Oblique decision tree induction by cross-entropy optimization based on the von Mises–Fisher distribution

Ferdinand Bollwein¹ · Stephan Westphal¹

Received: 20 May 2021 / Accepted: 7 January 2022 / Published online: 29 January 2022

© The Author(s) 2022

Abstract

Oblique decision trees recursively divide the feature space by using splits based on linear combinations of attributes. Compared to their univariate counterparts, which only use a single attribute per split, they are often smaller and more accurate. A common approach to learn decision trees is by iteratively introducing splits on a training set in a top–down manner, yet determining a single optimal oblique split is in general computationally intractable. Therefore, one has to rely on heuristics to find near-optimal splits. In this paper, we adapt the cross-entropy optimization method to tackle this problem. The approach is motivated geometrically by the observation that equivalent oblique splits can be interpreted as connected regions on a unit hypersphere which are defined by the samples in the training data. In each iteration, the algorithm samples multiple candidate solutions from this hypersphere using the von Mises–Fisher distribution which is parameterized by a mean direction and a concentration parameter. These parameters are then updated based on the best performing samples such that when the algorithm terminates a high probability mass is assigned to a region of near-optimal solutions. Our experimental results show that the proposed method is well-suited for the induction of compact and accurate oblique decision trees in a small amount of time.

Keywords Oblique decision trees · Cross-entropy optimization · von Mises–Fisher distribution · Classification · Regression

✉ Ferdinand Bollwein
ferdinand.bollwein@tu-clausthal.de

Stephan Westphal
stephan.westphal@tu-clausthal.de

¹ Institute of Mathematics, Clausthal University of Technology,
Erzstraße 1, 38678 Clausthal-Zellerfeld, Germany

1 Introduction

Decision trees are among the most popular classification and regression models in the field of data mining and machine learning. Due to their human comprehensible structure, they are easy to interpret which helps providing insight into the data under consideration. The most popular employed decision tree models are univariate, involving only a single attribute per split. These splits can be interpreted as axis-parallel hyperplanes in the feature space. The restriction to a single attribute simplifies the decision tree induction process and the interpretation of the individual splits. Nevertheless, these trees are often less accurate than other machine learning models. Moreover, they often become extensively large when the data cannot be split adequately by axis parallel hyperplanes which lowers their explanatory value. In these situations, the univariate splits can also be misleading as they are only meaningful in combinations with other splits further down the respective branch. Oblique decision trees overcome these shortcomings by employing splits based on linear combinations of attributes which can be interpreted as affine hyperplanes in the feature space. Although the individual splits are harder to interpret, the reduced size can increase the explanatory value and an expert in the respective domain is able to draw meaningful conclusions from the coefficients of the oblique splits which can be viewed as weights for the individual attributes.

The problem of finding optimal oblique splits for a given splitting criterion, however, is far more complex than finding univariate splits which complicates the task of inducing oblique decision trees in an acceptable amount of time. Motivated by the underlying problem's structure, we develop an efficient cross-entropy optimization (CE) approach for finding near-optimal oblique splits that can be used in the well-known recursive partitioning scheme for inducing oblique decision trees. In our evaluation we show the advantages of our proposed method compared to the univariate recursive partitioning decision tree induction method and the popular OC1 (Murthy et al. 1993, 1994) algorithm for inducing oblique decision trees. Furthermore, we show that it is competitive with other popular prediction models.

2 Oblique decision tree induction

In this work, we address the problem of determining optimal splits for oblique decision tree induction. Given d real-valued attributes (features) $X_1, \dots, X_d$ and a response variable Y taking values in the domain $\text{dom}(Y)$, an oblique decision tree is a binary rooted tree structure for which every inner node is associated with a rule of the form

$$\sum_{j=1}^d a_j X_j + a_{d+1} \geq 0$$

and every leaf node is associated with a response value in $\text{dom}(Y)$. These rules can be interpreted as hyperplanes that divide the feature space into two halfspaces. To predict the response variable of unseen data points $x \in \mathbb{R}^d$, one pursues the unique path from the root node to one of the leaf nodes according to the specified rules and returns the

Fig. 1 Oblique decision tree for the Iris dataset constructed with the CE method

value stored in the leaf node at the end of this path. As an example, Fig. 1 shows an oblique decision tree for the well-known Iris dataset (Fisher 1936) constructed with our proposed CE method.

To learn high quality oblique decision trees one makes use of a training set (X, y) consisting of a matrix of n observations $X \in \mathbb{R}^{n \times d}$ and a vector of responses $y \in \text{dom}(Y)^n$. Throughout this work, we use the notation x_i for $i = 1, \dots, n$ to refer to a specific row of the matrix X , corresponding to the i -th observation of the training set and y_i for $i = 1, \dots, n$ to refer to the response variable of the i -th observation. We follow the commonly applied strategy of greedily dividing the training data in a top-down manner until a stopping criteria is met or no further splitting is possible because the data is indistinguishable. For classification tasks the stopping criterion is usually that all leaf nodes are sufficiently pure, i.e. the response values in the associated subset of training samples are mostly equal. For continuous domains, one usually stops whenever the mean squared error or mean absolute error of the response values of the subset of training samples is below a certain threshold. Another commonly applied strategy is to stop branching along a path when the sample size associated with the terminal node falls below a prior defined threshold. Finally, each leaf node is assigned a response variable. For classification, this is usually the class of the majority of the associated training samples and for regression the mean value or the median of the responses is used.

To determine good splits an adequate splitting criterion has to be defined beforehand. For classification trees, criteria are typically based on impurity measures which are defined as functions $i : \mathcal{P}_m \rightarrow \mathbb{R}_{\geq 0}$ on the standard $(m - 1)$ -simplex $\mathcal{P}_m := \{(p_1, \dots, p_m) \in [0, 1]^m : \sum_{k=1}^m p_k = 1\}$. The most frequently used impurity measures are:

- Classification error: $e(p) = 1 - \max_{k=1, \dots, m} p_k$
- Entropy: $h(p) = - \sum_{k=1}^m p_k \log p_k$
- Gini impurity: $g(p) = 1 - \sum_{k=1}^m p_k^2$

For a classification task with $m \in \mathbb{N}$ different classes, let $p = (p_1, \dots, p_m)$ denote the vector of relative frequencies of class labels of some subset of observations. Then, $i(p)$ can be interpreted as a measure of heterogeneity of the class labels for the subset. Consequently, in order to evaluate a split, one uses the weighted sum of the impurities of the resulting two subset. For regression trees typical criteria to measure the heterogeneity of a set of n observations (X, y) are:

- Mean squared error: $\frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2$
- Mean absolute error: $\frac{1}{n} \sum_{i=1}^n |y_i - \tilde{y}_i|$

where $\tilde{y}_i$ denotes the predicted value for observation x_i which usually corresponds to the mean or median of the vector y . A splitting criterion is again derived by taking the weighted sum of these measures for the two subsets resulting from a split. Various other splitting criteria, such as the twoing rule for classification (Breiman et al. 1984), have been proposed in the literature but we refrain from going into further detail as the presented heuristic is applicable to all of these criteria. Throughout the remainder of this work, the splitting criterion will be denoted by the function $q : \mathbb{R}^{d+1} \times \mathbb{R}^{n \times d} \times \text{dom}(Y)^n \rightarrow \mathbb{R}$ such that $q(a, X, y)$, expresses the value of the splitting criterion of an oblique split defined by $a \in \mathbb{R}^{d+1}$ on a training set (X, y) . Without loss of generality, we assume that q should be minimized.

The focus of this work is the method of determining high quality oblique splits with respect to the employed splitting criterion and in the remainder of this work we describe our adaption of the CE method based on the von Mises–Fisher distribution.

3 Related work and contribution

Oblique decision tree induction has been an interesting topic of research over the past decades. The major problem related to this topic is the complexity of inducing these trees. Heath (1993) shows that even the task of finding a single optimal oblique split is NP-complete for some splitting criteria including classification error. One approach to overcome this problem is to build oblique decision trees in a top-down manner by recursively introducing heuristically obtained oblique splits. Various heuristics have been applied for finding near-optimal splits in this context. The first one, called CART-LC, is introduced by Breiman et al. (1984) who suggest a deterministic hill-climbing approach that sequentially updates the coefficients of the split until a local optimum is reached. To escape local optima, Heath et al. (1993) propose a simulated annealing heuristic which perturbs the hyperplane parameters one at a time. In their algorithm called OC1, Murthy et al. (1993, 1994) improve Breiman's hill climbing approach by introducing randomization techniques also with the goal of avoiding premature convergence. Other strategies for finding oblique splits are based on meta-heuristics such as simulated annealing, genetic algorithms or evolutionary algorithms (Cantú-Paz and Kamath 2003) or algorithms based on logistic regression (Mola and Siciliano 2002; Truong 2009), linear discriminants (Loh and Shih 1997; Li et al. 2003; Siciliano et al. 2008; López-Chau et al. 2013) or Householder transformations (Wickramarachchi et al. 2016). Recently, also mathematical optimization approaches have been proposed for inducing oblique decision trees that neglect the recursive partitioning scheme.

Bertsimas and Dunn (2017) propose an integer linear programming formulation for optimizing decision trees of a predetermined depth. Blanquero et al. (2020) propose a continuous optimization approach instead for optimizing randomized oblique decision trees. The major drawback of these approaches lies in the complexity of solving the proposed mathematical optimization programs. As a consequence, they can only be solved optimally in a reasonable amount of time for small to medium-sized datasets and for small depths.

The CE method was first introduced by Rubinstein for the estimation of rare event probabilities (Rubinstein 1997) and was later adapted for solving continuous and combinatorial optimization problems (Rubinstein 1999; Rubinstein and Kroese 2004; De Boer et al. 2005). CE is a model-based search method (Zlochin et al. 2004) that uses a parameterized probabilistic model for sampling feasible solutions that is iteratively updated until the model assigns a high probability mass to a region of near-optimal solutions. It has been successfully applied to a wide range of optimization problems and it has also been used in the context of machine learning by Mannor et al. (2005) who develop a CE method for building support vector machines to solve classification problems.

In this work, we follow the recursive top–down approach to induce oblique decision trees. We develop a CE method based on the von Mises–Fisher distribution that is well-suited for finding high-quality oblique splits for classification and regression tasks and therefore poses an interesting alternative to existing algorithms. It is inspired by the geometry of the underlying problem structure and uses easily comprehensible parameters for fine-tuning the execution of the algorithm. Our evaluation shows that it is suitable for inducing highly accurate and compact oblique decision trees in a small amount of time which makes it an interesting option for real-life applications in the field of machine learning and data analytics.

4 Preliminaries

4.1 The cross-entropy optimization method

In this section, we briefly summarize the CE method applied in this work. For a more comprehensive description of the method and its applications the interested reader is referred to Rubinstein and Kroese (2004), De Boer et al. (2005).

We consider a generic minimization problem of the form

$$\min_{a \in \mathcal{A}} q(a) \quad (1)$$

with an arbitrary objective function $q : \mathcal{A} \rightarrow \mathbb{R}$ defined on the set of feasible solutions $\mathcal{A}$.

Let $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ denote a family of probability density functions (pdf) on the domain $\mathcal{A}$ parameterized by an element θ in the parameter space Θ . The CE method iteratively generates a sequence of parameters $\theta_1, \dots, \theta_T$ and a sequence of levels $\gamma_1, \dots, \gamma_T$ for $T \in \mathbb{N}$ as follows: The first parameter θ_1 is given as an input.

In each iteration $t = 1, \dots, T$ a random sample $x_1, \dots, x_N \in \mathcal{A}$ of size $N \in \mathbb{N}$ is derived from f_{θ_t} . Then, the samples are ordered based on their performance, i.e. $q(x_{(1)}) \leq \dots \leq q(x_{(N)})$, and γ_t is calculated as the ρ -quantile $\gamma_t = q(x_{(\lceil \rho N \rceil)})$ for $0 < \rho < 1$. Subsequently, θ_{t+1} is derived by calculating the maximum likelihood parameter θ_E for the elite samples $E := \{x_i : i = 1, \dots, N; q(x_i) \leq \gamma_t\}$ and setting $\theta_{t+1} = \theta_E$. Alternatively, some smoothed update formula can be used that appropriately combines θ_E and θ_t to obtain θ_{t+1} . Intuitively, this update of the model parameter based on the best performing samples increases the new pdf's likelihood of generating good solutions and therefore, the final pdf f_{θ_T} concentrates its probability mass to a region of near-optimal solutions. Ideally, one hopes that the sequence of pdfs $(f_{\theta_t})_{t \in \mathbb{N}}$ converges to the point mass of an optimal solution a^* such that the sequence of levels $(\gamma_t)_{t \in \mathbb{N}}$ converges to the optimal objective value γ^* .

4.2 The von Mises–Fisher distribution

The CE method relies on a family of pdfs on the solution space that has to be specified in advance. In this work, we make use of the von Mises–Fisher distribution (Fisher 1953). This distribution is often referred to as the equivalent of the Gaussian distribution on the $(d-1)$ -sphere $S_{d-1} = \{a \in \mathbb{R}^d : \|a\|_2 = 1\}$. The pdf of the d -dimensional von Mises–Fisher distribution is given by

$$f_{\theta}(x) = C_d(\|\theta\|_2) \exp(\theta^T x)$$

for $\theta \in \mathbb{R}^d$. The parameter $\kappa := \|\theta\|_2$ is referred to as the *concentration parameter* and $\mu := \frac{\theta}{\|\theta\|_2}$ denotes the *mean direction*. Intuitively, for $\kappa = 0$ one obtains a uniform distribution on S_{d-1} and a higher value of κ indicates a higher concentration of the distribution around the mean direction μ . The normalization constant $C_d(\kappa)$ is defined by

$$C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}$$

where I_{α} denotes the modified Bessel function of the first kind of order α . Figure 2 illustrates the von Mises–Fisher distribution for different concentration parameters κ and a constant mean direction μ .

Maximum likelihood estimation To update the parameters of the pdfs in the CE method, the maximum likelihood estimates (MLE) need to be determined. As mentioned in Banerjee et al. (2005), Mardia and Jupp (1999), Dhillon and Sra (2003), given a sample $a_1, \dots, a_N \in \mathbb{R}^d$ the MLE of μ and κ for the von Mises–Fisher distribution are given by

$$\mu = \frac{\sum_{i=1}^N a_i}{\|\sum_{i=1}^N a_i\|_2}, \quad \kappa = A_d^{-1}(\bar{R}) \quad (2)$$

Fig. 2 Illustration of the von Mises–Fisher distribution for different concentration parameters and a constant mean direction $\mu = (\frac{1}{3}, \frac{2}{3}, \frac{2}{3})$

where

$$A_d(\kappa) = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} = \frac{\sum_{i=1}^N a_i}{N} = \bar{R}. \quad (3)$$

Since there doesn't exist an analytical solution for the inverse of the ratio of modified Bessel functions, κ can only be approximated. Nevertheless, sufficiently accurate approximations are available for our purposes:

- Banerjee et al. (2005) give a simple approximation of κ :

$$\hat{\kappa}_0 = \frac{\bar{R}(d - \bar{R}^2)}{1 - \bar{R}^2} \quad (4)$$

- Sra (2012) develops a more exact approximation by performing a few iterations of the Newton method starting from $\hat{\kappa}_0$, as derived in Eq. 4:

$$\begin{aligned}\hat{\kappa}_1 &= \hat{\kappa}_0 - \frac{A_d(\hat{\kappa}_0) - \bar{R}}{1 - A_d(\hat{\kappa}_0)^2 - \frac{d-1}{\hat{\kappa}_0} A_d(\hat{\kappa}_0)} \\ \hat{\kappa}_2 &= \hat{\kappa}_1 - \frac{A_d(\hat{\kappa}_1) - \bar{R}}{1 - A_d(\hat{\kappa}_1)^2 - \frac{d-1}{\hat{\kappa}_1} A_d(\hat{\kappa}_1)}\end{aligned}\quad (5)$$

This approximation is more accurate than the one in Eq. 4, yet the evaluation of $A_d(\hat{\kappa}_0)$ is expensive in high dimensions and also introduces the risk of floating point over- and underflows.

Sampling from the von Mises–Fisher distribution For our CE algorithm, in order to ensure low running times, we need an efficient algorithm for sampling from the von Mises–Fisher distribution. The description of such an algorithm is beyond the scope of this paper, yet, the interested reader is referred to Wood’s algorithm (Wood 1994) which is based on Ulrich’s results (Ulrich 1984).

5 The cross-entropy method for optimization of oblique splits

5.1 Geometric motivation

In this work, the problem under consideration is finding the coefficients $a \in \mathbb{R}^{d+1}$ of an oblique split on a training set (X, y) that minimizes a certain splitting criterion q . Note that the quality of an oblique split does not depend on the length $\|a\|_2$ of the vector $a \in \mathbb{R}^{d+1}$ but solely on its direction $a/\|a\|_2$. Thus, as $a = 0$ is no meaningful solution, we can, more adequately, identify the feasible region by the d -sphere $S_d = \{a \in \mathbb{R}^{d+1} : \|a\|_2 = 1\}$ instead of $\mathbb{R}^{d+1}$. Hence, the problem can alternatively be expressed as:

$$\min_{a \in S_d} q(a, X, y) \quad (6)$$

Note that each observation x_i for $i = 1, \dots, n$ from the training set X defines a hyperplane $H(x_i) := \{a \in \mathbb{R}^{d+1} : a_1 x_{i1} + \dots + a_d x_{id} + a_{d+1} = 0\}$ which divides S_d into two half spheres

$$S_d^+(x_i) := \left\{ a \in S_d : \sum_{j=1}^d a_j x_{ij} + a_{d+1} \geq 0 \right\}$$

and

$$S_d^-(x_i) := \left\{ a \in S_d : \sum_{j=1}^d a_j x_{ij} + a_{d+1} < 0 \right\}.$$

Fig. 3 Geometric interpretation of the solution space for a dataset with 30 observations randomly chosen from $[-1, 1] \times [-1, 1]$. The solution space can be represented by S_2 and the dataset divides it into multiple connected regions

Observation x_i satisfies the rule $\sum_{j=1}^d a_j X_j + a_{d+1} \geq 0$ if and only if $a \in S_d^+(x_i)$. Therefore, the nonempty sets in $\mathcal{R} := \{R = \bigcap_{i=1}^n S_d^{\sigma_i}(x_i) : \sigma \in \{+, -\}^n \wedge R \neq \emptyset\}$ correspond to connected regions of solutions on the sphere S_d with the same value for the splitting criterion. Hence, the problem of finding an optimal oblique split is equivalent to finding an optimal region $R \in \mathcal{R}$. It follows from Cover's findings (Cover 1965) that there are up to $2 \sum_{j=0}^d \binom{n-1}{j}$ regions in $\mathcal{R}$ which also explains the complexity of finding an optimal oblique split and the necessity of relying on heuristics. Figure 3 shows the resulting regions on S_2 for a dataset with 30 observations chosen uniformly at random from $[-1, 1] \times [-1, 1]$.

The previous observation motivates a heuristic that randomly samples points from S_d and iteratively updates the sampling distribution by favoring the best points until the sampling density assigns most of its probability mass to a near-optimal region on the d -sphere. The CE optimization framework is suitable for achieving that goal and the von Mises–Fisher distribution provides an efficient parametric sampling scheme and easy to compute accurate approximations of MLEs which are necessary for the CE method.

5.2 Algorithmic description

A summary of our proposed CE method is presented in Algorithm 1. The algorithm gets a training set (X, y) and the splitting criterion as an input. Apart from that, the sample size $N \in \mathbb{N}$, the quantile parameter $0 < \rho < 1$, the smoothing parameter $0 \leq \alpha < 1$ and a termination parameter $K \in \mathbb{N}_0$ have to be specified.

First, our algorithm normalizes the dataset X . The resulting near-optimal solution a^* is denormalized at the end of the algorithm such that it equivalently divides the dataset as the solution found for the normalized dataset. The reason for normalizing is discussed in Sect. 5.4.

Initially, we set the iteration counter $t = 1$. The parameter κ_1 is initialized to zero and μ_1 is set to the $(d + 1)$ -dimensional vector of zeros. This way, the first sample is drawn from the uniform distribution on the d -sphere S_d . The incumbent solution a^* is also initialized uniformly at random. The variable γ^* , which is used to track the minimum value of the level parameters γ_t , is initially set to infinity. To count the

Algorithm 1: Cross-entropy method for oblique splits

Data: Training set (X, y) ; splitting criterion q ; sample size $N \in \mathbb{N}$; quantile parameter $0 < \rho < 1$; smoothing parameter $0 \leq \alpha < 1$; termination parameter K

Result: Near-optimal Solution a^* ;

```

1  $X' \leftarrow \text{Normalize}(X)$ ;
2  $t \leftarrow 1$ ;
3  $k \leftarrow 0$ ;
4  $(\kappa_1, \mu_1) \leftarrow (0, \mathbf{0})$ ;
5  $\theta_1 \leftarrow \kappa_1 \mu_1$ ;
6  $\gamma^* \leftarrow +\infty$ ;
7  $a^* \leftarrow$  Random sample from the pdf  $f_{\theta_1}$ ;
8 while  $k < K$  do
9   Draw  $N$  iid solutions  $a_1, \dots, a_N \in \mathbb{R}^{d+1}$  from the pdf  $f_{\theta_t}$  using Wood's algorithm;
10  Sort the solutions in increasing order  $a_{(1)}, \dots, a_{(N)}$  according to their performances such that
    $q(a_{(1)}, X, y) \leq \dots \leq q(a_{(N)}, X, y)$ ;
11   $\gamma_t \leftarrow q(a_{(\lceil \rho N \rceil)}, X, y)$ ;
12  if  $q(a_{(1)}, X, y) \leq q(a^*, X, y)$  then
13     $a^* \leftarrow a_{(1)}$ ;
14  end
15  if  $\gamma_t < \gamma^*$  then
16     $\gamma^* \leftarrow \gamma_t$ ;
17     $k \leftarrow 0$ ;
18  end
19  else
20     $k \leftarrow k + 1$ ;
21  end
22   $E \leftarrow \{a_i : q(a_i, X, y) \leq \gamma_t; i = 1, \dots, N\}$ ;
23   $\mu_E, \kappa_E \leftarrow \text{MLE}(E)$ ; ▷ See Sect. 4.2
24   $\mu_{t+1} \leftarrow \frac{\alpha \mu_t + (1-\alpha) \mu_E}{\|\alpha \mu_t + (1-\alpha) \mu_E\|_2}$ ;
25   $\kappa_{t+1} \leftarrow \alpha \kappa_t + (1-\alpha) \kappa_E$ ;
26   $\theta_{t+1} \leftarrow \kappa_{t+1} \mu_{t+1}$ ;
27   $t \leftarrow t + 1$ ;
28 end
29 return  $\text{Denormalize}(a^*)$ 

```

number of consecutive iterations without improvement of γ^* , the auxiliary variable k is introduced and initialized to zero.

In the t -th iteration, the algorithm draws N samples from the von Mises–Fisher distribution with density f_{θ_t} for $\theta_t = \kappa_t \mu_t$ using Wood's algorithm. The samples are sorted in increasing order based on their performance and the level γ_t is computed as the ρ -quantile of the sample performances. If the best sample is better than the incumbent solution a^* , the incumbent solution is updated accordingly. If $\gamma_t < \gamma^*$, the variable γ^* is updated and k is reset to zero. Otherwise, the counter k is increased by one. Subsequently, the set of elite samples E with an objective value smaller or equal to γ_t is determined and the maximum likelihood estimates μ_E and κ_E for the elite samples are approximated as described in Sect. 4.2. The new parameters κ_{t+1} and μ_{t+1} are then derived using the smooth update formulas

$$\mu_{t+1} = \frac{\alpha \mu_t + (1 - \alpha) \mu_E}{\|\alpha \mu_t + (1 - \alpha) \mu_E\|_2}, \quad \kappa_{t+1} = \alpha \kappa_t + (1 - \alpha) \kappa_E$$

which combine the current mean direction and concentration with the maximum likelihood estimates for the elite samples. Note that the mean direction of the von Mises–Fisher distribution is a unit vector which explains the scaling in the update formula for μ_{t+1} . Without scaling, the concentration would be affected. If the parameter α is set to 0, no smoothing is carried out and the new parameters for μ_{t+1} and κ_{t+1} are set directly to the maximum likelihood estimates of the elite samples.

The algorithm stops when the minimum observed level γ^* has not been updated for K consecutive iterations. This indicates that the algorithm has converged to a region of similar solutions and no further improvement of the incumbent solution can be expected.

5.3 Illustration

In this section, we briefly illustrate the execution of our proposed algorithm for the Iris dataset. For illustration purposes, the dataset is restricted to the attributes sepal length and sepal width and it is normalized using the robust scaling method described in the next section. As objective function the weighted gini impurity is used and the parameters are set to $\rho = 0.1$, $\alpha = 0.8$ and $K = 1$.

Figure 4 summarizes the progress of our method. As we can see, in the first iteration the samples are taken uniformly at random from S_2 . In the following iterations the mean direction converges and the concentration increases such that in the final iteration the pdf θ_6 concentrates its probability mass tightly around a region of optimal solutions. It can be shown that the optimal solution has an objective value of $1/3$ which corresponds to the value taken on by level parameter γ_5 . The algorithm terminates after iteration $t = 6$ as the level parameter γ^* could not be improved.

Figure 5 shows the obtained split for both, the normalized and the original dataset and we can see, that the split perfectly separates the samples of class setosa from the rest of the dataset.

5.4 Normalization

Empirically, we found that our algorithm can vastly benefit from data normalization techniques. We observed that normalization improves the quality of the splits as well as the runtime of our algorithm. The reason for this is that without normalization, the regions in $\mathcal{R}$ are not spread evenly across the sphere. To illustrate this, we again consider the Iris dataset restricted to the attributes sepal length and sepal width. Figure 6 shows how the S_2 sphere is divided into regions by the samples in the dataset before and after normalization. For this, example, we used robust normalization, i.e. we set $x'_{ij} = \frac{x_{ij} - \text{Med}(X_j)}{\text{IQR}(X_j)}$ where $\text{Med}(X_j)$ and $\text{IQR}(X_j)$ correspond to the median and the interquartile range of feature X_j , respectively.

As we can see, without normalization most of the regions in $\mathcal{R}$ are concentrated very tightly within a small portion of the unit sphere while after normalization, the regions are spread out more heterogeneously. Thus, normalization enables our algorithm to find meaningful solutions earlier in the optimization process which improves the overall runtime.

Fig. 4 Progress of the cross-entropy method for the Iris dataset restricted to the first two attributes. The samples drawn in each iteration are represented by a cross and the elite samples by a circle. The arrow represents the mean direction μ_t used for sampling in iteration $t = 1, \dots, 6$

Additionally, normalization can help to reduce numerical instabilities in the sampling process and the approximation of the maximum likelihood estimates as the region of equivalent solutions become larger and thus, smaller values of κ are required in the later stages of our algorithm.

Other normalization variants which might lead to better results in certain situations include:

Fig. 5 The resulting oblique split for the normalized and original Iris dataset restricted to the first two attributes of the Iris dataset

Fig. 6 Illustration of solution space for the first two attributes of the Iris dataset before and after normalization

- Min–Max–Scaling: $x'_{ij} = \frac{x_{ij} - \text{Mean}(X_j)}{\text{Max}(X_j) - \text{Min}(X_j)}$
- Standard–Scaling: $x'_{ij} = \frac{x_{ij} - \text{Mean}(X_j)}{\sigma(X_j)}$

Here $\text{Mean}(X_j)$, $\text{Max}(X_j)$, $\text{Min}(X_j)$ and $\sigma(X_j)$ denote the mean, maximum, minimum and standard deviation of feature X_j , respectively.

At the end of our algorithm, we denormalize the obtained oblique split such that it divides the dataset in the same manner as the split for the normalized data. If we use any normalization method of the form $x'_{ij} = \frac{x_{ij} - \delta_j}{\Delta_j}$, this can be achieved by setting

$$\bar{a}_j := \begin{cases} \frac{a_j^*}{\Delta_j}, & \text{if } j < d + 1 \\ a_{d+1}^* - \sum_{j=1}^d \frac{a_j^* \delta_j}{\Delta_j}, & \text{if } j = d + 1 \end{cases}$$

for $j = 1, \dots, d + 1$. Then, $\bar{a}$ has the same value of the splitting criterion on (X, y) as a^* on the normalized dataset (X', y) .

5.5 Feature reduction

It is sometimes desirable to exclude some of the attributes in a split in order to decrease the computational effort of the CE method and to increase interpretability of the splits. As an example, consider the situation in which a node relatively deep within the tree should be split. It is quite common that in such a situation the number of attributes is higher than the number of observations associated with the node. As a result, there exist optimal oblique splits which don't require all of the attributes or equivalently satisfy $a_j = 0$ for some $j = 1, \dots, d$. It is, however, not clear beforehand which of the attributes can be left out safely without eliminating the optimal oblique splits.

Yet, there is a simple way to eliminate features due to linear dependencies that, for every oblique split involving all of the attributes, guarantees the existence of an equivalent split which uses less attributes than there are observations in the dataset. The key observation for this idea is summarized by the following lemma.

Lemma 1 *Let (X, y) denote a training set consisting of n observations and d features and let $X_1, \dots, X_d$ denote the columns of the matrix X . Further, let $\mathbf{1}$ denote the n -dimensional all-ones vector. Let $J \subseteq \{1, \dots, d\}$ denote a maximal subset of indices such that the set of vectors $\{X_j : j \in J\} \cup \{\mathbf{1}\}$ is linear independent. Then, for every $a \in \mathbb{R}^{d+1}$ there exists $\bar{a} \in \mathbb{R}^{d+1}$ with $\bar{a}_j = 0$ for every $j \notin J$ that satisfies*

$$\sum_{j=1}^d a_j X_j + a_{d+1} \mathbf{1} = \sum_{j=1}^d \bar{a}_j X_j + \bar{a}_{d+1} \mathbf{1}.$$

Proof As the set $\{X_j : j \in J\} \cup \{\mathbf{1}\}$ is linear independent and J is maximal, each $X_{j'}$ for $j' \notin J$ can be expressed as a linear combination of these vectors. It follows that there exist $\lambda_j \in \mathbb{R}$ for $j \in J$ and $\lambda_{d+1} \in \mathbb{R}$ such that

$$\sum_{j \in J} \lambda_j X_j + \lambda_{d+1} \mathbf{1} = \sum_{j' \notin J} a_{j'} X_{j'}.$$

Consequently, it holds that

$$\sum_{j=1}^d a_j X_j + a_{d+1} \mathbf{1} = \sum_{j \in J} (a_j + \lambda_j) X_j + (a_{d+1} + \lambda_{d+1}) \mathbf{1}$$

Defining $\bar{a} \in \mathbb{R}^{d+1}$ by

$$\bar{a}_j := \begin{cases} (a_j + \lambda_j), & \text{if } j \in J \cup \{d+1\} \\ 0, & \text{else} \end{cases}$$

concludes the proof. $\square$

Clearly, as $\text{rank}((\mathbf{1} \ X)) \leq \min(n, d+1)$ the previous lemma implies the existence of an optimal oblique split involving less attributes than observations if $n \leq d$.

A subset J of attributes as required by the lemma can easily be obtained by performing Gaussian elimination on the transposed matrix $(\mathbf{1} \ X)^T$.

One should note that the choice of attributes is not necessarily unique and can influence the heuristic search process of the proposed CE method and thus, also affect the objective value of the final solution. Intuitively, to simplify the heuristic search one would rather eliminate attributes that require additional information than ones that are expressive on their own. In order to accommodate for this, we propose the following strategy: One first computes the best univariate split for each attribute and then ranks the attributes by their performance. In each iteration of the Gaussian elimination procedure, we then choose the next pivot element based on this ranking if there are multiple candidates.

5.6 Parallelization

Our proposed CE method can vastly benefit from parallelization. In order to determine the value of the splitting criterion for a sample $a \in S_d$, n scalar products have to be evaluated, one for each sample in the training set. Consequently, the algorithm spends most of its execution time computing those. Clearly, this can easily be parallelized by dividing the samples evenly into multiple batches and assigning each batch to a different thread which computes the respective scalar products. This way, we can leverage the capabilities of modern multicore processors to speed up the execution time of our algorithm.

6 Evaluation

In this section, a comprehensive evaluation of our proposed method, which in the remainder of this section will be denoted by CE-DT, is presented. Our algorithm was implemented in C++ and all experiments were executed on a computer equipped with an Intel Xeon E3-1231v3 @3.40 GHz (4 Cores) and 32 GB DDR3 RAM running Ubuntu 20.04.

We use 20 classification datasets from the UCI Machine Learning repository (Dheeru and Karra Taniskidou 2017) to conduct our experiments. An overview of the employed datasets is presented in Table 1. One thing to note here is that the number of attributes includes the dummy variables resulting from one-hot encoding of categorical attributes, as explained below. We evaluate three different quality criteria which are out-of-sample accuracy, tree size in terms of the number of leaf nodes and the time necessary for building the trees.

CE-DT is compared to the OC1 oblique decision tree induction algorithm and to our own implementation of the greedy top-down algorithm for inducing univariate decision trees (DT). The latter basically corresponds to the well-known CART algorithm but by using our own implementation, we ensure maximum comparability to CE-DT as only the method for deriving the splits is replaced. We also include results of the decision trees when an additional post-pruning step is executed. For this, we choose minimal cost-complexity pruning, which was introduced by Breiman et al. (1984).

Table 1 Dataset description

Dataset	Data points	Attributes	Classes
Balance scale	625	56	4
Breast cancer wisconsin	699	9	2
Chess (king rook vs. king pawn)	3196	73	2
Congressional voting records	435	16	2
Dermatology	368	34	6
Glass	226	9	6
Ionosphere	351	34	2
Monks problems 1	124	6	2
Optical recognition	3895	63	16
Parkinsons	195	22	2
Pen based recognition	7494	16	10
Pima Indian diabetes	768	8	2
Seeds	210	7	3
Semeion	1593	256	10
Spambase	4601	57	2
Statlog heart	270	13	2
Statlog landsat	4435	36	6
Statlog German credit (numeric)	1000	20	2
Tic tac toe	958	27	2
Wine	178	13	3

Additionally, for out-of-sample accuracy we compare CE-DT to two other popular classification models, namely neural networks (NN) and support vector machines (SVM). Moreover, we include results for the two tree-based ensemble methods random forest (RF) and Gradient Boosting (GB). For these comparison classifiers we use the implementations provided in Python's Scikit-learn library (Pedregosa et al. 2011) at default parameters.

For each dataset 10-fold cross-validation is performed and the average performance and the standard deviation, each rounded to two decimals, is reported. For the results of the pruned decision trees 10% of the training data is held-out for pruning. Missing numerical values are imputed beforehand with the median of the values in the training set for the respective attribute. For categorical attributes the most frequent category is used instead. Subsequently, one-hot encoding is employed to deal with categorical attributes. This preprocessing guarantees that in each run of the 10-fold cross-validation scheme all of the classifiers are trained and tested on the same subsets of observations.

We use the gini impurity as a splitting criterion and for all three decision tree induction algorithms the recursive partitioning is stopped when a leaf node holds less than four samples. This is also the standard setting used by OC1. For the CE method we choose the parameters $\alpha = 0.8$, $\rho = 0.1$, $K = 3$ and $N = \max(100, 2d \log_2(n_l))$ where d denotes the number of attributes and n_l denotes the number of training

observations associated with the leaf node l to be split. This choice ensures that the number of samples drawn in each iteration of the CE method for the root node grows logarithmically with the total number of training observations in the dataset. Moreover the number of samples decreases with every split in the tree but it never falls below 100. This guarantees that at least 10 elite samples for updating the mean direction and the concentration are used even when the number of training observations associated with the node to be split is very small. Note that these parameters have not been optimized specifically for each individual dataset but they generally resulted in a good performance in our preliminary experiments. In practice the given parameters can serve as an orientation but should be tuned further, for example by increasing the number of samples, to improve the performance of the algorithm.

For the evaluations, we follow Demšar's (2006) guidelines to compare multiple classifiers and perform a Friedman test (Friedman 1937, 1940) to check whether the performances of at least two classifiers differ significantly. This non-parametric statistical test ranks the performances of the algorithms for each dataset from 1 to k where k is the number of algorithms to be compared. The lowest rank is assigned to the best and the highest rank to the worst performing algorithm. In case of ties the average rank is assigned. The null-hypothesis is that the classifiers perform equally well and in this case the observed ranks should be approximately equal. This hypothesis is rejected if the p -value computed by the Friedman test is below a certain significance level. If a significant difference for at least two classifiers is observed, a subsequent Holm test (Holm 1979) with CE-DT as control method is carried out to check which of the other methods perform significantly better or worse than CE-DT. For each other method the null-hypotheses is therefore that it performs equally well as CE-DT. The p -values for each of those are calculated and ordered from lowest to highest. If k denotes the number of hypotheses and i the position in the ordering, the Holm test adjusts these p -values by setting $p'_1 = \min\{1, kp_1\}$ and $p'_i = \min\{1, \max\{p'_{i-1}, (k+1-i)p_i\}\}$ for $i = 2, \dots, k$. Smaller p -values are therefore increased more strongly than higher ones which ensures that the family-wise error rate is below the required significance level. Then, in order of the original p -values, the adjusted p -values are compared to the significance level until it is exceeded for the first time. All hypotheses before that are rejected and we can conclude that there is a significant difference to the performance of the control method. For our evaluation we set the significance level for both statistical tests to 0.05.

6.1 Comparison of accuracy

In this section, we evaluate the accuracy of our proposed method which is defined as the ratio of the number of well-classified samples to the overall number of samples in the testing data.

Table 2 summarizes the average accuracies over the ten runs of the cross-validation process of the unpruned decision trees using the decision tree induction algorithms CE-DT, DT and OC1 for each of the employed datasets. With a value of 1.5 CE-DT has the lowest average rank of the three methods and OC1 and DT share the second place with an average rank of 2.25. With a p -value of 0.02 the Friedman test

Table 2 Comparison of accuracy between unpruned CE-DT, DT and OC1

Dataset	Accuracy (%)		
	CE-DT	DT	OC1
balance scale	90.71±4.17	76.8±4.01	91.67±2.39
breast cancer wisconsin	94.99±2.5	92.97±4.82	92.98±3.78
chess	98.69±0.86	99.53±0.38	95.37±1.46
congressional voting records	94.25±2.96	94.5±4.47	93.8±3.07
dermatology	96.2±1.79	94.84±2.26	95.07±4.28
glass	73.33±9.85	72.18±9.23	71.27±10.76
ionosphere	89.46±4.43	90.32±5.13	88.61±2.82
monks problems 1	91.35±7.24	83.65±12.64	91.16±8.5
optical recognition	94.86±1.12	95.02±1.21	95.1±1.21
parkinsons	88.71±6.86	87.76±7.17	84.63±4.49
pen based recognition	97.62±0.29	96.2±0.92	96.38±0.56
pima indian diabetes	68.75±5.6	70.31±5.03	70.19±3.21
seeds	93.33±4.36	87.62±7.44	91.43±4.67
semeion	86.75±2.81	77.21±2.43	72.13±3.04
spambase	92.91±1.14	91.85±1.73	92.48±1.24
statlog heart	76.67±9.38	71.48±7.42	78.15±5.61
statlog landsat	86.56±1.6	85.55±1.93	85.52±2.41
statlog german credit	95.0±3.26	95.3±2.15	94.0±1.26
tic tac toe	95.82±2.93	94.15±2.2	94.68±2.67
wine	96.63±3.71	88.76±3.52	91.63±5.68
Average Rank	1.5	2.25	2.25

The best results are highlighted in bold

Table 3 Holm post-hoc test for accuracy of the unpruned trees

Algorithm	Rank	Holm adjusted p -value	Hypothesis
DT	2.25	3.5×10^{-2}	Rejected
OC1	2.25	3.5×10^{-2}	Rejected

Control method: CE-DT, Rank: 1.5

reports a significant difference for at least two methods and the post-hoc Holm test, as summarized in Table 3, rejects both null-hypotheses. This allows the conclusion that CE-DT induced significantly more accurate decision trees in this evaluation.

The results for the pruned decision trees, as summarized in Table 4, are similar. CE-DT again has the lowest average rank of 1.45, OC1 has a rank of 2.25 and DT's rank is 2.3. The Friedman test returns a value of 0.01 and the null-hypotheses is again rejected. As for the unpruned trees, the Holm test, summarized in Table 5, confirms that CE-DT significantly outperforms the other two methods.

Table 6 summarizes the results of pruned CE-DT, the non-tree-based classifiers NN and SVM and the tree-based ensemble methods GB and RF. RF has the lowest average rank of 2.3, followed by GB with a rank of 2.375. With a rank of 3.225 CE-DT is in

Table 4 Comparison of accuracy between pruned CE-DT, DT and OC1

Dataset	Accuracy (%)		
	CE-DT	DT	OC1
Balance scale	90.23±4.36	78.39±5.26	90.39±5.09
Breast cancer wisconsin	96.42±2.08	93.41±3.78	94.98±4.09
Chess	98.53±0.79	99.31±0.56	95.4±1.96
Congressional voting records	96.09±3.26	95.63±2.84	94.01±2.98
Dermatology	95.4±3.77	95.13±3.58	91.51±7.12
Glass	73.1±8.78	68.31±13.13	71.41±5.04
Ionosphere	88.9±3.68	88.6±3.62	87.48±7.08
Monks problems 1	85.51±9.54	81.92±10.56	79.62±15.06
Optical recognition	94.91±1.26	95.04±0.98	94.78±1.43
Parkinsons	87.05±11.16	83.0±11.17	83.03±6.28
Pen based recognition	96.82±0.51	95.92±0.89	96.4±0.72
Pima Indian diabetes	74.08±4.61	72.27±4.23	72.63±5.92
Seeds	92.38±5.3	89.52±4.67	93.81±6.04
Semeion	86.88±2.35	75.01±3.36	68.48±2.78
Spambase	93.28±0.8	92.18±1.25	93.46±1.15
Statlog heart	79.26±6.46	77.04±6.15	75.18±6.84
Statlog landsat	87.64±1.95	85.93±1.35	86.04±1.15
Statlog german credit	95.4±2.62	96.4±1.5	95.5±1.63
Tic tac toe	92.37±3.83	92.91±3.86	92.38±3.9
Wine	95.46±4.27	87.65±4.13	91.63±4.46
Average rank	1.45	2.3	2.25

The best results are highlighted in bold

Table 5 Holm post-hoc test for accuracy of the pruned trees

Algorithm	Rank	Holm adjusted p -value	Hypothesis
DT	2.3	1.4×10^{-2}	Rejected
OC1	2.25	1.4×10^{-2}	Rejected

Control method: CE-DT, Rank: 1.45

third place before SVM and NN with ranks 3.4 and 3.7, respectively. Though with a p -value of 1.3×10^{-2} the Friedman test reports a significant difference between at least two classifiers, the post-hoc Holm test, which is summarized in Table 7, does not provide sufficient evidence for a significant difference between CE-DT and any of the other methods.

6.2 Comparison of tree size

Next, we compare the sizes of the decision trees induced by CE-DT, DT and OC1. The tree size is measured in terms of the number of leaf nodes which corresponds to the number of distinct regions the feature space is divided into by the decision tree.

Table 6 Comparison of accuracy between pruned CE-DT, NN, SVM, GB and RF

Dataset	Accuracy (%)				
	CE-DT	NN	SVM	GB	RF
Balance scale	90.23±4.36	93.12 ±2.88	89.92±3.64	87.84±3.68	81.76±3.83
Breast cancer wisconsin	96.42±2.08	95.85±2.52	96.7 ±2.14	95.7±3.08	96.27±2.9
Chess	98.53 ±0.79	97.68±0.69	97.25±0.81	97.81±0.85	98.5±0.39
Congressional voting records	96.09±3.26	95.64±3.87	95.64±3.14	95.86±2.26	96.1 ±2.51
Dermatology	95.4±3.77	97.0±2.6	69.29±4.86	96.73±1.64	97.03 ±1.87
Glass	73.1±8.78	43.52±10.1	33.72±7.49	78.34 ±8.0	74.3±6.19
Ionosphere	88.9±3.68	91.75±4.81	94.29 ±2.22	92.3±2.25	93.15±2.92
Monks problems 1	85.51±9.54	75.96±13.59	80.38±12.11	92.56 ±9.46	80.45±10.78
Optical recognition	94.91±1.26	95.35±0.84	93.91±1.12	95.33±0.74	95.56 ±0.95
Parkinsons	87.05±11.16	81.05±10.07	81.58±8.34	92.82 ±6.21	89.26±6.11
Pen based recognition	96.82±0.51	98.85±0.36	99.51 ±0.23	98.85±0.23	98.4±0.51
Pima Indian diabetes	74.08±4.61	66.79±3.98	75.66 ±2.99	74.88±5.75	75.52±5.14
Seeds	92.38±5.3	76.19±23.13	90.0±6.19	93.33 ±4.86	92.86±3.84
Semeion	86.88±2.35	91.84±1.54	95.23 ±1.67	93.03±1.39	86.57±2.7
Spambase	93.28±0.8	92.35±2.38	70.83±1.07	94.44 ±0.78	94.35±0.94
Statlog heart	79.26±6.46	76.67±14.06	64.81±8.16	79.26±10.76	81.48 ±7.22
Statlog landsat	87.64±1.95	77.41±2.32	88.86±1.57	90.12 ±1.17	89.45±0.86
Statlog german credit	95.4±2.62	91.7±2.49	71.2±3.49	96.0±1.79	96.3 ±2.19
Tic tac toe	92.37±3.83	98.12±0.63	98.22 ±1.05	97.7±1.38	94.15±1.99
Wine	95.46±4.27	55.23±30.11	71.05±13.71	95.49±2.26	96.67 ±5.09
Average rank	3.225	3.7	3.4	2.375	2.3

The best results are highlighted in bold

Table 7 Holm post-hoc test for accuracy of pruned CE-DT, NN, SVM, RF and GB

Algorithm	Rank	Holm adjusted p -value	Hypothesis
NN	3.7	0.68	Not rejected
SVM	3.4	0.73	Not rejected
GB	2.375	0.27	Not rejected
RF	2.3	0.26	Not rejected

Control method: CE-DT, Rank: 3.225

Table 8 Comparison of tree size between unpruned CE-DT, DT and OC1

Dataset	Leaves		
	CE-DT	DT	OC1
Balance scale	23.0±3.07	94.9±3.18	20.4±2.06
Breast cancer wisconsin	10.7±1.95	29.2±3.03	15.7±1.62
Chess	8.7±0.78	48.6±1.5	50.8±7.15
Congressional voting records	6.7±0.64	25.5±2.97	10.3±1.79
Dermatology	7.1±1.04	16.8±0.6	8.0±0.45
Glass	24.8±1.6	37.2±1.6	27.1±1.14
Ionosphere	7.7±0.64	21.1±1.22	13.9±1.97
Monks problems 1	5.9±1.37	16.3±2.57	6.5±1.63
Optical recognition	49.9±2.84	104.2±3.54	57.7±1.27
Parkinsons	7.7±1.27	12.2±0.98	11.3±1.1
Pen based recognition	81.9±4.59	201.1±4.85	98.3±6.44
Pima Indian diabetes	65.2±2.6	105.7±5.18	68.4±2.15
Seeds	6.7±1.1	12.8±0.98	8.8±1.17
Semeion	19.8±1.54	186.8±3.43	58.6±5.62
Spambase	81.7±6.44	240.3±5.25	41.9±3.05
Statlog heart	17.5±1.69	36.1±2.84	15.9±1.51
Statlog landsat	97.4±6.71	299.0±8.12	237.6±4.63
Statlog german credit	13.3±1.49	29.4±2.15	14.2±1.78
Tic tac toe	9.5±3.93	65.5±5.52	9.9±1.87
Wine	3.3±0.64	10.4±1.74	7.4±0.92
Average rank	1.15	2.95	1.9

The best results are highlighted in bold

The results for the decision tree induction algorithms without pruning are presented in Table 8. For all but three dataset CE-DT induced the smallest decision trees and thus it has the lowest rank of 1.15. OC1 is in the second place with a rank of 1.9 and DT has a rank of 2.95 as it induced the largest trees for all but one dataset. With a p -value of 8×10^{-8} the Friedman test confirms a significant difference. The subsequent Holm test, which is summarized in Table 9, rejects both null-hypotheses from which we can conclude that CE-DT induced significantly smaller decision trees than DT and OC1 in our experiments.

Table 9 Holm post-hoc test for tree size of the unpruned trees

Algorithm	Rank	Holm adjusted p -value	Hypothesis
DT	2.95	2.5×10^{-8}	Rejected
OC1	1.9	1.8×10^{-2}	Rejected
Control method: CE-DT, Rank: 1.15			

Table 10 Comparison of tree size between pruned CE-DT, DT and OC1

Dataset	Leaves		
	CE-DT	DT	OC1
Balance scale	10.5±6.18	31.6±23.6	10.1±4.87
Breast cancer wisconsin	2.6±1.8	8.2±6.58	2.9±1.37
Chess	4.2±1.17	28.7±7.14	27.9±14.97
Congressional voting records	2.0±0.0	4.4±3.9	3.0±1.79
Dermatology	6.0±0.0	8.0±2.1	6.3±1.1
Glass	14.4±5.26	7.9±3.67	15.1±5.8
Ionosphere	3.3±1.1	4.2±1.72	4.5±2.25
Monks problems 1	4.7±2.15	7.2±5.0	4.6±1.69
Optical recognition	26.2±12.01	27.3±12.16	28.2±9.04
Parkinsons	3.8±2.27	5.2±3.66	5.6±3.35
Pen based recognition	50.8±13.98	155.1±25.75	78.2±16.98
Pima Indian diabetes	4.9±2.98	17.1±14.86	7.1±8.6
Seeds	3.1±0.3	4.4±1.62	3.2±0.4
Semeion	10.3±0.64	83.4±32.79	15.3±5.92
Spambase	22.3±11.47	73.1±39.45	7.0±4.27
Statlog heart	7.1±4.74	8.2±7.28	4.6±3.23
Statlog landsat	20.6±13.01	94.4±47.73	38.4±16.43
Statlog german credit	2.3±0.9	5.9±3.83	2.5±0.81
Tic tac toe	5.4±3.53	40.7±15.32	5.3±1.49
Wine	3.5±1.02	5.3±1.85	3.5±0.67
Average rank	1.325	2.75	1.925

The best results are highlighted in bold

For the pruned decision trees the results are summarized in Table 10. CE-DT again has the lowest rank of 1.325 followed by OC1 with a rank of 1.925 and DT ranks last with a value of 2.75.

The Friedman test reports a significant difference with a p -value of 5×10^{-5} but the Holm test, summarized in Table 11, only confirms that CE-DT performs significantly better than DT. It does not provide sufficient evidence that the null-hypothesis for OC1 can be rejected.

Table 11 Holm post-hoc test for tree size of the pruned trees

Algorithm	Rank	Holm adjusted p -value	Hypothesis
DT	2.75	1.3×10^{-5}	Rejected
OC1	1.925	5.8×10^{-2}	Not rejected

Control method: CE-DT, Rank: 1.325

6.3 Comparison of induction times

Finally, the induction times for the three tree-based induction methods are summarized in Table 12. DT is the fastest method for all datasets and thus has the rank 1. OC1 has a rank of 2.4 and CE-DT is in the third place with a rank of 2.6. Not surprisingly, the Friedman test reports a significant difference with a p -value of 3×10^{-7} and the post-hoc Holm test, summarized in Table 13, confirms that DT is significantly faster than CE-DT. It does, however, not provide evidence for a significant difference between the induction times of OC1 and CE-DT.

Table 12 Comparison of induction time between unpruned CE-DT, DT and OC1

Dataset	Time (ms)		
	CE-DT	DT	OC1
Balance scale	138.53±13.22	5.67 ±0.16	222.15±16.43
Breast cancer wisconsin	94.6±15.87	5.84 ±0.26	289.73±48.17
Chess	1647.28±120.6	82.03 ±1.7	12137.18±1422.07
Congressional voting records	112.35±17.09	5.78 ±0.21	148.29±26.18
Dermatology	274.55±42.99	7.39 ±0.23	124.7±3.99
Glass	150.45±14.65	2.99 ±0.15	81.23±5.18
Ionosphere	497.28±62.28	10.54 ±1.09	307.97±33.7
Monks problems 1	31.48±6.71	1.42 ±0.05	20.05±2.86
Optical recognition	11374.74±752.42	187.7 ±9.16	18827.1±4354.87
Parkinsons	180.73±46.81	4.31 ±0.15	94.04±16.12
Pen based recognition	2798.59±93.83	164.14 ±2.13	15465.42±676.45
Pima indian diabetes	447.94±15.6	8.69 ±0.29	339.56±7.28
Seeds	41.66±7.85	2.06 ±0.06	33.97±3.63
Semeion	82475.29±3690.95	365.07 ±7.02	10512.61±852.33
Spambase	25401.36±3748.26	415.67 ±22.53	10034.34±2401.98
Statlog heart	203.72±24.6	4.18 ±0.23	85.0±8.44
Statlog landsat	11076.23±528.93	235.38 ±7.52	13966.24±694.47
Statlog german credit	493.61±96.57	13.62 ±0.6	459.94±63.24
Tic tac toe	378.06±155.89	17.14 ±0.34	291.48±43.83
Wine	31.96±14.05	2.43 ±0.09	37.77±3.42
Average rank	2.6	1.0	2.4

The best results are highlighted in bold

Table 13 Holm post-hoc test for induction time of the unpruned trees

Algorithm	Rank	Holm adjusted p -value	Hypothesis
DT	1	8.4×10^{-7}	Rejected
OC1	2.4	0.53	Not rejected

Control method: CE-DT, Rank: 2.6

6.4 Discussion of the results

Overall our evaluation shows that the presented CE method is well-suited for inducing compact and accurate decision trees. The induction algorithm itself, without additional post-pruning, significantly outperforms both the univariate and the other oblique decision tree induction algorithm in terms of accuracy and tree size. We therefore conclude that the induction algorithm is more powerful than the two existing methods regarding these criteria.

With additional post-pruning, CE-DT is still significantly better in terms of accuracy. Although it still has the lowest rank for tree size, this difference can only be confirmed to be significant for DT but not for OC1. We conclude OC1 could benefit more from pruning than CE-DT. We assume that this observation can be explained by the fact that CE-DT's splits are more effective at dividing the feature space and therefore induce smaller trees in the first place. As a consequence the splits are less likely to be pruned.

Compared to DT the performance increase comes at the cost of increased induction time. This is due to the fact that determining optimal univariate splits is far less complex than finding optimal oblique splits. Still our results indicate that the increase in induction time is acceptable in many situations. Considering our results, we assume that the runtime of the algorithm is in the same order of magnitude as other oblique decision tree induction algorithms such as OC1.

In terms of accuracy CE-DT also managed to achieve the lowest rank compared to the non-tree-based classifiers NN and SVM. The fact that the difference is not significant is so far not surprising as clearly there is no prediction model that works best in all situations. Nonetheless, the result supports the claim that CE-DT is competitive with those algorithms and can even outperform them for a variety of datasets.

Combining multiple prediction models into ensembles typically yields more robust and accurate predictors than using single models. This explains why GB and RF are the most accurate classifiers in our evaluation. Although the difference to CE-DT is not confirmed to be significant, we believe that ensemble methods are generally superior in terms of pure prediction accuracy over single decision trees. The drawback is that these ensemble methods cannot be interpreted which is the main reason to use a single decision tree instead. These ensemble methods, however, are not restricted to univariate decision trees as base models and therefore, our proposed CE method can also be employed for gradient boosting or to build oblique random forests which is an interesting open topic for further research.

7 Conclusion

In this work, we illustrate how the problem of finding optimal oblique splits can be equivalently interpreted as locating an optimal region on a hypersphere intersected by hyperplanes defined by the observations in a dataset. This view on the problem motivates the application of the proposed CE algorithm that uses the von Mises–Fisher distribution. Our evaluation indicates that this method is well-suited for efficiently inducing accurate and compact oblique decision trees that are often superior to decision trees induced by existing algorithms and competitive with other non-tree-based prediction models. As a future work, we would like to test our method on real-life classification and regression tasks. Furthermore, it will be interesting to investigate the advantageous effects of our oblique decision trees on ensemble methods such as random forests or gradient boosting.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Banerjee A, Dhillon IS, Ghosh J, Sra S (2005) Clustering on the unit hypersphere using von Mises–Fisher distributions. *J Mach Learn Res* 6(Sep):1345–1382
- Bertsimas D, Dunn J (2017) Optimal classification trees. *Mach Learn* 106(7):1039–1082. <https://doi.org/10.1007/s10994-017-5633-9>
- Blanquero R, Carrizosa E, Molero-Río C, Morales DR (2020) Sparsity in optimal randomized classification trees. *Eur J Oper Res* 284(1):255–272. <https://doi.org/10.1016/j.ejor.2019.12.002>
- Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth and Brooks, Monterey
- Cantú-Paz E, Kamath C (2003) Inducing oblique decision trees with evolutionary algorithms. *IEEE Trans Evol Comput* 7(1):54–68. <https://doi.org/10.1109/TEVC.2002.806857>
- Cover TM (1965) Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans Electron Comput* 3:326–334. <https://doi.org/10.1109/PGEC.1965.264137>
- De Boer PT, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. *Ann Oper Res* 134(1):19–67. <https://doi.org/10.1007/s10479-005-5724-z>
- Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
- Dheeru D, Karra Taniskidou E (2017) UCI machine learning repository. <http://archive.ics.uci.edu/ml>. Accessed 2 October 2020

- Dhillon IS, Sra S (2003) Modeling data using directional distributions. Technical reports on TR-03-06, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, USA
- Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Ann Eugen* 7(2):179–188. <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>
- Fisher RA (1953) Dispersion on a sphere. *Proc R Soc Lond Ser A Math Phys Sci* 217(1130):295–305. <https://doi.org/10.1098/rspa.1953.0064>
- Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J Am Stat Assoc* 32(200):675–701. <https://doi.org/10.1080/01621459.1937.10503522>
- Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. *Ann Math Stat* 11(1):86–92. <https://doi.org/10.1214/aoms/1177731944>
- Heath D, Kasif S, Salzberg S (1993) Induction of oblique decision trees. In: Proceedings of the thirteenth international joint conference on artificial intelligence. Morgan Kaufmann Publishers, pp 1002–1007
- Heath DG (1993) A geometric framework for machine learning. Ph.D. Thesis, Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA
- Holm S (1979) A simple sequentially rejective multiple test procedure. *Scand J Stat* 6(2):65–70
- Li XB, Sweigart JR, Teng JT, Donohue JM, Thombs LA, Wang SM (2003) Multivariate decision trees using linear discriminants and tabu search. *IEEE Trans Syst Man Cybern Part A Syst Hum* 33(2):194–205. <https://doi.org/10.1109/TSMCA.2002.806499>
- Loh WY, Shih YS (1997) Split selection methods for classification trees. *Stat Sinica* 7(4):815–840
- López-Chau A, Cervantes J, López-García L, Lamont FG (2013) Fisher's decision tree. *Expert Syst Appl* 40(16):6283–6291. <https://doi.org/10.1016/j.eswa.2013.05.044>
- Mannor S, Peleg D, Rubinstein R (2005) The cross entropy method for classification. In: Proceedings of the 22nd international conference on machine learning, association for computing machinery, New York, NY, USA, ICML'05, pp 561–568. <https://doi.org/10.1145/1102351.1102422>
- Mardia KV, Jupp PE (1999) Directional statistics, vol 494. Wiley, Chichester
- Mola F, Siciliano R (2002) Discriminant analysis and factorial multiple splits in recursive partitioning for data mining. In: Roli F, Kittler J (eds) Multiple classifier systems. Springer, Berlin, pp 118–126. https://doi.org/10.1007/3-540-45428-4_12
- Murthy SK, Kasif S, Salzberg S, Beigel R (1993) Oc1: A randomized algorithm for building oblique decision trees. In: Proceedings of AAAI. Citeseer, pp 322–327
- Murthy SK, Kasif S, Salzberg S (1994) A system for induction of oblique decision trees. *J Artif Intell Res* 2:1–32. <https://doi.org/10.1613/jair.63>
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
- Rubinstein R (1999) The cross-entropy method for combinatorial and continuous optimization. *Methodol Comput Appl Probab* 1(2):127–190. <https://doi.org/10.1023/A:1010091220143>
- Rubinstein RY (1997) Optimization of computer simulation models with rare events. *Eur J Oper Res* 99(1):89–112. [https://doi.org/10.1016/S0377-2217\(96\)00385-2](https://doi.org/10.1016/S0377-2217(96)00385-2)
- Rubinstein RY, Kroese DP (2004) The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning, vol 133. Springer, Berlin
- Siciliano R, Aria M, D'Ambrosio A (2008) Posterior prediction modelling of optimal trees. In: Brito P (ed) COMPSTAT 2008, Physica-Verlag HD, Heidelberg, pp 323–334. https://doi.org/10.1007/978-3-7908-2084-3_27
- Sra S (2012) A short note on parameter approximation for von Mises–Fisher distributions: and a fast implementation of $I_S(x)$. *Comput Stat* 27(1):177–190. <https://doi.org/10.1007/s00180-011-0232-x>
- Truong AKY (2009) Fast growing and interpretable oblique trees via logistic regression models. Ph.D. Thesis, Oxford University, Oxford, United Kingdom
- Ulrich G (1984) Computer generation of distributions on the m-sphere. *J R Stat Soc Ser C (Appl Stat)* 33(2):158–163. <https://doi.org/10.2307/2347441>
- Wickramarachchi D, Robertson B, Reale M, Price C, Brown J (2016) HHCART: an oblique decision tree. *Comput Stat Data Anal* 96:12–23. <https://doi.org/10.1016/j.csda.2015.11.006>
- Wood AT (1994) Simulation of the von Mises–Fisher distribution. *Commun Stat Simul Comput* 23(1):157–164. <https://doi.org/10.1080/03610919408813161>
- Zlochín M, Birattari M, Meuleau N, Dorigo M (2004) Model-based search for combinatorial optimization: a critical survey. *Ann Oper Res* 131(1–4):373–395. <https://doi.org/10.1023/B:ANOR.0000039526.52305.af>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.