Variational oblique predictive clustering trees for multi-target regression

Viktor Andonovikj[a,b,*], Sašo Džeroski[a], Biljana Mileva Boshkoska[a,c], Pavle Boškoski[a]

[a] *Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[b] *Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[c] *Faculty of Information Studies in Novo mesto, Ljubljanska cesta 31b, 8000 Novo mesto, Slovenia*

Abstract

Oblique predictive clustering trees (SPYCTs) are semi-supervised multi-target prediction models mainly used for structured output prediction (SOP) problems. They are computationally efficient and when combined in ensembles they achieve state-of-the-art results. However, one major issue is that it's challenging to interpret an ensemble of SPYCTs without the use of a model-agnostic method. We propose variational oblique predictive clustering trees, which address this challenge. The parameters of each split node are treated as distributions and they are learned through the Variational Bayes method. We evaluate the model on several benchmark datasets of different sizes. The experimental analyses show that a single variational oblique predictive clustering tree (VSPYCT) achieves state-of-the-art or on-par performance with ensemble of standard SPYCTs. We also present a method for extracting feature importance scores from the model. Finally, we present a method to visually interpret the model's decision making process through analysis of the relative feature importance in each split node.

1. Introduction

In the evolving field of machine learning, the fusion of semi-supervised learning and decision tree models has led to the development of models like predictive clustering trees (PCTs) (Kocev et al., 2013) and their advanced variant, oblique predictive clustering trees (SPYCTs) (Stepišnik and Kocev, 2021). These models leverage both labeled and unlabelled data to enhance prediction accuracy and efficiency. SPYCTs, in particular, have pioneered the use of oblique splits for structured output prediction (SOP), allowing for more intricate decision boundaries beyond the capabilities of traditional axis-parallel splits. Despite their advantages, a significant challenge with SPYCTs is their reliance on ensemble methods for optimal performance. While effective in improving accuracy, ensembles tend to obscure the model's interpretability—a fundamental attribute of decision trees—and do not inherently provide a means to quantify prediction uncertainty, crucial for informed decision-making across various applications.

*Corresponding author.
Email addresses: viktor.andonovikj@ijs.si (Viktor Andonovikj), saso.dzeroski@ijs.si (Sašo Džeroski), biljana.mileva@ijs.si (Biljana Mileva Boshkoska), pavle.boskoski@ijs.si (Pavle Boškoski)

To address these limitations, we introduce the variational oblique predictive clustering tree (VSPYCT) model. VSPYCT integrates the variational Bayes method (Blei et al., 2017) to facilitate complex decision-making within a singular model framework, thus maintaining the interpretability inherent to decision trees without necessitating ensembles. Furthermore, by embedding model-specific uncertainty quantification directly into the decision tree structure through Bayesian inference, VSPYCT enhances the depth of insight into the model's decision process and confidence levels. By combining the robustness of SPYCT ensembles with the clarity and interpretability of a single tree model, alongside introducing uncertainty quantification, our research provides a comprehensive tool aimed to overcome current limitations of predictive clustering methodologies.

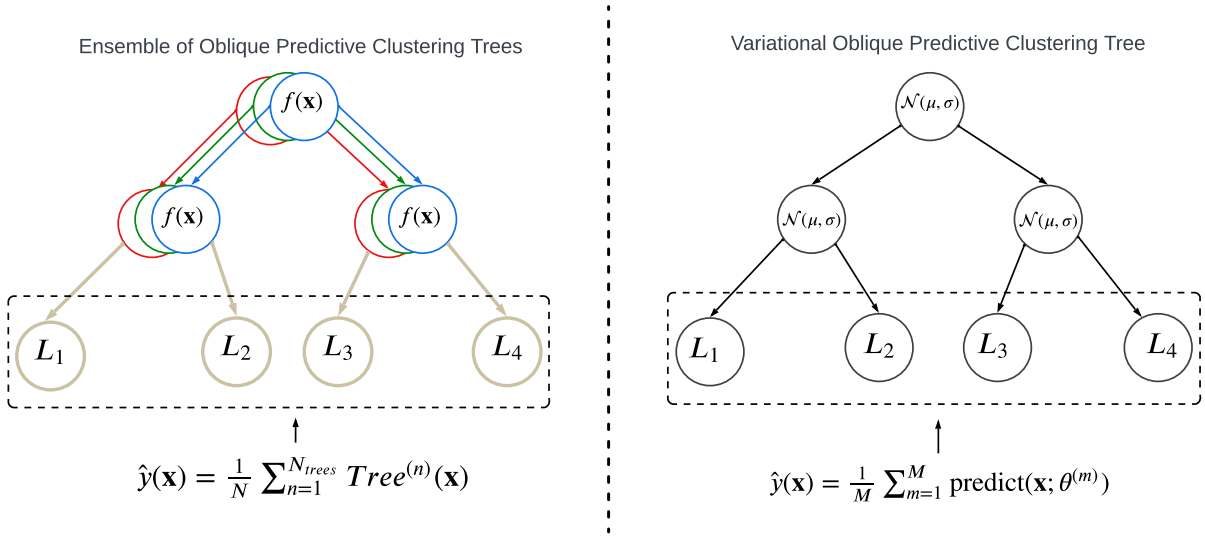

Figure 1: Ensemble of trees vs. Variational tree

Our proposed VSPYCT framework diverges from traditional ensemble approaches by directly incorporating Bayesian principles into oblique decision trees' architecture. This model avoids conventional ensemble strategies, which aggregate multiple trees' outputs, for a probabilistic treatment of model parameters, offering refined mechanisms for uncertainty quantification and predictive performance. The core of VSPYCT's innovation lies in applying the variational Bayes method for optimising parameters defining the oblique splits, modelling these parameters as random variables with Gaussian distributions. This represents a paradigm shift toward a probabilistic understanding of decision-making processes within tree-based models, significantly enhancing interpretability, especially in uncertainty estimation domains. The whole framework is given in Figure 2

It sets a new benchmark for interpretable, efficient, and reliable machine learning models.

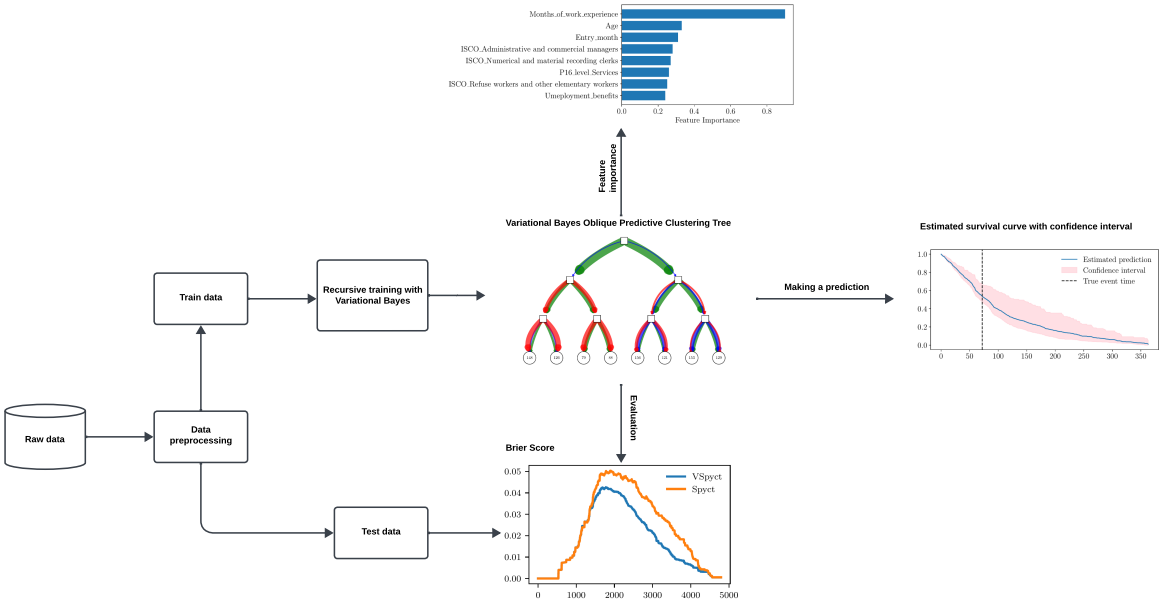

Figure 2: Flowchart of the methodology

2. Related work

PCTs have been an essential development in extending decision tree capabilities to a variety of predictive modelling tasks, including structured output prediction. PCTs are versatile and can be combined into ensembles to achieve state-of-the-art performance (Kocev et al., 2013). However, as the dimensionality of the output space increases, the learning time of PCTs scales poorly, posing challenges in tasks like hierarchical multi-label classification where outputs can consist of hundreds of potential labels.

PCTs are celebrated for their interpretability and efficiency, yet they often fall short in predictive performance due to their myopic, greedy nature of selecting splits. The introduction of option predictive clustering trees (OPCTs) (Stepisnik et al., 2020) aimed to address this limitation by incorporating multiple alternative splits within option nodes. This approach mitigates the inherent myopia and achieves competitive performance with ensemble methods like bagging and random forests, while still maintaining a degree of interpretability. OPCTs, despite mitigating myopia and enhancing performance, still face challenges in maintaining interpretability when extended to complex tasks.

On the other hand, SPYCTs (Stepišnik and Kocev, 2021) utilise oblique splits that incorporate linear combinations of features, allowing splits to correspond to arbitrary hyperplanes in the input space. This makes SPYCTs highly efficient for high-dimensional data and capable of leveraging data sparsity. Experimental evaluations on numerous benchmark datasets have shown that SPYCTs achieve performance on par with state-of-the-art methods while being significantly faster than standard PCTs (Andonovikj et al., 2024).

Despite these advancements, ensemble models such as random forests and gradient boosting machines remain popular due to their superior predictive accuracy. However, their complexity often obscures interpretability, making it challenging for users to understand the model's decision-making process. Efforts to bridge this gap include the iForest visual analytics system (Zhao et al., 2019), which summarises decision paths and tree structures within random forests, thereby making the ensemble's decisions more transparent. Additionally, the Tree Space Prototypes approach (Tan et al., 2020) uses representative points (prototypes) for each class, offering a more intuitive understanding of ensemble classifiers compared to traditional feature-based explanations. Visual analytics systems like iForest and prototype-based approaches offer some clarity but do not fully resolve the complexity issue.

In critical applications, such as medical diagnosis, the need for transparent and trustworthy predictions is paramount. Methods for explaining classifier predictions and estimating the reliability of regression predictions, as demonstrated in breast cancer recurrence prediction, provide users with additional insights and build trust by clarifying the decision-making process. Similarly, the MAPLE (Plumb et al., 2018) model combines local linear modelling with a dual interpretation of random forests, offering faithful self-explanations and maintaining high predictive accuracy. MAPLE addresses the accuracy-interpretability trade-off and provides both example-based and local explanations, making it a comprehensive tool for understanding model behaviour. Methods like MAPLE and other interpretability frameworks strike a balance between accuracy and transparency, but they often fall short in handling high-dimensional, sparse data efficiently.

While significant progress has been made in improving decision tree models and their ensembles, several weaknesses remain. Many of these approaches lack inherent mechanisms for uncertainty quantification, which is crucial for applications requiring high reliability and decision certainty.

Our proposed VSPYCTs model addresses these critical challenges by integrating Bayesian principles directly into the decision tree framework. This approach maintains the clarity and simplicity of a single tree model while achieving state-of-the-art performance, thus offering a significant improvement over existing methods. By introducing uncertainty quantification directly into the decision-making process, VSPYCTs enhances the reliability and applicability of the model across various domains where decision certainty is crucial.

3. Preliminaries

3.1. Variational Bayes method

In conducting a stochastic analysis, like survival analysis, the process of inferring model parameters hinges on applying Bayes' theorem. For observations x produced by a system defined by parameters θ, Bayes' theorem is given as:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \tag{1}$$

Here, the model-prescribed likelihood $p(x|\theta)$ and the selected prior $p(\theta)$ are generally well-defined. The primary computational challenge arises in determining the posterior distribution $p(\theta|x)$ due to the denominator $p(x)$, which is derived as:

$$p(x) = \int_{\theta} p(x|\theta)p(\theta)\, dx, \tag{2}$$

This integral rarely yields a closed-form solution. For cases with multiple dimensions, the computational burden often renders Monte Carlo methods infeasible. The VB approach offers an approximation to this issue by closely mimicking the true posterior $p(\theta|x)$ with an approximate distribution $q_{\omega^*}(\theta)$. This variational distribution is part of a family of distributions $q_\omega(\theta) \in \mathcal{Q}$, parameterized by $\omega \in \Omega$, where Ω represents the potential latent parameter values. Typically, the mean-field variational family is utilized, assuming independence among the parameters ω. To find the optimal ω^*, one minimises the Kullback–Leibler (KL) divergence $KL(q_\omega(\theta) \parallel p(\theta|x))$:

$$\omega^* = \arg\min_{\omega \in \Omega} KL(q_\omega(\theta) \parallel p(\theta|x)) \tag{3}$$

Given the unknown nature of the true posterior, a rearrangement is needed to compute the KL divergence, simplifying to:

$$\begin{aligned} KL(q_\omega(\theta) \parallel p(\theta|x)) &= \mathbb{E}_q \left[\log \frac{q_\omega(\theta)}{p(\theta|x)}\right] \\ &= \mathbb{E}_q \left[\log q_\omega(\theta)\right] - \mathbb{E}_q \left[\log p(\theta|x)\right] \\ &= \mathbb{E}_q \left[\log q_\omega(\theta)\right] - \mathbb{E}_q \left[\log p(x, \theta)\right] \\ &= \mathbb{E}_q [\log q_\omega(\theta) - \log p(x, \theta) + \log p(x)] \\ &= \underbrace{-\mathbb{E}_q \left[\log q_\omega(\theta) - \log q_\omega(\theta)\right]}_{\text{ELBO}} + \log p(x) \end{aligned}$$

Maximising the ELBO inversely minimises the KL divergence, with $\log p(x)$ remaining constant and excluded from the optimisation. Typically, the ELBO maximisation criterion is non-convex, with no assurance of global extremum convergence by optimisation algorithms. Various algorithms have been employed for this challenge, including coordinate ascent variational inference and stochastic variational inference Hoffman et al. (2013), with the latter's gradient calculated as per the black-box variational inference Sashank et al. (2018). The adoption of an approximate variational distribution introduces bias contingent on the chosen

variational family $\mathcal{Q}$, necessitating a decision grounded in empirical evidence or expert knowledge. Despite this bias, VB's computational efficiency significantly surpasses traditional methods like Monte Carlo integration. In our work, the ADAM optimiser (Kingma and Ba, 2014) is used to facilitate the ELBO optimisation.

3.2. Semi-supervised Multi-target Regression in Survival Analysis

Survival analysis primarily focuses on time-to-event data, where not all events are fully observed due to some form of censoring that halts observation before the event transpires. The most prevalent form of censoring in this context is right-censoring, which occurs when there is incomplete knowledge about the survival time at the end of the follow-up period. As a result, each survival record is characterised by three elements: the actual survival time T, its observed component t, and a binary indicator δ, which is defined as:

$$\delta = \begin{cases} 0, & \text{if } T > t \text{ indicating a right-censored record,} \\ 1, & \text{if } T = t, \text{ denoting that the event has been observed and the record is not censored.} \end{cases} \tag{4}$$

Moreover, assume that each observation is described by a set of covariates or features $\mathbf{X}_i = \{x_{i_1}, \ldots, x_{i_N}\}$. Thus, each record i comprises the variables $\mathbf{X}_i$, t_i, and δ_i as specified in (4). If $\delta_k = 1$, the event is confirmed to have occurred at time t_k and $y_{k_j} = 0$ for all $j \geq k$. Conversely, if $\delta_k = 0$, it implies that the fate of the instance beyond time t_k is unknown, making y_{k_j} missing for $j \geq t_k$. This scenario of censored examples introduces missing data, which makes such datasets apt for semi-supervised learning methodologies."

4. Methodology

As a main part of the VSPYCT model lies the oblique tree structure, which is a form of a PCT where splits are made using linear combinations of the input features rather than single features. This allows for more complex decision boundaries, capturing intricate patterns in the data with fewer nodes compared to axis-aligned trees. However, unlike traditional oblique trees that fix the parameters of these linear combinations, VSPYCT introduces a probabilistic twist: the parameters of the splits are not single fixed values but are instead modelled as random variables with Gaussian distributions. This probabilistic representation captures the uncertainty in the model parameters, a feature that is particularly useful in scenarios with noisy data. The construction of the VSPYCT begins with a root node and iteratively expands by adding internal nodes or leaves through a learning process based on the variational Bayes method. Each node represents a binary decision point where instances are split into two child nodes. At each internal node, a variational Bayes approach is used to learn the parameters of an oblique split. This involves optimising the parameters of a linear model that best divides the data to minimise the ELBO. The linear model is defined as:

$$f(\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + b\right) \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the feature vector, $\mathbf{w} \in \mathbb{R}^N$ represents the weights, $b \in \mathbb{R}$ is the bias, and $\sigma(\cdot)$ denotes the sigmoid function ensuring the output lies in $(0, 1)$, indicating the probability of branching to the right child node. The learning process involves defining a prior distribution over the weights and biases, typically Gaussian, and updating this to a posterior distribution given the observed data through variational inference. The ELBO is optimised using stochastic gradient descent, with the learning rate α and potentially variable batch sizes B. The split at each node is determined by evaluating the learned linear model on the descriptive data. Instances are split based on the probability threshold of 0.5, with those having a probability greater than 0.5 moving to the right child node and the rest to the left. For leaf nodes, a prototype is computed as the mean of the target variables for the instances assigned to the leaf. This prototype represents the prediction for instances that fall into the leaf during the prediction phase.

4.1. Learning splits using Variational Bayes

The learning of these Gaussian-distributed parameters is achieved through a Variational Bayes (VB) method. In the context of VSPYCT, the VB method is employed to learn the Gaussian distributions representing the parameters of the oblique splits. The process involves defining a prior distribution over the parameters and then updating this prior to a posterior distribution given the observed data. This is done by optimising the ELBO, which balances the fit of the model to the data with the complexity of the model. The procedure for learning a split using the VB method is given in Algorithm 1.

4.2. Making a prediction

The procedure for making a prediction for a new instance using the VSPYCT model is given in Algorithm 3. It involves traversing the tree from the root to a leaf, at each internal node using the variational posterior to sample parameters and compute the probability of branching right or left. The prediction of the reached leaf's prototype is returned as the output. Operationally, when a new data point is to be classified or a prediction is to be made, it traverses the tree from the root to a leaf. At each node, the oblique split is determined by evaluating the linear combination of the input features, with parameters sampled from the learned Gaussian distributions. This probabilistic approach to decision-making within the tree allows for a more nuanced interpretation of the data, accounting for uncertainty in the model parameters.

4.3. Time complexity analysis

In this section, we analyse the time complexity of learning a split in VSPYCT and compare it with SPYCT. Our analysis focuses on the computational expense associated with determining a split, as this represents the principal distinction between the two approaches. In the predictive clustering framework,

Algorithm 1 Variational Learning of Split Parameters in VSPYCT

Input: Dataset $\mathcal{D}$, features $\mathbf{X}$, targets $\mathbf{Y}$, parameters θ, epochs E, batch size β, learning rate λ, selection probability σ
Output: Parameters Θ, guide $\mathcal{G}$
procedure LearnSplit($\mathcal{D}$, $\mathbf{X}$, $\mathbf{Y}$, θ, E, λ, β, σ)
    $\Phi \sim \text{Bernoulli}(\sigma)$ ▷ Select attributes based on probability σ
    $\mathbf{X}_\Phi \leftarrow$ subset of $\mathbf{X}$ using Φ
    Model $\mathcal{M} \leftarrow$ Define Impurity($\dim(\mathbf{X}_\Phi), 1, \theta$)
    Guide $\mathcal{G} \leftarrow$ Define AutoDiagonalNormal($\mathcal{M}$)
    Define optimization parameters $\alpha \leftarrow$ {"lr" : λ}
    Initialise optimizer Optimizer $\leftarrow$ Adam(α)
    Early stopping mechanism EarlyStopping $\leftarrow$ Initialize(β)
    Initialise loss list L
    Calculate number of batches $B \leftarrow \lceil \frac{|\mathcal{D}|}{\beta} \rceil$
    for $e \in \{1, \ldots, E\}$ **do**
        Initialise training loss $\mathcal{L}_{\text{train}} \leftarrow 0$
        for $b \in \{1, \ldots, B\}$ **do**
            $(\mathbf{X}_b, \mathbf{Y}_b) \leftarrow$ Get Batch($\mathbf{X}_\Phi, \mathbf{Y}, b, \beta$)
            Update training loss $\mathcal{L}_{\text{train}} \leftarrow \mathcal{L}_{\text{train}} +$ SVI Step($\mathcal{M}, \mathcal{G}, \mathbf{X}_b, \mathbf{Y}_b$)
        end for
        Append $\mathcal{L}_{\text{train}}$ to L
        if EarlyStopping($\mathcal{L}_{\text{train}}$) **then**
            break
        end if
    end for
    return $\mathcal{M}$, $\mathcal{G}$
end procedure

the set of clustering attributes K, which contributes to the variance calculation, encompasses both the features and the target variables. As a result, K equals $N+T$, with N representing the feature count and T denoting the target count. The time complexity for learning in SPYCTs can be expressed as $\mathcal{O}(NI_o(D+K))$, wherein I_o stands for the iteration count needed for optimisation. Notably, the computational demand for the SPYCTs' gradient-based version increases linearly with the attribute count.

The introduction of variational Bayes method in the architecture of SPYCT to create VSPYCT adds a new layer of computational complexity. Specifically, the variational approach requires iterating over the

Algorithm 2 Impurity

procedure IMPURITY(ξ, ζ, θ)
  $\mathbf{w}, b \sim \mathcal{N}(0, I)$ ▷ Weight and bias priors
  $\rho(\mathbf{x}) \leftarrow \sigma(\mathbf{w}^\top \mathbf{x} + b)$ ▷ Right selection probability
  $\lambda(\mathbf{x}) \leftarrow 1 - \rho(\mathbf{x})$ ▷ Left selection probability
  Compute $\mathrm{Var}_\rho(\mathbf{Y})$ and $\mathrm{Var}_\lambda(\mathbf{Y})$ ▷ Weighted variances
  $\Omega \leftarrow \lambda \cdot \mathrm{Var}_\lambda(\mathbf{Y}) + \rho \cdot \mathrm{Var}_\rho(\mathbf{Y}) + \|\mathbf{w}\|_{0.5}$
  return Ω
end procedure

dataset multiple times to optimise the ELBO for each split. This iterative process involves sampling from the posterior distributions of split parameters, evaluating the ELBO, and updating the parameters accordingly. Consequently, the time complexity for learning a split in VSPYCT can be described as $\mathcal{O}(MI_oNI_{vb})$, where M is the number of Monte Carlo samples used to approximate the posterior distributions, I_{vb} represents the number of iterations required for variational inference to converge, and the other variables retain their meanings from the SPYCT context.

Comparing VSPYCTs to SPYCTs, the introduction of variational Bayes method increases the computational load due to the necessity of Monte Carlo sampling and iterative ELBO optimisation. However, this additional complexity is counterbalanced by the enhanced predictive performance, model interpretability, and uncertainty quantification offered by VSPYCTs.

The choice between using VSPYCTs over SPYCTs hinges on the specific requirements of the application scenario. For tasks where model interpretability and uncertainty quantification are paramount, and computational resources are sufficient, VSPYCTs offer a compelling advantage. Conversely, for applications where computational efficiency is critical, and the interpretability requirements are less stringent, SPYCTs may still be favourable.

4.4. Feature importance

Feature importance is derived from the aggregated impact of features across all splits, quantified by examining the variational posterior distributions of the weights. The importance of each feature is proportional to its contribution to reducing impurity across splits.

Given a VSPYCT model, the feature importances are derived from the variational parameters defining the split decisions across the tree. The essence of the computation lies in evaluating the expected influence of each feature, taking into account the probabilistic nature of the model parameters. This is encapsulated by the following formula:

Algorithm 3 VSPYCT Prediction Process

Input: Feature vector $\mathbf{x}$, decision tree T rooted at node Ω
Output: Prediction $\hat{y}$
function PREDICT(Ω, $\mathbf{x}$)
    if Ω is a leaf **then**
        return $\Omega.prototype$
    else
        $\Theta \leftarrow$ SAMPLEPARAMETERS($\Omega.split\_model$, $\Omega.guide$)
        Define sigmoid function $\rho(\Theta, \mathbf{x}) = \frac{1}{1+\exp(-\Theta^\top \mathbf{x})}$
        if $\rho(\Theta, \mathbf{x}) \leq 0.5$ **then**
            return PREDICT($\Omega.left$, $\mathbf{x}$)
        else
            return PREDICT($\Omega.right$, $\mathbf{x}$)
        end if
    end if
end function
function SAMPLEPARAMETERS(model, guide)
    Let $\Theta = (\mathbf{w}, b)$ where:
    $\mathbf{w}$ are weights sampled from variational posterior $q_\omega(\mathbf{w}|\mathcal{D})$
    b is bias sampled from variational posterior $q_\omega(b|\mathcal{D})$
    return Θ
end function
Compute prediction $\hat{y}$ by invoking PREDICT(Ω, $\mathbf{x}$)
Return $\hat{y}$

$$\text{Imp}(T) = \sum_{s \in T} \left(\frac{s_n}{N} \right) \left(\frac{\mathbb{E}[\mathbf{w}_s]}{|\mathbb{E}[\mathbf{w}_s]|_1} \right) \tag{6}$$

where T represents the variational oblique predictive clustering tree, s iterates over the set of split nodes within T, s_n is the number of training examples that passed through node s, N denotes the total number of training examples used in the tree, $\mathbf{w}_s$ symbolizes the weight vector defining the oblique split at node s, treated as a random variable with a Gaussian distribution, $\mathbb{E}[\mathbf{w}_s]$ signifies the expected value (mean) of the weight vector $\mathbf{w}_s$, calculated from its posterior distribution as inferred by the variational Bayesian method. The normalization factor $|\mathbb{E}[\mathbf{w}_s]|_1$ ensures the comparability of feature importances by scaling them relative to the aggregate magnitude of expected weights. This formulation accounts for the stochastic nature of each parameter, reflecting the central tendency of the distributions defining the split hyperplanes.

For practical computation, the expected weights $\mathbb{E}[\mathbf{w}_s]$ are obtained by sampling from the posterior distributions inferred during the learning process. Each weight vector $\mathbf{w}_s$ is drawn such that $\mathbf{w}_s \sim q(\mathbf{w}_s|\mathcal{D})$, where $q(\mathbf{w}_s|\mathcal{D})$ represents the variational approximation to the posterior distribution of weights given the data $\mathcal{D}$. The importance of each feature is then a function of these expected weights, modulated by the frequency of training instances influencing each split, thereby emphasising the splits that impact a larger portion of the data. This mathematical framework for computing feature importances in VSPYCT not only highlights the contribution of individual features to the model's decision-making process but also integrates the inherent uncertainty of these contributions, offering a richer, more nuanced understanding of feature relevance within a probabilistic modelling context.

5. Data and evaluation metric

5.1. Data

5.2. R Survival datasets

Our evaluation encompasses four benchmark survival analysis datasets, each serving as a standard reference within the domain and accessible via the survival R package (Therneau, 2017).

1. *Retinopathy Dataset*: Investigates the efficacy of laser coagulation in delaying vision loss due to diabetic retinopathy among 197 patients, with survival times adjusted for a minimum event horizon of 6.5 months.
2. *Lung Dataset*: Analyses survival data for patients with advanced lung cancer, focusing on their lifespan post-diagnosis and daily activity performance levels.
3. *Primary Biliary Cirrhosis (PBC) Dataset*: Examines the progression of primary biliary cholangitis in 424 patients, with data drawn from a clinical trial conducted between 1974 and 1984, exploring disease progression towards cirrhosis.

4. *Heart Dataset*: Studies the survival of patients on the waiting list for heart transplants at Stanford, evaluating the length of time patients remain on the waiting list.

Table 1: Main properties of the datasets. Censoring shows the percentage of censored observations.

Dataset	# of examples	# of features	Censoring
Mayo Clinic Primary Biliary Cholangitis	312	17	60%
NCCTG Lung Cancer	228	8	
Stanford Heart Transplant	172	4	
Retinopathy	394	9	

5.3. Unemployment dataset

The dataset we consider consists of $74,086$ instances and contains anonymised personal and professional characteristics of jobseekers that have entered the Slovenian Public Employment Services (PES). The dataset is complex because its attributes come in different forms (categorical, numerical, date and time), and most have to undergo transformation. An overview of the dataset and its features is given in Table 2.

Table 2: Features of the dataset and their cardinalities (Andonovikj et al., 2024)

Continuous features	Discrete features
Day of PES entry	Education direction (30)
Months of work experience	Occupation (ISCO) (123)
Age	Reason for PES entry (5)
Month of PES entry	Reason for termination (43)
	Social benefits (2)
	Gender (2)
	eApplication (2)
	Unemployment benefits (2)

The general structure of the jobseekers' attributes is described by dividing the attributes into several prominent groups: socioeconomic variables (gender, age), information on job readiness (education, health limitations), and opportunities (regional labour market development), and all available labour market history information, such as prior work experience. The target variable is in numeric form. It is a counter of days till a jobseeker gets employed or gets out of the study and leaves no further information about the jobseeker's employment status.

Figure 3 shows the distributions of some features in the dataset. The sample of jobseekers is balanced in terms of gender. Most jobseekers enter the PES without any working experience. As expected, the age distribution is between 18 and 64 since this is the legal age interval of the labour-active population. It should be noted that the age distribution is bi-modal. The first peak is around 23-26 years since this is the expected graduation age for a university degree. The second peak is just before 60 years of age, i.e. jobseekers who became unemployed just before retirement due to various reasons. The final histogram shows that the dataset contains 30% of right censored data.

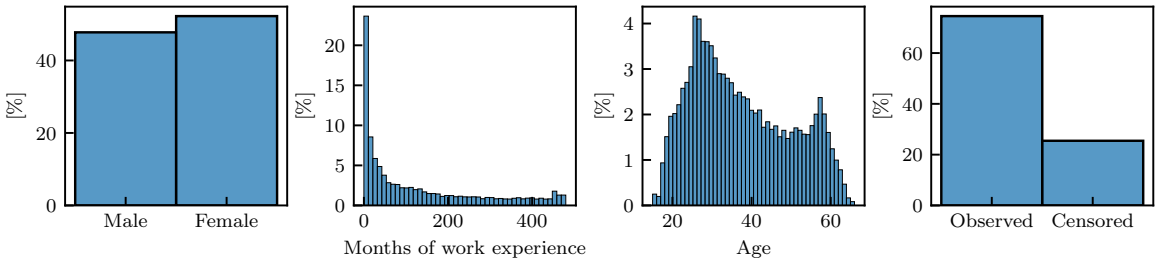

Figure 3: Distributions of some of the dataset features (Andonovikj et al., 2024)

The categorical features are diverse with respect to their cardinality. The categories of the features with high cardinality (> 1000 different class values per feature) are transformed using domain knowledge and the number of class values is significantly reduced. Because the *ISCO* standard and *Education direction* provide a hierarchical classification of occupations and education directions, we took the higher-level classifications and thus significantly reduced the number of categories. The remaining categorical features were encoded using one-hot encoding. The temporal (cyclical) features, such as "Day of PES entry" and "Month of PES entry" were transformed by calculating the feature's sin and cos components. In this way, the artificially significant difference between, for example, the last day of the previous month and the first day of the current month is removed, and the value of the feature can be viewed as a point $(\sin(x), \cos(x))$ of a circle.

5.4. Evaluation metric

The efficacy of survival models is predominantly assessed through the Concordance Index (Brentnall and Cuzick, 2016) and the Brier score (Gerds and Schumacher, 2006). While the Concordance Index offers valuable insights, its inability to account for the error magnitude in predicted probabilities limits its utility. The Brier score addresses this gap by factoring in the error magnitude, making it a more comprehensive metric for our experimental evaluation. The Brier score assesses the precision of a forecasted survival function at a specific time t, quantifying the mean squared differences between observed survival statuses and forecasted survival probabilities. It ranges from 0 to 1, where a score of 0 indicates perfect accuracy. This metric not only assesses the precision of time-to-event forecasts, akin to a squared error loss, but also

necessitates adjustments for right-censored samples in the dataset. This adjustment employs the inverse probability of censoring weights method to weight the squared differences.

The conditional survival function estimator for censoring times, $\hat{G}(t) = P(C > t)$, utilises the Kaplan-Meier method, with C denoting the censoring time. The Brier score is thus calculated as (Kvamme and Borgan, 2023):

$$BS(t) = \frac{1}{M} \sum_{i=1}^{N} \left(\frac{\hat{S}(t_i, \mathbf{X}i) \cdot 1T_i \geq t, \delta_i = 1}{\hat{G}(T_i^-)} + \frac{(1 - \hat{S}(t_i, \mathbf{X}i)) \cdot 1T_i > t, \delta_i = 0}{\hat{G}(T_i)} \right) \tag{7}$$

Here, M signifies the count of test instances, $\mathbf{X}_i$ the feature vector for each instance, $\hat{S}(t, \mathbf{X}_i)$ the predicted survival probability at time t_i, δ_i a boolean indicator, and T_i and T_i^- the survival times for δ_i being 0 and 1, respectively. The Brier score is computed at various timestamps to evaluate how the method's performance varies with predictions extended into the future.

The performance was estimated by using k-fold cross-validation with the parameter k set to 5 (Hastie et al., 2009). The prediction model yields an estimated survival curve for each individual in the study.

6. Results

The resulting BS curves are shown in Figure 4. It should be noted that lower BS indicates a more accurate forecast. The analysis revealed distinct performance patterns between the two models across different datasets. Specifically, in the PBC and lung cancer datasets, VSPYCT outperformed SPYCT, evidenced by consistently lower Brier scores throughout the observation period. This superior performance of VSPYCT could be attributed to its robust handling of parameter uncertainty through the VB method, which likely improved its predictive accuracy in these contexts. The VSPYCT model demonstrated a particularly notable advantage in maintaining lower peak Brier scores, suggesting enhanced reliability in predicting survival outcomes for patients with PBC and lung cancer.

Conversely, in the heart transplant and retinopathy datasets, SPYCT showed better performance compared to VSPYCT, as indicated by lower Brier scores over time. This suggests that the simpler, more direct approach of SPYCT may be more effective in certain contexts, possibly due to less complex data structures or the nature of the underlying survival processes in these specific conditions. The lower Brier scores achieved by SPYCT in these cases highlight its efficiency in accurately predicting the survival outcomes of heart transplant patients and those undergoing treatment for diabetic retinopathy.

6.1. Interpretability

Figure 5 presents the visual representation of the VSPYCT applied to the Unemployment dataset. Based on that, one can distill a concise interpretation enriched with practical insights.

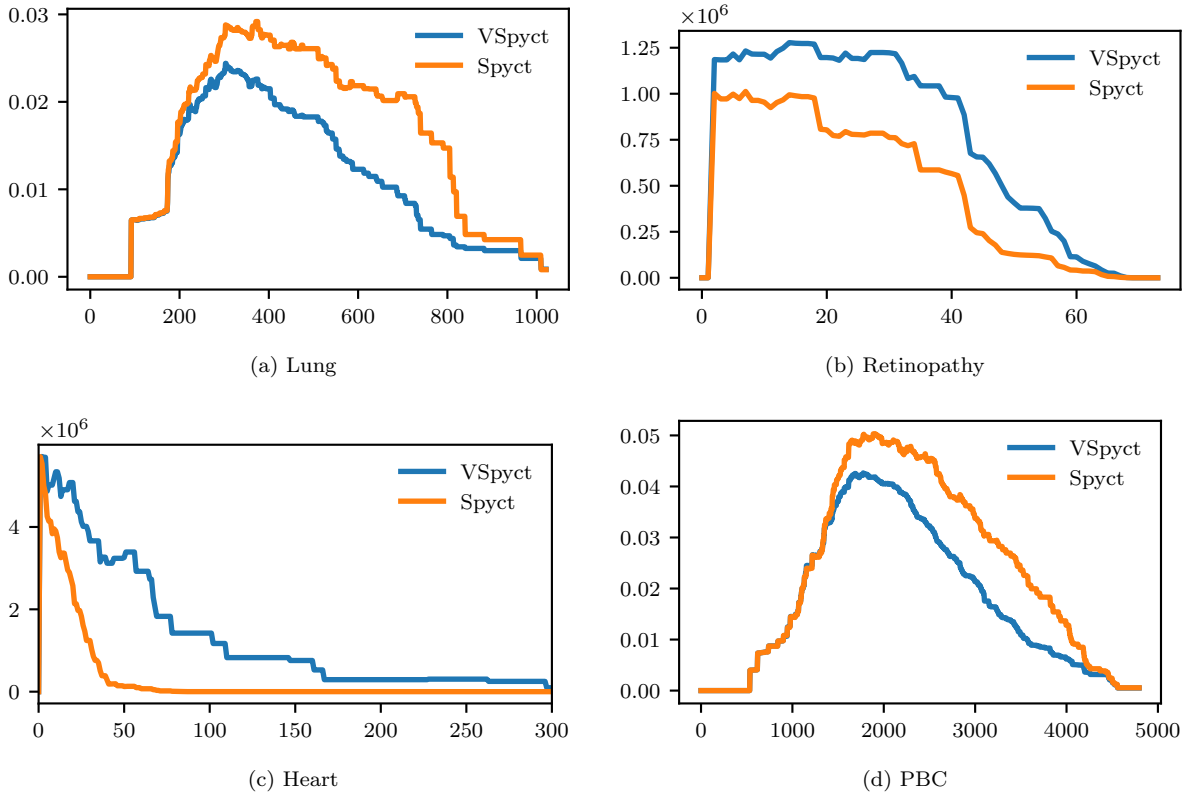

Figure 4: Performance evaluation of VSPYCT and SPYCT models measured through the Brier score on the R survival datasets

At the heart of this VSPYCT model lies the ability to visually demarcate the influence of various feature categories on unemployment duration predictions. The tree's root node is predominantly influenced by features categorised under "other" (represented in green), encompassing age, months of work experience, and various demographic characteristics. This indicates that such features play a pivotal role in initial split decisions, underscoring the broad impact of demographic and experiential factors on unemployment trends.

As we navigate through the tree, a dichotomy emerges based on the expected survival times—essentially, the projected duration an individual is likely to remain unemployed. On branches leading to higher expected survival times, features categorised under "p16 level of education" (coloured in blue) become increasingly dominant. This suggests that higher levels of education are associated with longer periods of unemployment, potentially reflecting challenges in finding employment that matches an individual's qualifications or aspirations.

Conversely, on branches associated with lower expected survival times, the dominance shifts towards features represented by the "ISCO classification of professional occupation" (coloured in red). This dominance implies that certain professional occupations, as defined by the ISCO classification, may expedite reemployment, possibly due to higher demand or greater flexibility in job roles within these sectors.

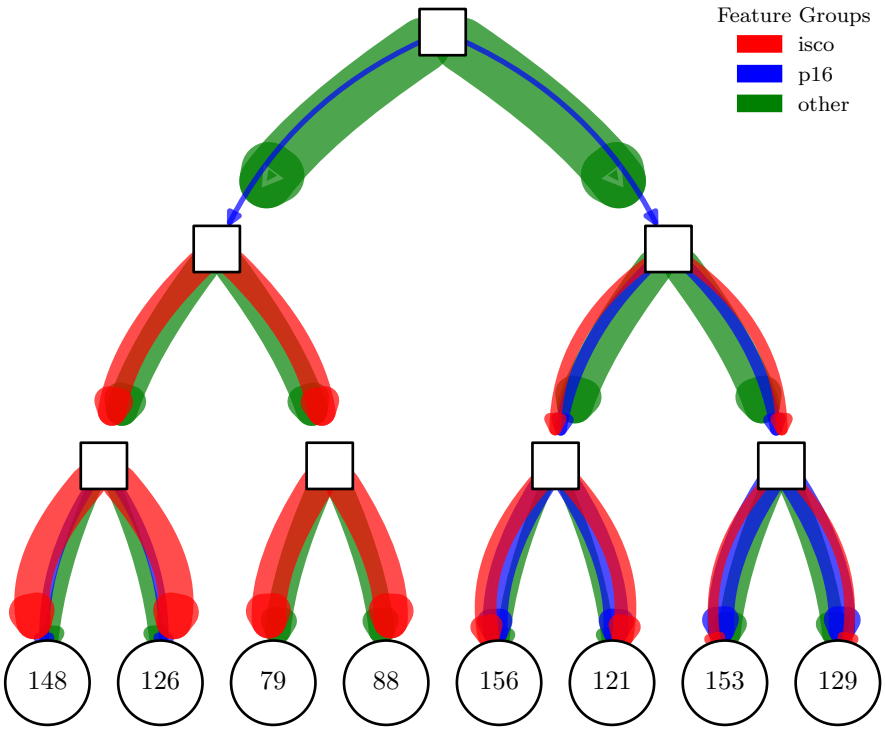

Figure 5: The VSPYCT model applied to the Unemployment dataset. The colouring distinguishes the different feature groups of the data set

This visual breakdown not only highlights the model's interpretability but also offers tangible insights into how different factors—ranging from personal demographics and work experience to education and professional classifications—impact unemployment durations. Such insights are invaluable for policymakers and stakeholders aiming to devise targeted interventions that address specific barriers to employment faced by diverse demographic groups.

In essence, the VSPYCT model's tree structure, with its intuitive colour-coded feature categories and quantified predictions at the leaf nodes, offers a powerful tool for understanding the multifaceted nature of unemployment. It bridges the gap between complex machine learning models and practical socio-economic applications, providing a clear, actionable framework for addressing unemployment through informed, data-driven strategies.

7. Conclusion

We introduced the VSPYCT model, an approach that strategically combines the benefits of oblique splits, variational Bayes, and Bayesian inference within a single, interpretable framework. The VSPYCT model addresses the trade-off between the performance of ensemble models and the interpretability of single decision trees, and the absence of inherent uncertainty quantification in predictions. Our results demonstrate that the VSPYCT model is competitive to, and on some data sets exceeds the performance of ensemble of SPYCTs, without sacrificing the model's interpretability. By employing variational Bayes method, we introduce a mechanism for uncertainty quantification directly into the decision-making process, enhancing the reliability and applicability of the model across various domains where decision certainty is paramount. The model opens new possibilities for research in semi-supervised learning and Bayesian inference, promising to bridge the gap between theoretical advancements and practical applications. Future work will focus on further refining the model's performance, and extending its applications in diverse SOP settings.

Acknowledgement

The authors acknowledge the research core funding No. P2-0001, P2-0103 and P2-0383 financially supported by the Slovenian Research Agency. The authors also acknowledge the funding received from the European Union's Horizon 2020 research and innovation programme project HECAT under grant agreement No. 870702 .

References

Andonovikj, V., Boškoski, P., Džeroski, S., Boshkoska, B.M., 2024. Survival analysis as semi-supervised multi-target regression for time-to-employment prediction using oblique predictive clustering trees. Expert Systems with Applications 235, 121246. URL: http://dx.doi.org/10.1016/j.eswa.2023.121246, doi:10.1016/j.eswa.2023.121246.

Blei, D.M., Kucukelbir, A., McAuliffe, J.D., 2017. Variational inference: A review for statisticians. Journal of the American Statistical Association 112, 859–877. URL: http://dx.doi.org/10.1080/01621459.2017.1285773, doi:10.1080/01621459.2017.1285773.

Brentnall, A.R., Cuzick, J., 2016. Use of the concordance index for predictors of censored survival data. Stat Methods Med Res 27, 2359–2373. doi:10.1177/0962280216680245.

Gerds, T.A., Schumacher, M., 2006. Consistent estimation of the expected brier score in general survival models with right-censored event times. Biom. J. 48, 1029–1040. doi:10.1002/bimj.200610301.

Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H., 2009. The elements of statistical learning: data mining, inference, and prediction. volume 2. Springer.

Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J., 2013. Stochastic variational inference. Journal of Machine Learning Research .

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kocev, D., Vens, C., Struyf, J., Džeroski, S., 2013. Tree ensembles for predicting structured outputs. Pattern Recognition 46, 817–833. URL: http://dx.doi.org/10.1016/j.patcog.2012.09.023, doi:10.1016/j.patcog.2012.09.023.

Kvamme, H., Borgan, Ø., 2023. The brier score under administrative censoring: problems and a solution. Journal of Machine Learning Research 24, 1–26.

Plumb, G., Molitor, D., Talwalkar, A.S., 2018. Model agnostic supervised local explanations. Advances in neural information processing systems 31.

Sashank, J.R., Satyen, K., Sanjiv, K., 2018. On the convergence of adam and beyond, in: International conference on learning representations.

Stepisnik, T., Osojnik, A., Dzeroski, S., Kocev, D., 2020. Option predictive clustering trees for multi-target regression. Computer Science and Information Systems 17, 459–486. URL: http://dx.doi.org/10.2298/CSIS190928006S, doi:10.2298/csis190928006s.

Stepišnik, T., Kocev, D., 2021. Oblique predictive clustering trees. Knowledge-Based Systems 227, 107228. URL: http://dx.doi.org/10.1016/j.knosys.2021.107228, doi:10.1016/j.knosys.2021.107228.

Tan, S., Soloviev, M., Hooker, G., Wells, M.T., 2020. Tree space prototypes: Another look at making tree ensembles interpretable, in: Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference, ACM. URL: http://dx.doi.org/10.1145/3412815.3416893, doi:10.1145/3412815.3416893.

Therneau, T.M., 2017. A Package for Survival Analysis in R. URL: https://CRAN.R-project.org/package=survival.

Zhao, X., Wu, Y., Lee, D.L., Cui, W., 2019. iforest: Interpreting random forests via visual analytics. IEEE Transactions on Visualization and Computer Graphics 25, 407–416. URL: http://dx.doi.org/10.1109/TVCG.2018.2864475, doi:10.1109/tvcg.2018.2864475.