PEMBOT: Pareto-Ensembled Multi-task Boosted Trees

Gokul Swamy*
swagokul@amazon.com
Amazon
Seattle, WA, USA

Anoop Saladi*
saladias@amazon.com
Amazon
Bangalore, KA, India

Arunita Das*
arunita@amazon.com
Amazon
Bangalore, KA, India

Shobhit Niranjan
shobhnir@amazon.com
Amazon
Bangalore, KA, India

ABSTRACT

Multi-task problems frequently arise in machine learning when there are multiple target variables, which share a common synergy while being sufficiently different that optimizing on any of the task does not necessarily imply an optimum for the others. In this work, we develop PEMBOT, a novel Pareto-based multi-task classification framework using a gradient boosted tree architecture. The proposed methodology involves a) generating multiple instances of Pareto optimal trees, b) diverse subset selection using a determinantal point process (DPP) model, and c) ensembling of diverse Pareto optimal trees to yield the final output. We tested our framework on a problem from an e-commerce domain wherein the task is to predict at order placement time the different adverse scenarios in the order shipment journey such as the package getting lost or damaged during shipment. This model enables us to take preemptive measures to prevent these scenarios from happening resulting in significant operational cost savings. Further, to show the generality of our approach, we demonstrate the performance of our algorithm on a publicly available wine quality prediction dataset and compare against state-of-the-art baselines.

KEYWORDS

Pareto Optimal, Multi-task, Gradient Boosting, Detrimental Point Process

ACM Reference Format:
Gokul Swamy, Anoop Saladi, Arunita Das, and Shobhit Niranjan. 2024. PEMBOT: Pareto-Ensembled Multi-task Boosted Trees. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3637528.3671619

1 INTRODUCTION

In various fields, tackling multiple objectives simultaneously is a common challenge. Multi-task models are instrumental in this context, as they empower simultaneous prediction and management of several interrelated tasks. These models capitalize on the shared information and underlying patterns across different tasks, enabling more robust and generalizable solutions. By addressing multiple objectives concurrently, multi-task models optimize performance, and enhance decision-making. Consequently, this approach has gained significant attention and is widely used in real-world applications, where diverse variables and outcomes need to be considered collectively.

For example, consider e-commerce operations, where the logistics network handles millions of orders. While a large majority of these orders are successfully fulfilled, a small fraction of these orders can result in a refund claim due to various types of defects (lost packages, incorrect item being shipped or returned, damaged package, etc.) in the forward leg or reverse leg of the fulfilment cycle. While each of these different defect types have significant amount of synergistic information owing to overlapping elements in the supply-chain, they require different business levers (e.g. secure packaging, attended delivery) to prevent them. Thus, it is imperative to be able to predict which order is likely to result in a particular type of defect and dynamically assign the appropriate business lever to minimise operational losses. Training and maintaining a separate prediction model for each type of defect is resource intensive and places enormous burden on technical resources for model maintenance and troubleshooting. This disjointed approach also does not take into account the synergies between the various defect types. Our proposed work mitigates this limitation through deployment of a single unified model for multi-task prediction that is capable of exploiting label synergies to bring about improvements in the predictive performance.

The various elements of an e-commerce supply chain can often be characterized by a set of categorical and continuous features thus rendering the problem to be one of tabular data prediction. Gradient boosted trees have been the method of choice for predictions on tabular data owing to their superior performance and computational efficiency [17]. In this study, we propose a novel **P**areto **E**nsembled **M**ulti-task **BO**osted **T**ree (PEMBOT) architecture for predictions on tabular data. Specifically, the major contributions of our work are as follows:

- We propose a novel multi-task prediction architecture parameterized by a gradient boosted tree to generate solutions along the Pareto front.

*First three authors contributed equally to this research.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671619

- Subset selection scheme based on a determinantal point process algorithm to select the most diverse solutions along the Pareto front, and use them in the ensemble for real-time inferencing without the need to specify any task specific weights.
- Extensive experimentation on a real-world and a public dataset to demonstrate superiority of the proposed approach compared to state-of-the-art baselines. By leveraging multi-task synergies, we deliver significant performance gains especially for the minority tasks with limited training data and/or severe class imbalance.

In the next section, we start with introducing the related work in the tabular prediction domain and provide a detailed description of the proposed methodology in following sections.

2 RELATED WORK

Multi-task (or multi-objective) models primarily arise when there are multiple target variables, which share a common synergy while being sufficiently different such that optimizing on any of the task does not necessarily imply an optimum for the others. In fact, very often, optimizing for a particular task degrades the performance of the other tasks and therefore it becomes impossible to define a single global optimum that is optimal for all the tasks. The generic approach for optimizing these models is to employ linear scalaraization wherein the multiple objectives are combined through a predetermined weighting scheme [23]. Selection of weights however is very subjective and may have a large influence on the final performance. Despite these challenges, a multi-task setup has the potential to significantly improve the generalization performance by leveraging the domain-specific information contained in all the related tasks [5]. This aspect has been exploited by researchers to design frameworks wherein a part of the network is common between all the tasks to learn the synergistic information. A neural network based model consisting of shared hidden layers followed by a task specific layer was first reported in [4]. It was shown in [1] that the risk of overfitting the parameters in the shared layer of this network was N times smaller than that in the task specific layers (where N is the number of tasks) and this is what led to a superior generalization performance of the multi-task algorithm. A variation of this concept does away with the concept of shared layers, but instead trains a different neural network for each task while ensuring that the parameters of the trained models are as similar to each other as possible [14, 32]. These neural networks can be trained using alternate sampling (or any other sampling strategy) or loss weighting of the training examples to represent each of the tasks. Deep learning approaches have also been extended to multi-task learning wherein for example Deblina et. Al., [2] proposed a transformer based architecture for learning a shared representation across the task and Duan et. Al., [13] proposed a CNN architecture for addressing the multi-task problem in the robotics domain. These approaches are however more tuned towards vision tasks and do not readily extend to other domains. These methods works well when the tasks are non-competing which is usually never the case. In practice, a multi-task problem is often also a multi-objective problem where the different tasks compete with each other [29]. More recently, a gated additive tree ensemble [21] method was proposed for multi-task prediction that uses a gating mechanism along with a non-linear decision tree ensemble to achieve significant performance gains over multiple SOTA baselines.

Predictions based on tabular input features continue to hold significant importance for multiple industrial domains wherein gradient boosted tree models have been the method of choice [17]. Advances in deep learning methodologies attempt to bridge this gap by either using a transformer based dense representation of categorical features as postulated in TabTransfomer [20] or attempting to mimic a tree based feature selection as in TabNet[30]. More recently, several modifications have been proposed to the transformer architecture with FT-Transformer [16] employing a feature tokenizer prior to the transfomer layer and SAINT [31] employing row and column attention to encode the tabular features. These approaches can also be extended for multi-task learning using either linear scalarization or simply training them for each individual task.

Multi-objective problems are often characterized using a Pareto front wherein the solutions along the Pareto front represent different tradeoffs between the multiple objectives and by definition it is not possible to find a different optimum wherein the performance on all the tasks is improved simultaneously (such a solution is said to be un-dominated). There have been a slew of methods involving both gradient-based and gradient-free optimization to approximate solutions along the Pareto front. In the gradient-free approach, evolutionary algorithms have been the method of choice wherein parameterizations that lead to undominated solutions are propagated over others [15, 19]. This approach has also been extended to evolve multi-objective decision trees [10, 33]. Evolutionary approaches however tend to quickly break down with scale and are not feasible but for the simplest of problems [6]. Gradient based approaches involve finding a descent direction at each iteration such that the performance on all the objectives is simultaneously improved with the iterations continuing until a feasible solution exists [11, 28]. Gradient based methods are much more resilient to scale, but may end up generating solutions that bunch together in a narrow zone of the Pareto front [25]. More recently, a multi-task learning framework was proposed wherein the authors decomposed the original problem into multiple sub-problems with each sub-problem characterizing a different trade-off between the multiple objectives [25]. The way this was done was by constraining the parameter vector to lie within a certain zone and then solving the ensuing constrained optimization problem. This method however cannot be extended towards optimizing multi-objective decision trees as it is not possible to create an explicit parameterization of a decision tree apriori. The proposed PEMBOT architecture overcomes several of the limitations stated above and achieves state-of-the-art performance for multi-task tabular data prediction problems.

The rest of the paper is structured as follows. In section 3, we briefly touch upon Pareto optimality and discuss the conditions under which a solution is Pareto stationary. In section 4.1, we describe the multi-objective GBT framework in detail and elaborate on a scheme to sample solutions from the Pareto frontier. In section 4.2, we outline a novel determinantal point process (DPP) based algorithm to select a subset of the most diverse Pareto-optimal GBT

models. In section 5, we elaborate on a real-world e-commerce problem pertaining to supply chain loss prevention and demonstrate the application of the proposed methodology to drive incremental operational cost savings. In section 6, we test the approach on a public wine-quality dataset and compare the performance against multiple state-of-the-art baselines. Finally, some conclusions and future work is outlined in section 7.

3 PRIMER ON PARETO OPTIMALITY

Consider a multi-task binary classification task with M objectives defined by minimization problem:

$$\min_{\theta} F(\theta) = \min_{\theta}\{F_1(\theta), F_2(\theta), ..., F_M(\theta)\} \tag{1}$$

where θ is the parametric representation of an instance of any classification model (neural network, GBT, etc) and $F_i(\theta)$ is the binary cross-entropy loss for the i^{th} task given by:

$$F_i((\mathbf{x}, \mathbf{y}); \theta) = \sum_{j=1}^{N} f_i((x_j, y_j); \theta) \tag{2}$$

$$= \sum_{j=1}^{N} y_j log(p_j(\theta)) + (1 - y_j) * log(1 - p_j(\theta)); y_j \in [0, 1]$$

where (x_j, y_j) is a training data pair from the i^{th} task instance, N is the number of training samples for the i^{th} task and $p_j(\theta)$ is the probability that $y_j = 1$ given the model θ. For notational simplicity we avoid an explicit task specific index for the training samples and instead assume that the training samples correspond to the loss function that is being referenced.

Eq 1 can be optimized by finding a descent direction for the parameter θ, at each iteration, such that each of the objectives is simultaneously minimized *i.e*

$$F_i(\theta^{k+1}) \leq F_i(\theta^k), \ \forall \ i \in [1, M] \tag{3}$$

Here we have omitted an explicit reference to the training data pair (x, y) for the i^{th} task from the expression for F_i for convenience. More formally, let $q(\theta)$, represent a unit vector along a descent direction satisfying eq 3. $q(\theta)$ must then satisfy:

$$-\nabla(F_i(\theta^k)).q(\theta^k) \geq 0, \forall \ i \in [1, M] \tag{4}$$

One way to find the descent direction satisfying eq 4 is to solve the following quadratic optimization problem (see [28] for details):

$$min_{\alpha}\{\left\|\sum_{i=1}^{M} \alpha_i * \nabla(F_i(\theta^k))\right\|_2^2\} \quad s.t. \ \ \alpha_i \geq 0 \mid_{i=1..M} \sum_{i=1}^{M} \alpha_i = 1 \tag{5}$$

Eq 5 can be solved using a constrained quadratic optimization program with the descent direction satisfying eq 4 being inferred from the optimal α's (α^{opt}) as [28]:

$$q(\theta^k) = -\sum_{i=1}^{M} \alpha_i^{opt} * \nabla(F_i(\theta^k)) \tag{6}$$

The absolute minimum for eq 5 is obtained when the convex hull formed by the gradients contains the zero vector. In such a scenario, $q(\theta^k) = 0$ and there is no descent direction for which eq 4 is satisfied (for every other case $q(\theta^k)$ yields a valid descent direction). Such a point is said to be Pareto stationary wherein it is not possible to simultaneously minimize the value of all of the objectives without degrading at least one. The descent direction obtained by solving eq 5 is by no means unique and there are several other descent directions that are likely to satisfy eq 4. A procedure to obtain a set of such feasible descent directions is elaborated in [3]. Each descent direction, will however eventually terminate at a Pareto stationary point. A complete set of Pareto stationary points defines the Pareto frontier for the multi-task problem.

In the next section, we make use of the Pareto stationarity property to detail the construction of a novel multi-task learning framework using gradient boosted decision trees wherein each task is an instance of a binary classification problem.

4 PROPOSED APPROACH

4.1 Pareto based multi-class gradient boosted trees

Gradient boosted trees (GBT) are a popular choice for classification problems, especially when the feature set consists of many categorical variables. GBT involves fitting tree stumps as weak learners (see [26]):

$$S_m(x) = S_{m-1}(x) + \gamma * h_m(x) \quad for \ m = 1....n_{boost} \tag{7}$$

where $h_m(x)$ is a decision tree weak learner and $S_0(x)$ is some chosen initialization. $h_m(x)$ can be obtained by fitting a decision tree weak learner (θ) to the pseudo-residual $r_{jm}(x_j)$ computed from the loss function f as follows:

$$r_{jm}(x_j) = -\frac{\partial f(y_j, S_{m-1}(x_j))}{\partial S_{m-1}(x_j)} \ \ \& \ \ h_m(x) = \theta(r_{jm}(x), j) \mid_{j=1...N} \tag{8}$$

Eq 7 and 8 are akin to a gradient descent procedure with the iterations terminating when the gradient of the loss function approaches zero *i.e*.

$$\nabla F = \sum_{j=1}^{N} \nabla f(y_j, x_j) \approx 0 \tag{9}$$

In the case of a multi-task framework, we can define a scalarization loss function L, which lies in the convex hull of the individual loss functions, as (see [12] for more details):

$$F = \sum_{i=1}^{M} \alpha_i * F_i((\mathbf{x}, \mathbf{y}); \theta), \alpha_i \geq 0 \mid_{i=1...M} \ \ \sum_{i=1}^{M} \alpha_i = 1 \tag{10}$$

When a GBT model is trained on the entire multi-task data with the loss being computed as in 10, then at convergence the gradient of the loss should ideally approach zero:

$$\nabla F = \sum_{i=1}^{M} \alpha_i * \nabla F_i \approx 0 \tag{11}$$

Eq 11 happens to satisfy the very criteria for Pareto stationarity as outlined in Eq 5 and thereafter. Therefore, the output $S_{n_{boost}}(x)$, which is obtained after n_{boost} rounds of boosting is an instance of a locally Pareto stationary solution if the gradient upon convergence is less than a certain threshold ($\nabla F \leq \epsilon$). Our method in-part is inspired by a multi-task neural network, comprising of a shared hidden layer and a task specific final layer. To mimic this aspect in a boosted decision tree, we continue boosting the Pareto stationary solution $S_{n_{boost}}(x)$ with $n_{additional}$ rounds but with the task specific loss function $F_i(\mathbf{x}, \mathbf{y}), \forall i \in [1, M]$, to yield a final prediction output (T^i) for each of the individual tasks. The initial portions of the gradient boosted tree encode synergistic information from the various tasks, while the later portions extend this to learn a more task specific nuance.

The proposed methodology overcomes the limitation of having to specify task weights to learn the joint representation, but the output is likely to be significantly influenced by the point of convergence of the Pareto stationary solution along the Pareto front. For instance, as in the previous section, choosing the min-norm combination of α's in the convex-hull leads to one such realization. In order to overcome this, we sample several sets ($N_\alpha \sim 100$) of α's satisfying the condition laid out in eq 10 and build gradient boosted trees using the procedure described above to yield multiple instances of Pareto stationary solutions which are likely representative of the entire Pareto front. The task specific predictions, T^i_j, obtained by extending the Pareto stationary solutions, are then ensembled to yield the final prediction for each task:

$$pred^i = \sum_{j=1}^{N_\alpha} T^i_j \mid_{i=1\ldots M} \tag{12}$$

Here the super-script i ($i \in [1, M]$), indexes the task and the sub-script j ($j \in [1, N_\alpha]$), indexes the pareto-stationary solutions. T^i_j is obtained by boosting the j^{th} pareto stationary solution with the task specific extension for the i^{th} task. It must be noted that it is essential to have a large enough N_α so as to obtain solutions that are representative of the Pareto front. While this is feasible for offline predictions, it is computationally intractable to perform inference for such large number of models for real-time applications. In order to solve this bottleneck, we make an assumption that solutions that are far apart along the Pareto front are likely to yield boosted decision trees that are more diverse as opposed to solutions that are close together. To make the methodology computationally tractable, we select a much smaller subset (~10) from the entire set of Pareto stationary solutions by ensuring that the selected subset is maximally diverse. This is achieved using a novel determinantal point process (DPP) based subset selection methodology for decision trees, which is described in detail in the next sub-section.

4.2 Diverse Model Subset Selection

Here, we describe our methodology based on Determinantal Point Processes (DPPs) to select a subset of k-diverse models from a set of N_α GBT models. We first create a vector representation of each of the trained GBT models by utilizing the inherent tree structure. This approach entails capturing the decision paths traversed by data points within the trees into embeddings. Following this, our DPP based algorithm then picks a diverse subset of models by incrementally maximizing the volume of the parallelepiped spanned by the GBT model embeddings. Maximizing the volume translates to maximizing the determinant of the matrix $det(G^T G)$ since this represents the squared volume of feature vectors associated with selected GBT models $G = (g_1, \ldots, g_k)$. Selecting k diverse models from an overall list of N_α models can be formulated as a DPP problem when we have the $N_\alpha \times N_\alpha$ similarity matrix.

4.2.1 GBT Model Embeddings. In our work, we explored 3 different vectorization schemes to create GBT model representations based on features picked up by individual boosted trees within a booster object – (i) Use one-hot encodings of independent features (ii) Use natural logarithm of information gain associated with the features (iii) Use natural logarithm of coverage associated with the independent features. To illustrate, let's say we train a GBT model with t iterations. In each of the iterations we build a decision tree using a subset of all the available features in the dataset and thus, we have t individual decision trees in the given booster object. We create one-hot encoding of features used by decision tree at each iteration. Assuming the number of features in the training dataset as f, we get a vector with binary values having a shape of $[1 \times f]$ associated with each decision tree. By concatenating the vectors of all the decision trees, we create an overall representation for a booster object with shape of $[1 \times t * f]$. Similarly, instead of using a one-hot representation, we use natural logarithm of information gain/coverage associated with independent features to create vector representation of each decision tree which can then be concatenated in a similar fashion. Based on empirical performance, we selected method (ii) based on logarithm of information gain associated with features for our experimentation. Additionally, in order to reduce the dimensionality and create a more dense representation, we used Singular Value Decomposition (SVD) [24] technique. By applying SVD, we create a compact and dense d-dimensional representation for each GBT model. The distance between any two GBT models can then be computed by taking a cosine similarity/euclidean distance between their vector representations.

4.2.2 DPP Overview. A point process $\mathbf{P}$ on a set $\mathbf{S} = \{1, 2.., N\}$ is a probability distribution on the powerset of $\mathbf{S}$. That is, $\forall S \subseteq \mathbf{S}$, $\mathbf{P}$ assigns some probability $P(S)$, and $\sum_{S \subseteq \mathbf{S}} P(S) = 1$. DPPs represent a family of such probability distributions whose parameters can be tuned such that the probability of a subset, $P(S)$ is proportional to a combined measure of quality and diversity of these items. Although DPP is able to assign probability to subsets of all size, we are interested in selecting a subset with fixed size k. In this case, the DPP can be specialized to k-DPP where $P(X = K) \propto det(S_K)$ if $|K| = k$ and 0 otherwise. Thus, the more diverse the set k, the higher the volume of the parallelepiped of the represented items and consequently the higher its probability of being sampled. Although there are other reasonable methods such as Maximal Marginal Relevance (MMR) to take diversity into account, we chose DPP because of two reasons – (i) Basis the literature survey, DPPs tend to outperform other methods [7] (ii) DPPs are probabilistic in nature and therefore we can take advantages of algorithms for probabilistic operations.

Table 1: Notations used in section 4.2.3 along with descriptions

Notation	Description
N_α	Total number of models trained basis methodology described in above section
J	Subset of indices corresponding to a subset of models
I	Identity matrix
L	Matrix of latent vector representations of all the N_α models
L_J	Submatrix of L having vector representations of models restricted to indices in J
S	Similarity matrix of models obtained as $L^T L$ where $S_{ij} =< L_i.L_j >$
q_j	Quality score of j-th GBT model generated using out-of-sample F1-scores

4.2.3 Greedy-DPP algorithm for subset selection. We use the notations as indicated in Table 1 for further discussions. Basis what we have mentioned in the DPP overview section, we can formulate our objective of finding k diverse models as set out in the equation below with $|J| = k$

$$\max_{J:|J|=k} \sum_{j \in J} q_j + \lambda \left(\log\det(S_{JJ} + \eta I)\right) \tag{13}$$

In Eq 13, the first term represents the quality score of the Pareto optimal GBT model and promotes the selection of best performing models. The second term in the objective prefers diverse models to be picked up. Here, λ is a trade-off parameter that balances between the twin objective of performance and diversity. η is a pre-specified constant that keeps the objective function well defined and numerically stable. The log-determinant is undefined if the input matrix is singular (i.e., non-invertible) and hence, adding the scaled identity matrix ηI ensures that the resulting matrix is always positive definite (and hence, invertible), thereby avoiding the log-determinant from being undefined. In practice, the constant η is typically chosen to be a small positive value to ensure numerical stability while minimally perturbing the original matrix. It can be seen as an instance of Tikhonov regularization for solving ill-posed problems. S is a similarity matrix of dimension $N_\alpha \times N_\alpha$ and S_{JJ} is of dimension $k \times k$.

We use a greedy algorithm (Greedy DPP) to solve for the above objective. The algorithm adds an index to a running index set in each iteration to solve for the reformulated objective where we replace S with matrix structure $L_j L_j^T$ in Eq 14. It has a $(1 - 1/e)$ approximation guarantee with respect to the objective in Eq 13. This greedy procedure can be viewed as seeking the mode of the DPP. While this combinatorial problem is hard to solve, since the objective is submodular and monotone, Nemhauser's result applies to the objective [7, 27].

$$\max_{J:|J|=k} \left(\sum_{j \in J} q_j + \lambda \log\det(\sum_{j \in J} L_j L_j^T + \eta I) \right) \tag{14}$$

Algorithm 1 Greedy Determinental Point Process (Greedy DPP) based Algorithm

function GDPP(L, q, k, λ, η) $J\{\max_j q_j\}$
  while $|J| \leq k$ **do** $JJ \cup \{j^*\}$ where
    $j^* \leftarrow \arg\max_{j \notin J} \left(q_j + \lambda \log\det(L_J L_J^T + L_j L_j^T + \eta I\right)$ (15)

5 E-COMMERCE LOSS PREVENTION

5.1 Problem Scope

E-commerce operations typically involve shipping millions of orders through an intricate network of logistic hubs to reach the end-customer. While a large majority of these orders are successfully fulfilled a tiny fraction could result in operational losses stemming from either supply-chain defects or fraudulent claims. We consider three main types of operational losses categorized into a) customer claims of having never received the package, b) claims arising from a damaged or materially different product being delivered and c) losses from incorrect / fake items being returned. Predicting the loss propensity for each order and for each loss-type enables remedial actions to be triggered to minimize the occurrence of such losses. The process schematic for loss-intervention is presented in figure 1 wherein we do not disclose the nature of interventions to preserve confidentiality. Given that the supply-chain elements are common between these tasks and that fraudulent entities typically exploit all available avenues to commit fraud, we expect the task labels to share a high-degree of synergistic information.

Given the scale of e-commerce operations, any improvement in loss prediction can have a significant impact on the overall operational cost savings. In following sub-sections, we detail the experimental evaluation of PEMBOT and the business impact stemming from implementing this framework for loss minimization on a large e-commerce domain.

5.2 Dataset & Baselines

We considered all e-commerce orders placed in a given month and tagged each order with a one-hot vector of binary class labels indicative of whether the indexed aforementioned loss-type was encountered for the given order. To represent these operational loss-types we use the generic nomenclature of *op_loss*1, *op_loss*2 and *op_loss*3 without specifying which index corresponds to which loss-type in order to preserve confidentiality of business critical information. To facilitate prediction, we curate a feature set for each order comprising of 187 features spanning across product,logistics and customer level historical aggregates. In practise, these losses are rare and the overall incidence rate of these losses (combined) is less than 5% constituting a class imbalanced dataset for each of the tasks. In order to mitigate the class imbalance in the training

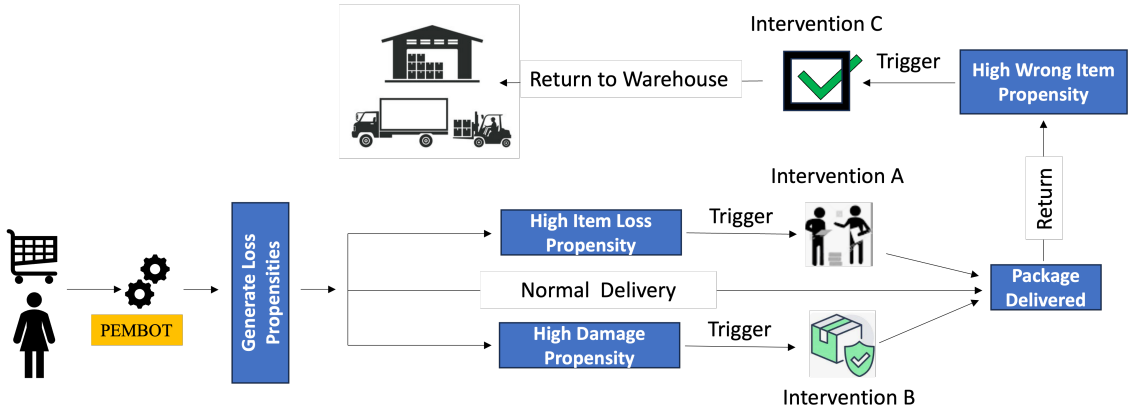

Figure 1: Loss Prevention Process Flow

data we performed a stratified sampling comprising of all orders (from a given time period) which belonged to either of the loss types and augmented this data with a 10% sample of orders that were successfully delivered (no losses). Test data comprised of all orders from a future time period.

To test the efficacy of the proposed approach on the multi-task prediction problem, we compare against the hugely popular gradient boosted tree benchmarks of XGBoost [8] and LightGBM [22] and also a more recent multi-task tree based ensemble approach GATE [21]. In addition, we also did a comparative analysis against state-of-the-art deep-learning architectures for tabular data learning including TabTransformer [20], TabNet [30], SAINT [31], FT-Transformer [16] and ResNet based tabular data architecture [16]. The hyper-parameters of all baseline models were tuned basis a simple grid search.

5.3 Experimental Setup

We implemented PEMBOT framework atop the XGBoost library. For training multiple instances of pareto optimal boosted trees in the first stage (~100 models per task), we adopt a stochastic approach for the max depth parameter by varying it randomly between 7 to 15. We use binary logistic function as the training objective with 0.07 learning rate. We generate 128 dimensional embeddings for each of the trained GBT models based on the methodology detailed in 4.2.1. We reduce the number of models used in the final ensemble to 10 through DPP based subset selection. As a general principle we have observed that setting N_α to $\sim 30 \times N_{\text{tasks}}$ and the number of DPP selected models to $\sim$ 10% of N_α yields an optimal operating point. The hyper parameters for model training are selected basis grid-search across multiple runs. To ensure the reliability of our results, we repeat all the experiments 10 times and report the average value across the runs.

5.4 Results

To evaluate the performance of multiple approaches and measure the quality of predictions, we looked at standard classifier evaluation metrics such as precision, recall, area under the precision recall curve (AUC-PR) and area under the curve (AUC-ROC). The AUC-PR for PEMBOT and all other baselines, for each loss-type, is summarized in Table 9 with AUC-ROC being presented in Table 3.

From an operational standpoint, our objective is to identify the top 5% of risky orders and apply necessary interventions to ensure a successful delivery. Table 4 summarizes the model recall for all the models when the top 5% of the risky orders (from each model) are considered. Note that we are more interested in identifying the total order value at risk instead of just the number of risky orders, so we report value recall in the table. Following from the AUC-PR metrics, we observe a significant (~10%) improvement in task-2 recall with the PEMBOT model as opposed to all other baselines. Figure 2 depicts the precision / recall curve for each of the tasks wherein we observe that PEMBOT achieves a superior performance compared to baseline models at any chosen operating point.

The largest recall gain (table 4) of ~30% is realized for op_loss2 which had the lowest label incidence rate among all the loss-types. We postulate that the shared tree construction with the loss specified in eq 10 enables the minority task to learn synergistic patterns from other tasks which help augment the label information present in the minority task leading to significant predictive performance gains.

5.4.1 Ablation Study. We performed an ablation study on the validation set, using recall metric as performance criterion. We addressed few specific questions with this study - (i) Does the approach based on diverse model subset selection using greedy DPP algorithm result in better performance than using random k models? (ii) How sensitive is the performance of our approach to the value of k during inference stage?

Results presented in Table 5 confirm that using a DPP model for diverse subset selection on Pareto based multi-class gradient boosted trees leads to a better performance than random selection of models. Table 6 demonstrates that the value recall increases as the value of k increases till 8 and then the performance almost flattens out.

Table 2: AUC PR for Proposed approach and Baselines

Label	PEMBOT	XGBoost	LightGBM	ResNet	SAINT	TabNet	GATE	TabTransformer	FT-Transformer
op_loss1	**16.21%**	13.16%	10.30%	11.79%	14.46%	9.96%	11.07%	7.72%	12.69%
op_loss2	**17.25%**	14.11%	12.50%	12.56%	14.71%	12.16%	12.10%	10.03%	13.56%
op_loss3	**35.58%**	32.86%	31.70%	31.81%	33.52%	31.39%	31.59%	24.85%	32.24%

Table 3: Model AUC-ROC for Proposed approach and Baselines

Label	PEMBOT	XGBoost	LightGBM	ResNet	SAINT	TabNet	GATE	TabTransformer	FT-Transformer
op_loss1	**82.18%**	80.17%	79.00%	79.24%	80.41%	75.68%	77.56%	74.28%	78.31%
op_loss2	**93.28%**	92.68%	92.00%	91.69%	92.65%	91.73%	91.75%	90.89%	92.43%
op_loss3	**93.10%**	92.52%	91.50%	92.03%	92.22%	91.47%	91.68%	90.05%	92.29%

Table 4: Value Recall for Top 5% risky orders as flagged by various models

Label	PEMBOT	XGBoost	LightGBM	ResNet	SAINT	TabNet	GATE	TabTransformer	FT-Transformer
op_loss1	**23.30%**	20.12%	20.52%	19.83%	20.12%	20.67%	22.47%	18.97%	19.88%
op_loss2	**40.00%**	31.30%	29.05%	25.99%	21.45%	37.31%	20.54%	33.24%	24.84%
op_loss3	**57.75%**	55.52%	37.73%	23.88%	49.02%	53.13%	20.75%	41.52%	29.66%

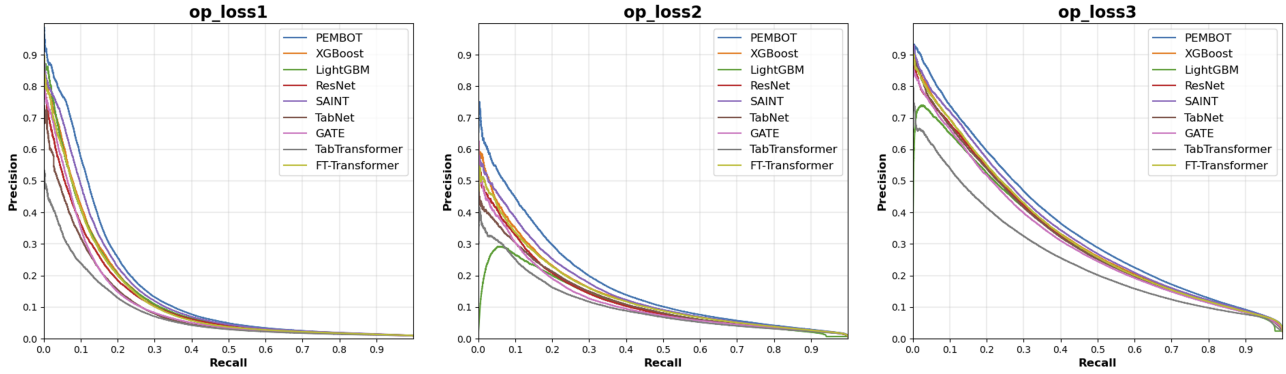

Figure 2: Precision Recall Plot for E-Commerce dataset

5.5 Implementation & Overall Impact

To enable real-time inferencing with the PEMBOT model, we distribute the pareto ensemble computation over multiple cores to allow for fast parallel computation. We observed the following average inference time per order: PEMBOT - 0.7 *ms*, XGBoost - 0.02 *ms*, LightGBM - 0.01 *ms*, SAINT - 17.6 *ms*, TABNET - 56.8 *ms*, TabTransformer - 21.2 *ms* and FT-Transformer - 21.31 *ms*. XGBoost, LightGBM and PEMBOT inference was performed on a 64-core vCPU machine while we used the NVIDIA T4 GPU instance, with 4-GPU cores and 192 GB memory, for inference with the deep-learning architectures. Although, PEMBOT is slower than XGBoost, it is still much faster than other approaches and < 1 ms latency allows for scaling the model for high throughput real-time computations.

PEMBOT has been deployed in production for an emerging marketplace and is invoked for every order that is placed to predict the loss propensities and trigger appropriate interventions. By replacing the legacy gradient boosted tree framework, PEMBOT was able to prevent additional operational losses to the tune of 5 basis points of revenue resulting in several millions of dollars of overall cost savings.

6 WINE QUALITY PREDICTION

We also evaluated PEMBOT on a publicly available wine quality dataset [9]. While multi-task challenges with synergistic task labels are common in e-commerce and associated fields, tabular datasets that capture such complexities, are seldom found in the public domain. The wine-quality prediction dataset from the UCI Machine Learning Repository provides the closest approximation to

Table 5: Comparison of value recall of top 5% risky orders for diverse subset models vs random k models with $k = 10$

Label	op_loss1	op_loss2	op_loss3
Random k GBT models with varying hyper-parameters	1.16%	32.66%	56.10%
Random k Pareto-optimal models	22.32%	37.49%	57.14%
Diverse subset models from greedy k-DPP	23.30%	40.00%	57.75%

Table 6: Comparison of recall of top 5% risky orders for different values of k in inference stage

Label	1	2	3	4	5	6	7	8	9	10
op_loss1	20.95%	22.42%	22.68%	22.71%	22.78%	22.97%	23.14%	23.26%	23.28%	23.30%
op_loss2	32.60%	37.25%	38.07%	38.18%	38.39%	39.01%	39.52%	39.84%	39.95%	40.00%
op_loss3	55.85%	57.03%	57.25%	57.27%	57.33%	57.48%	57.62%	57.74%	57.75%	57.75%

Table 7: Wine Quality dataset
Basic Statistics of Training and Evaluation data-sets

	Training		Evaluation	
Statistic	Value	Incidence %	Value	Incidence %
# Observations	5,193	-	1,299	-
Quality 3	24	0.46%	6	0.46%
Quality 4	173	3.33%	43	3.31%
Quality 5	1,710	32.93%	428	32.95%
Quality 6	2,269	43.69%	567	43.65%
Quality 7	863	16.62%	216	16.63%
Quality 8	154	2.97%	39	3.00%

Table 8: AUC PR for Wine Quality Prediction

Label	PEMBOT	XGBoost	LightGBM	ResNet	SAINT	TabNet	GATE	TabTransformer	FT-Transformer
AUC PR Quality 3	10.01%	8.05%	8.48%	0.37%	3.30%	3.08%	2.72%	6.48%	**13.66%**
AUC PR Quality 4	**28.45%**	24.49%	25.56%	16.28%	15.41%	4.98%	14.61%	15.38%	22.38%
AUC PR Quality 5	**76.56%**	75.67%	74.81%	63.70%	62.53%	56.07%	64.74%	57.79%	65.34%
AUC PR Quality 6	**75.61%**	73.24%	73.35%	61.78%	57.60%	55.89%	59.30%	59.14%	61.12%
AUC PR Quality 7	**66.66%**	62.27%	64.30%	42.82%	44.24%	35.60%	44.50%	35.19%	42.89%
AUC PR Quality 8	**48.81%**	40.30%	45.15%	11.28%	18.19%	4.32%	11.51%	8.64%	15.64%

this scenario, allowing prediction tasks to exploit the interplay between various wine qualities basis 11 physio-chemical input features. While the repository contains two datasets for red and white wines, for our analysis, we have created a combined dataset to increase the sample size. The target variable wine-quality takes 7 distinct values-3,4,5,6,7,8,9. Quality grade 9 has only 5 samples in the entire data and no meaningful train-test split can be obtained for this class. Hence, we drop this class without forfeiting the purpose of this analysis. The dataset consisting of 6,492 observations is divided into training (80%) and evaluation (20%) set and data distribution of training and evaluation data is depicted in table 7.

6.1 Results

We compare the model performances in table 8 and note that PEMBOT consistently outperforms other strong baselines across all tasks barring task 3, wherein FT-transformer achieves a higher AUC-PR. In line with the e-commerce example, we once again observe larger improvements for minority classes of Quality 3, 4 and 8 wherein PEMBOT is able to harness label synergies to drive performance lifts.

7 CONCLUSION

In this study, we have proposed a novel Pareto-based multi-task learning framework using a gradient boosted tree architecture. We presented a DPP based model for diverse model subset selection to make the inference computationally tractable. We demonstrated that the framework is general, flexible and performs well on a highly imbalanced dataset from the e-commerce domain and a publicly available wine-quality dataset. We compared the performance of PEMBOT against multiple state-of-the-art algorithms and observe

that PEMBOT is able to significantly outperform these baselines for multi-task problems with inherent label synergies. We have deployed this model in production for a large e-commerce domain with a throughput comprising of millions of orders per-day and have realized significant cost-savings from the improved predictive performance. While we have used GBT as the base-architecture, our method could potentially be adapted for any architecture such as transformer networks, CNNs and graphical models. Future work also entails extending PEMBOT towards multi-task image classification problems that is of particular significance to multiple e-commerce applications. [18]

REFERENCES

[1] Jonathan Baxter. 1997. A Bayesian/Information Theoretic Model of Learning to Learn ViaMultiple Task Sampling. *Mach. Learn.* 28, 1 (July 1997), 7–39. https://doi.org/10.1023/A:1007327622663

[2] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. 2022. MulT: An End-to-End Multitask Learning Transformer. (2022). arXiv:2205.08303 [cs.CV]

[3] P. A. N. Bosman. 2012. On Gradients and Hybrid Evolutionary Algorithms for Real-Valued Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* 16, 1 (2012), 51–69.

[4] Richard Caruana. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. (1993), 41–48.

[5] Rich Caruana. 1997. Multitask Learning. *Mach. Learn.* 28, 1 (July 1997), 41–75. https://doi.org/10.1023/A:1007379606734

[6] Huangke Chen, Ran Cheng, Jinming Wen, Haifeng Li, and Jian Weng. 2020. Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations. *Inf. Sci.* 509 (2020), 457–469.

[7] Laming Chen, Guoxin Zhang, and Hanning Zhou. 2018. Fast Greedy MAP Inference for Determinantal Point Process to Improve Recommendation Diversity. arXiv:1709.05135 [cs.IR]

[8] Tianqi Chen and Carlos Guestrin. 2016. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16* (2016). https://doi.org/10.1145/2939672.2939785

[9] P Cortez, A Cerdeira, F Almeida, T Matos, and J Reis. 2009. Modeling wine preferences by data mining from physicochemical properties. *Elsevier* 47, 4 (2009), 547–553.

[10] Marcin Czajkowski and Marek Kretowski. 2019. A Multi-Objective Evolutionary Approach to Pareto-Optimal Model Trees. *Soft Comput.* 23, 5 (March 2019), 1423–1437. https://doi.org/10.1007/s00500-018-3646-3

[11] Jean-Antoine Désidéri. 2014. Multiple-Gradient Descent Algorithm (MGDA) for Pareto-Front Identification. 34 (2014). https://hal.inria.fr/hal-01096049

[12] Madalina Drugan. 2015. Linear Scalarization for Pareto Front Identification in Stochastic Environments. (03 2015). https://doi.org/10.1007/978-3-319-15892-1_11

[13] Shengqi Duan, Guohui Tian, Zhongli Wang, Shaopeng Liu, and Chenrui Feng. 2023. A semantic robotic grasping framework based on multi-task learning in stacking scenes. *Engineering Applications of Artificial Intelligence* 121 (2023), 106059. https://doi.org/10.1016/j.engappai.2023.106059

[14] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser. (July 2015), 845–850. https://doi.org/10.3115/v1/P15-2139

[15] Hamidreza Eskandari and Christopher D. Geiger. 2008. A Fast Pareto Genetic Algorithm Approach for Solving Expensive Multiobjective Optimization Problems. *Journal of Heuristics* 14, 3 (June 2008), 203–241. https://doi.org/10.1007/s10732-007-9037-z

[16] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting Deep Learning Models for Tabular Data. (2021).

[17] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (2022).

[18] Insu Han, Prabhanjan Kambadur, Kyoungsoo Park, and Jinwoo Shin. 2017. Faster Greedy MAP Inference for Determinantal Point Processes. arXiv:1703.03389 [cs.DM]

[19] J. Horn, N. Nafpliotis, and D. E. Goldberg. 1994. A niched Pareto genetic algorithm for multiobjective optimization. (1994), 82–87 vol.1.

[20] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. 2020. TabTransformer: Tabular Data Modeling Using Contextual Embeddings. (2020). arXiv:2012.06678 [cs.LG]

[21] Manu Joseph and Harsh Raj. 2023. GATE: Gated Additive Tree Ensemble for Tabular Classification and Regression. (2023). arXiv:2207.08548 [cs.LG]

[22] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017), 3146–3154.

[23] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2017. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. (2017). http://arxiv.org/abs/1705.07115 cite arxiv:1705.07115.

[24] V. Klema and A. Laub. 1980. The singular value decomposition: Its computation and some applications. *IEEE Trans. Automat. Control* 25, 2 (1980), 164–176. https://doi.org/10.1109/TAC.1980.1102314

[25] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. 2019. Pareto Multi-Task Learning. (2019), 12060–12070. http://papers.nips.cc/paper/9374-pareto-multi-task-learning.pdf

[26] Alexey Natekin and Alois Knoll. 2013. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics* 7 (2013). https://doi.org/10.3389/fnbot.2013.00021

[27] Wolsey L.A. Fisher M.L Nemhauser, G.L. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* (1978), 265–294. https://doi.org/10.1007/BF01588971

[28] Sebastian Peitz and Michael Dellnitz. 2017. Gradient-Based Multiobjective Optimization with Uncertainties. *Studies in Computational Intelligence* (Sep 2017), 159–182. https://doi.org/10.1007/978-3-319-64063-1_7

[29] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. (2018), 527–538. http://papers.nips.cc/paper/7334-multi-task-learning-as-multi-objective-optimization.pdf

[30] O Arik Sercan and Tomas Pfister. 2019. TabNet: Attentive Interpretable Tabular Learning. *arXiv preprint arXiv:1908.07442* (2019).

[31] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. 2021. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. *arXiv preprint arXiv:2106.01342* (2021).

[32] Yongxin Yang and Timothy M. Hospedales. 2016. Trace Norm Regularised Deep Multi-Task Learning. *CoRR* abs/1606.04038 (2016). arXiv:1606.04038 http://arxiv.org/abs/1606.04038

[33] Huimin Zhao. 2007. A Multi-Objective Genetic Programming Approach to Developing Pareto Optimal Decision Trees. *Decis. Support Syst.* 43, 3 (April 2007), 809–826. https://doi.org/10.1016/j.dss.2006.12.011

Received 1 February 2024

Table 9: E-commerce Data: Std Deviation of AUC PR for Proposed approach and Baselines

Label	PEMBOT	XGBOOST	LIGHTGBM	ResNet	SAINT	TabNet	GATE	TabTransformer	FT-Transformer
op_loss1	0.451%	0.435%	0.440%	1.211%	1.231%	0.136%	1.348%	0.974%	1.257%
op_loss2	1.160%	1.151%	1.155%	1.163%	1.192%	0.144%	0.135%	1.186%	0.158%
op_loss3	0.441%	0.429%	0.445%	1.263%	1.271%	2.384%	0.055%	0.375%	1.253%