:     Select a node as the current node;
21:     For the current node, select the optimal feature according to ob-
        jective function 5 by solving the two-edges shortest path problem;
22:     The data samples of current node are divided into new nodes ac-
        cording to their values of the selected feature;
23: until Reach the termination condition.
```

Due to the limitation of labeled data in the semi-supervised scenario, the situation assigning the same label to different unlabeled peaks might happen and decline the performance of algorithm. To deal with such a situation, the algorithm performs a robust assignation strategy for the unlabeled peaks. Specifically, denoting x_{p_1}, x_{p_2} as the selected peaks and x_{p_1} has the highest density. If one of them has true label y, then the unlabeled one is assigned to label $1 - y$. If both of them are unlabeled, x_{p_1} is first assigned the label y_{p_1} and then the label of x_{p_2} is reset to $1 - y_{p_1}$.

4. Experiment

4.1. Experimental setup

To evaluate the performances of the proposed methods, we use the following algorithms for comparison:

- Decision tree classifier (DTC) algorithm, the CART decision tree algorithm with gini index is utilized;
- Random forest (RF) algorithm, it combines multiple DTC algorithms as the ensemble classifier;
- SPO tree classifier (SPOTC) algorithm, it classifies the data according with the two-edges shortest path problem;
- SPO random forest (SPORF) algorithm, it combines multiple SPOTC algorithms as the ensemble classifier;
- Label propagation (LP) algorithm, it utilizes the similarity and label information to construct the graph and performs "label propagation" for classification.

The DTC, RF, SPOTC and SPORF algorithms are considered as the supervised classifiers, while the LP algorithm is the semi-supervised classifier. In the experiment, the classification accuracy is utilized as the measurement for evaluating the classification performance.

For calculating the distances between data samples, the euclidean distance is utilized after standardizing the data in the experiment. To reduce the effect of the overfitting, the maximum depth for all the decision tree algorithms is set to 5, and the minimum number of samples within the leaf is set to 20. The number of decision tree classifiers is set to 10 for the RF, SPORF and SS-SPORF algorithms. For the SS-SPOTC and SS-SPORF algorithms, the setup for k is descried as follows. First we select a percentage p from 1% to 8% by increasing 1% each time, then p is multiplied by the number of data, i.e., $k = p \cdot N_{tra}$ and N_{tra} denotes the number of training data. Here we reset $k = 1$ if the value of k is less than 1 when using $k = p \cdot N_{tra}$. The best results are reported by setting different values of k for the SS-SPOTC and SS-SPORF algorithms. For the LP algorithm, the RBF kernel is utilized and the gamma parameter is selected in $\{0.1, 1, 10\}$, and then the best result is reported.

4.2. Experiment on real-world data

4.2.1 Data description

The UCI datasets are utilized, which include: breast cancer, ionosphere, heart, iris and wine datasets. The number of data n_d, the number of features n_f and the number of classes n_c for each dataset are listed in Tab. 1.

Table 1. The UCI datasets utilized in the experiment.

name	n_d	n_f	n_c
breast cancer	569	30	2
ionosphere	351	34	2
heart	270	13	2
iris	150	4	3
wine	178	13	3

For the multi-classes datasets, i.e., the iris and wine datasets, they are transformed into two-classes classification tasks by selecting one of the three classes as the positive class, and the remaining are considered as the negative class. The set of labels is denoted as $\{0, 1, 2\}$, then the multi-classes dataset can be formed as 3 tasks, which are named by dataset_positive class_vs_others respectively.

For each task, the dataset is randomly partitioned into the training and testing set with equal (or basically equal) size. In order to evaluate the influence of labeled data on the algorithms, we report the accuracy on testing set by setting different masked sizes for the training set, which are set as 0.0 (supervised scenario), 0.35, 0.65 and 0.95 (95% of the training samples are considered as the unlabeled data).

4.2.2 Experimental results

The classification results are shown in Tab. 2 and Tab. 3. It can be observed that, when masked size is set to 0.0, the SS-SPORF and SS-SPOTC algorithms have similar performances with the SPORF and SPOTC respectively. With the reduction of labeled samples, the accuracies of SPOTC, SPORF, DTC and RF algorithms perform the decreasing trends in most tasks, while the SS-SPOTC, SS-SPORF algorithms still achieve the stable classification results. Specially, in some of the tasks, compared with the results obtained in the supervised scenario, the proposed algorithms perform even better when masked sizes are set to 0.35, 0.65 and 0.95, which reveals the effectiveness of proposed strategies for mining the unlabeled data. It can also be observed that, the proposed algorithms have the comparable or better classification performances compared with the LP algorithm in most tasks when masked size is set to 0.95. Moreover, they outperform the supervised decision tree algorithms in the semi-supervised learning scenario.

Table 2. The experimental results on the real-world datasets (two-classes).

	masked size	SS-SPORF	SS-SPOTC	LP	SPORF	SPOTC	RF	DTC
breast cancer	0.0	0.923	0.912	**0.954**	0.923	0.912	0.944	0.912
	0.35	0.951	0.947	**0.958**	0.926	0.888	0.940	0.888
	0.65	**0.944**	0.891	0.940	0.916	0.912	0.912	0.898
	0.95	**0.909**	**0.909**	0.825	0.611	0.611	0.611	0.611
ionosphere	0.0	**0.903**	0.892	0.881	**0.903**	0.892	0.892	0.886
	0.35	**0.909**	0.903	0.864	0.841	0.864	0.710	0.869
	0.65	**0.909**	**0.909**	0.858	0.830	0.795	0.648	0.812
	0.95	**0.847**	0.841	0.710	0.648	0.648	0.648	0.648
heart	0.0	0.756	0.726	0.756	0.756	0.726	**0.785**	0.726
	0.35	**0.778**	0.741	0.748	**0.778**	0.652	0.748	0.652
	0.65	**0.756**	0.719	0.733	0.659	0.622	0.489	0.622
	0.95	**0.778**	0.756	0.667	0.511	0.511	0.511	0.511

Table 3. The experimental results on the real-world datasets (multi-classes).

	masked size	SS-SPORF	SS-SPOTC	LP	SPORF	SPOTC	RF	DTC
iris_0_vs_others	0.0	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.947	**1.000**
	0.35	**1.000**	**1.000**	**1.000**	0.840	0.813	0.680	0.813
	0.65	**1.000**	**1.000**	**1.000**	0.680	0.680	0.680	0.680
	0.95	**1.000**	**1.000**	**1.000**	0.680	0.680	0.680	0.680
iris_1_vs_others	0.0	0.907	0.747	**0.960**	0.907	0.747	0.680	0.947
	0.35	**0.987**	0.973	0.973	0.733	0.613	0.680	0.600
	0.65	**0.973**	**0.973**	0.893	0.680	0.680	0.680	0.680
	0.95	**0.973**	**0.973**	0.853	0.680	0.680	0.680	0.680
iris_2_vs_others	0.0	**0.960**	**0.960**	0.920	**0.960**	**0.960**	0.840	0.947
	0.35	**0.960**	**0.960**	0.920	0.773	0.747	0.600	0.840
	0.65	**0.960**	**0.960**	0.867	0.600	0.600	0.600	0.600
	0.95	0.933	**0.960**	0.880	0.600	0.600	0.600	0.600
wine_0_vs_others	0.0	0.910	0.910	**0.944**	0.910	0.910	0.933	0.910
	0.35	0.910	0.910	**0.944**	0.910	0.910	0.854	0.910
	0.65	**0.966**	0.910	0.921	0.787	0.213	0.787	0.213
	0.95	**0.685**	**0.685**	0.562	0.213	0.213	0.213	0.213
wine_1_vs_others	0.0	0.921	0.910	**0.944**	0.921	0.910	0.910	0.910
	0.35	0.910	0.944	**0.955**	0.921	0.910	0.618	0.910
	0.65	0.933	**0.955**	0.933	0.618	0.382	0.618	0.382
	0.95	**0.955**	**0.955**	0.685	0.618	0.618	0.382	0.382
wine_2_vs_others	0.0	0.944	0.933	**0.989**	0.944	0.933	0.944	0.921
	0.35	0.966	0.933	**1.000**	0.921	0.933	0.753	0.921
	0.65	0.955	0.944	**0.966**	0.753	0.753	0.753	0.753
	0.95	**0.933**	**0.933**	0.809	0.753	0.753	0.753	0.753

4.2.3 Parameter sensitivity analysis

The input parameter k is crucial for the performances of the proposed algorithms. Here the sensitivity analysis to k is made. The breast cancer dataset is used and the masked sizes are set to 0.35, 0.65 and 0.95 for simulating different semi-supervised scenarios. The value of k is set to $k = p \cdot N_{tra}$ and p is selected from 1% to 8% by increasing 1% each time, and k is reset to 1 if the value of k is less than 1. We report the accuracies of proposed algorithms on the testing set with inputting different values of k. The remaining parameters utilize the same setup descried in Sec. 4.1.

The parameter sensitivity analysis is shown in Fig. 2, Fig. 3 and Fig. 4. The horizontal axis represents the percentage p, which is proportional to k. And the vertical axis represents the accuracy. It can be observed that the SS-SPOTC (blue-square) and SS-SPORF (red-square) algorithms have the stable performances in breast cancer dataset with different values of k. But the performance of SS-SPOTC algorithm decreases slightly at percentages 7% and 8% with setting the masked size to 0.35, as shown in Fig. 2. The possible explanation is that, the SS-SPOTC algorithm does not perform the average decision, which might be unstable compared with the SS-SPORF algorithm.

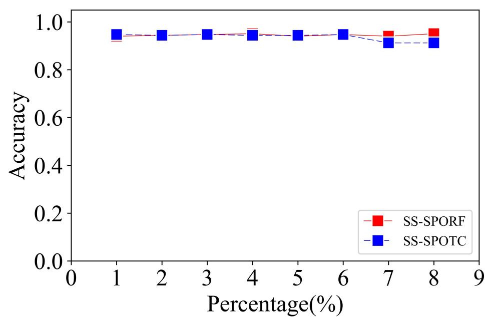

Figure 2. Sensitivity analysis on the breast cancer dataset, the masked size is set to 0.35.

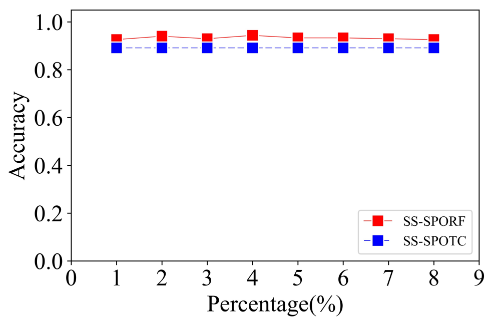

Figure 3. Sensitivity analysis on the breast cancer dataset, the masked size is set to 0.65.

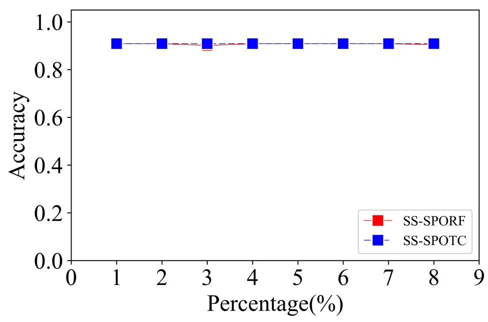

Figure 4. Sensitivity analysis on the breast cancer dataset, the masked size is set to 0.95.

5. Conclusions

The SPO tree algorithm is a new decision tree method designed for the decision-making problem. However, it can not handle the semi-supervised classification scenario. In this paper, the label-to-cost converting strategy is proposed, and the SPO tree can be treated as a classifier by involving the two-edges shortest path problem. Based on the DPC framework, the soft-margin strategy and the semi-supervised SPO tree methods are proposed, which can utilize the information of labels, densities and distances from data. The experimental results demonstrate that the proposed methods gain the robust performances in the semi-supervised learning scenario.

6. Acknowledgment

This research is supported by the Project of Educational Commission of Guangdong Province of China (2019KQNCX071), the Innovation Project of Educational Commission of Guangdong Province of China (2021KTSCX065) and the Natural Science Talents Special Project of Lingnan Normal University (ZL2021015).

References

[1] Zhi-Hua Zhou and Ji Feng. Deep forest: Towards an alternative to deep neural networks. In *International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3553–3559, 2017. 1

[2] Haidi Rao, Xianzhang Shi, Ahoussou Kouassi Rodrigue, Juanjuan Feng, Yingchun Xia, Mohamed Elhoseny, Xiaohui Yuan, and Lichuan Gu. Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing*, 74:634–642, 2019. 1

[3] Lev V. Utkin, Maxim S. Kovalev, and Anna A. Meldo. A deep forest classifier with weights of class probability distribution subsets. *Knowledge-Based Systems*, 173:15–27, 2019. 1

[4] Nathan Kallus. Recursive partitioning for personalization using observational data, 2017. 1

[5] Dimitris Bertsimas, Jack Dunn, and Nishanth Mundru. Optimal prescriptive trees. *INFORMS Journal on Optimization*, 1(2):164–183, 2019. 1

[6] Adam N. Elmachtoub and Grigas Paul. Smart "predict, then optimize". *arXiv1710.08005*, 2017. 1

[7] Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 1658–1665, 2019. 1

[8] Adam N. Elmachtoub, Jason Cheuk Nam Liang, and Ryan McNellis. Decision trees for decision-making under the predict-then-optimize framework. In *International Conference on Machine Learning, ICML 2020*, volume 119, pages 2858–2867, 2020. 1

[9] Xiaojun Chen, Guowen Yuan, Feiping Nie, and Zhong Ming. Semi-supervised feature selection via sparse rescaled linear square regression. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):165–176, 2020. 1

[10] Ying Liu, Zhen Xu, and Chunguang Li. Online semi-supervised support vector machine. *Information Sciences*, 439-440:125–141, 2018. 1

[11] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning, ICML 2003*, pages 912–919, 2003. 1, 2

[12] Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. Learning nonparametric kernel matrices from pairwise constraints. In *International Conference on Machine Learning, ICML 2007*, volume 227, pages 361–368, 2007. 1, 2

[13] Jinfeng Zhuang, Ivor W. Tsang, and Steven C. H. Hoi. Simplenpkl: simple non-parametric kernel learning. In *International Conference on Machine Learning, ICML 2009*, volume 382, pages 1273–1280, 2009. 1, 2

[14] Kyoungok Kim. A hybrid classification algorithm by subspace partitioning through semi-supervised decision tree. *Pattern Recognition*, 60:157–163, 2016. 1, 2

[15] Jurica Levatić, Dragi Kocev, Michelangelo Ceci, and Sašo Džeroski. Semi-supervised trees for multi-target regression. *Information Sciences*, 450:109–127, 2018. 1, 3

[16] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. 1, 3

[17] Mingjing Du, Shifei Ding, and Yu Xue. A robust density peaks clustering algorithm using fuzzy neighborhood. *Int. J. Mach. Learn. Cybern.*, 9(7):1131–1140, 2018. 3

[18] Luis Carlos Castro Heredia and Armando Rodrigo Mor. Density-based clustering methods for unsupervised separation of partial discharge sources. *International Journal of Electrical Power & Energy Systems*, 107:224–230, 2019. 3

[19] Seyed Amjad Seyedi, Abdulrahman Lotfi, Parham Moradi, and Nooruldeen Nasih Qader. Dynamic graph-based label propagation for density peaks clustering. *Expert Systems with Applications*, 115:314–328, 2019. 3

[20] Huanqian Yan, Lei Wang, and Yonggang Lu. Identifying cluster centroids from decision graph automatically using a statistical outlier detection method. *Neurocomputing*, 329:348–358, 2019. 3