Arranged Forests: Enhancing Random Forests by Reducing Feature Overlap between Trees

Chu Luo HTTP://CLUO29.GITHUB.IO/

Yuehui Zhang* ZYH@SJTU.EDU.CN

** Corresponding Author*
School of Mathematical Sciences
Shanghai Jiao Tong University
Shanghai, 200240, China

Abstract

This paper proposes Arranged Forests which aims to improve Random Forests by reducing the feature overlap between decision trees. The mathematics involved in our work is an ”opposite” problem of the extremal set theory. We establish a dual version of the famous Erdős-Ko-Rado theorem to settle the corresponding problem. To quantify the feature overlap in a forest, we introduce two measures: pairwise and total repetition index. For trees in Arranged Forests, we design two feature distribution algorithms to construct feature sets with the lowest total repetition index and low pairwise repetition index. Based on mathematical analysis and empirical results, we show that Arranged Forests with certain parameters can achieve much lower repetition than Random Forests. Also, empirical results show that Arranged Forests can achieve the average performance of Random Forests and significantly outperform the bad models of Random Forests.

Keywords: Ensemble Learning, Random Forests, Combinatorics, Number Theory, Probability

1. Introduction

Random Forests is a widely researched machine learning approach. The concept with the term ”Random Decision Forests” was proposed by Tin Kam Ho (see Ho (1995)) in 1995, while the Random Forests algorithm was developed by Leo Breiman (see Breiman (2001)) in 2001. The Random Forests algorithm is also popular in developers to solve practical problems such as the tasks and competitions on Kaggle Inc. (2019). Although Random Forests is considered a successful classifier in practice (see Wyner et al. (2017)), it still has some weaknesses, such as overfitting. Thus, researchers have proposed several new approaches to enhance Random Forests, including XGBoost (Chen and Guestrin (2016)) and Feature-Budgeted Random Forest (Nan et al. (2015)).

Nevertheless, previous work about ensemble learning, such as Random Forests, did not deliberate feature overlap between predictors (e.g., decision trees in Random Forests). When the feature set is large, the importance of each feature often varies. If the feature set is very large, Random Forests models with randomly selected feature sets may not perform well (see Winham et al. (2013)). The predictive performance of Random Forests may be poor if decision trees heavily rely on a subset of bad or unstable features (e.g., because of software/hardware malfunctions, some features may not always work well in prediction time), although the performance may be good because of the overlap of good features (lucky good results of random feature distributions).

The mathematics involved in this problem is the extremal set theory, originated by E. Sperner in his paper (Sperner (1928)), which contains the Sperner's theorem that the largest antichain of $2^{[n]}$ is of length at most $C_{[n]}^{\lfloor \frac{n}{2} \rfloor}$. Sperner's theorem is the starting point of the extremal set theory, where one well-known result is EKR (Erdős-Ko-Rado) theorem proved by three famous mathematicians (P. Erdős, C. Ko and R. Rado **?**). This kind of problems asks for a given finite set X such that the maximal subsets of the power set 2^X of X consist of subsets of X with large intersections. To be precise, for $t \geq 1$, what is the largest size of $\mathcal{F} \subseteq 2^X$ such that $|x \bigcap y| \geq t$ for $x \neq y \in \mathcal{F}$?

In this paper, we concern the "opposite" problem of EKR theorem: namely, we search for a maximal subset of 2^X with small intersections. This problem is meaningful in mathematics as well as machine learning. However, the "dual version" of EKR theorem for this "opposite" problem is still lacking. The reason is simple: to find an intersection $|x \bigcap y|$ of two subsets x and y, one makes use of the formula $|x \bigcap y| = |x| + |y| - |x \bigcup y|$. Now, to require larger intersection $|x \bigcap y|$ amounts to smaller $|x \bigcup y|$, and to require smaller intersection $|x \bigcap y|$ amounts to larger $|x \bigcup y|$. So the "dual version" of EKR theorem is in fact more difficult and no known theory exists. We establish some basic knowledge of the "dual EKR theorem" and apply this dual theorem to machine learning.

Thus, we propose Arranged Forests which attempts to smartly distribute features into each tree with the lowest overlap. Based on this principle, any subset of features can only impose limited influence on trees in Arranged Forests. Although cannot maximise the predictive performance, Arranged Forests can achieve the average performance and avoid the bad cases caused by improper random feature combinations among trees. Note that it consumes more resources (e.g., physical memory and energy) to obtain the average predictive performance by creating a cluster of Random Forests models. Hence, Arranged Forests is an efficient machine learning algorithm in the trend of green computing against climate change.

1.1 Contributions

This paper makes several contributions:

1. In discussing the feature overlap between decision trees, we formally state the problem as an "opposite" problem of the extremal set theory. We propose two measures to quantify feature overlap: pairwise and total repetition index.

2. We propose two feature distribution algorithms to construct Arranged Forests with the lowest total repetition index and low pairwise repetition index.

3. We show several mathematical properties regarding the total repetition index and pairwise repetition index of Arranged Forests. We also give a mathematical estimation of the total repetition index for Random Forests.

4. Using a real-world dataset, we conduct experiments to empirically compare the repetition indices and predictive performance of Arranged Forests and Random Forests.

1.2 Organisation

We organise our paper as follows. In Section 2, we will give a brief background of Random Forests and our mathematical problem related to EKR theorem. Then, in Section 3, we will formulate our problem and our definition of repetition indices which measure feature overlap between trees. In Section 4, we will present our approach and analyse feature overlap of our approach compared to Random Forests. In Section 5, we will empirically compare the repetition indices and predictive performance of our approach and Random Forests using a real-world dataset. Lastly, we will discuss the design and experimental results of our approach in Section 6, with some deliberations about limitations and future directions.

2. Related Work

2.1 Random Forests

Random Forests (see Breiman (2001)) is a widely researched and used machine learning algorithm developed by Leo Breiman in 2001. The Random Forests algorithm was based on the concept ”Random Decision Forests” proposed by Tin Kam Ho (see Ho (1995)) in 1995. It can work for both classification and regression tasks. Random Forests is considered a successful classifier in practice (see Wyner et al. (2017)). It relies on the concept of ensemble learning. Among all features in a dataset, it follows the random subspace method (see Ho (1998a), Ho (1998b)) to randomly extract a number of feature subsets and to train the same number of decision tree models using each feature subset. Also, when training each model, it randomly selects a bootstrap set (see Efron (1982)) of training samples for this model, instead of using the original training samples. During prediction, the final output is decided by the votes from all the trees.

Before the construction of Random Forests, two hyper-parameters should be set by the user: the number of trees ($Ntree$); the number of features in each tree ($Mtry$). In the literature, both theoretical and empirical studies reported that suitable selection of $Ntree$ and $Mtry$ can improve the prediction performance (Probst and Boulesteix (2018), Huang and Boutros (2016), Kulkarni and Sinha (2012)).

Recently, researchers have proposed several new approaches to enhance Random Forests in different aspects. Chen and Guestrin (2016) proposed XGBoost (eXtreme Gradient Boosting) which adopts the concept of gradient tree boosting (Friedman (2001)) and regularisation. Gradient tree boosting trains tree models successively to overcome the weakness of previously trained models. The regularisation mechanism in XGBoost considers both L1 (Lasso) and L2 (Ridge) regularisation to control the model complexity. In addition, Nan et al. (2015) proposed Feature-Budgeted Random Forest to reduce feature acquisition cost (e.g., monetary costs and hardware usage for information access), which has been a widely studied area in machine learning (see Kanani and Melville (2008)). This approach runs a greedy search to find features for each tree without exceeding a given budget constraint.

However, previous work about ensemble algorithms, such as Random Forests, did not discuss feature overlap between predictors (e.g., decision trees in Random Forests).

2.2 Mathematics: Dual version of EKR theory

Extremal set theory is originated by E. Sperner in his paper (Sperner (1928)), which contains the Sperner's theorem that the largest antichain of $2^{[n]}$ is of length at most $C_{[n]}^{\lfloor\frac{n}{2}\rfloor}$. Sperner's theorem is the starting point of the extremal set theory, where one of the most famous theorem is known as EKR theorem (Erdős-Ko-Rado) proved by three famous mathematicians (P. Erdős, C. Ko and R. Rado **?**). This kind of problems asks for a given finite set X such that the maximal subsets of the power set 2^X of X consist of subsets of X with large intersections. To be precise, for $t \geq 1$, what is the largest size of $\mathcal{F} \subseteq 2^X$ such that $|x \bigcap y| \geq t$ for $x \neq y \in \mathcal{F}$?

EKR theorem has a wide range of applications in many fields, such as multi-language communication (e.g., in a group of people, everyone speaks a unique set of k languages. If there are n languages in total, how large is the maximal subgroup of people who can mutually communicate with each other), small-world network and random walk (see Bollobás et al. (2016), Carey and Godbole (2010)).

In this paper, we concern the "opposite" problem of EKR theorem: namely, we search for a maximal subset of 2^X with small intersections. This problem is meaningful in mathematics as well as machine learning. However, the "dual version" of EKR theorem for this "opposite" problem is still lacking. The reason is simple: to find an intersection $|x \bigcap y|$ of two subsets x and y, one makes use of the formula $|x \bigcap y| = |x| + |y| - |x \bigcup y|$. Now, to require larger intersection $|x \bigcap y|$ amounts to smaller $|x \bigcup y|$, and to require smaller intersection $|x \bigcap y|$ amounts to larger $|x \bigcup y|$. So the "dual version" of EKR theorem is in fact more difficult and no known theory exists. Fortunately, this difficult problem is found to be useful in machine learning. Thus, we establish some basic knowledge of the "dual EKR theorem" and apply this dual theorem to machine learning.

3. Problem Statement

Let m features and t estimators (e.g., trees in Random Forests) be the settings of machine learning tasks ($m > 0, t > 0$). Suppose there are $k \geq 2$ features for each estimator, and $m < tk$, so that the feature overlap is a practical concern. Let $m = kq + r,\ 0 \leq r < k$.

Let each feature be identified by a number, i.e., $[m] = \{1, 2, ..., m\}$. Denote by $\begin{pmatrix} [m] \\ k \end{pmatrix}$ the set of all subsets of $[m]$ with cardinality k, that is,

$$\begin{pmatrix} [m] \\ k \end{pmatrix} = \{X \subset [m] \,|\, |X| = k\}.$$

Denote by $\mathcal{F}([m], t)$ the set of all subsets of $\begin{pmatrix} [m] \\ k \end{pmatrix}$ with cardinality t, that is,

$$\mathcal{F}([m], t) = \{\mathcal{T} \subset \begin{pmatrix} [m] \\ k \end{pmatrix} \,|\, |\mathcal{T}| = t\}.$$

From these definitions, we obtain $|\begin{pmatrix} [m] \\ k \end{pmatrix}| = C_m^k,\ |\mathcal{F}([m], t)| = C_a^t$, where $a = C_m^k$.

3.1 Measuring Feature Overlap between Trees by Repetition Indices

For any two subsets $x, y \subset [m]$, denote *the pairwise repetition index of x and y* by $r(x, y) = |x \bigcap y|$.

Let $\mathcal{S}$ be a subset of $\left(\begin{array}{c} [m] \\ k \end{array} \right)$ with at least 2 elements. Define the total repetitive index of $\mathcal{S}$ by

$$R(\mathcal{S}) = \frac{1}{2} \sum_{x \neq y \in \mathcal{S}} (|x \bigcap y|) = \frac{1}{2} \sum_{x \neq y \in \mathcal{S}} r(x, y). \tag{1}$$

So the repetition ratio $r(\mathcal{S})$, or the average of repetition of any pair of subsets is defined to be

$$r(\mathcal{S}) = \frac{R(\mathcal{S})}{\frac{1}{2}(|\mathcal{S}|^2 - |\mathcal{S}|)} = \frac{1}{|\mathcal{S}|^2 - |\mathcal{S}|} \sum_{x \neq y \in \mathcal{S}} r(x, y) \tag{2}$$

According to the above setting, our aim is equivalent to the following two optimisation problems:

$$\min_{\mathcal{S} \in \mathcal{F}([m], t)} R(\mathcal{S}) \tag{3}$$

and

$$\min_{\mathcal{S} \in \mathcal{F}([m], t)} r(\mathcal{S}) \tag{4}$$

4. Arranged Forests

4.1 Overview

The goal of Arranged Forests is to reduce feature overlap between trees. Unlike Random Forests randomly distributing features into each tree, Arranged Forests follows a procedure to methodically select the combinations of features for each tree. After arranging features for each tree, Arranged Forests trains and uses decision tree models as Random Forests does.

Thus, we describe the theoretical analysis, procedures and properties of two proposed feature distribution strategies of Arranged Forests in the following subsections.

4.2 Theoretical Investigation of Feature Subsets with the Lowest Total Repetition

As stated in Section 3, given m features and t trees with k features per tree ($k \geq 2, m < tk$), we need to find one (or more, if possible) set of feature subsets for the two goals: $\min_{\mathcal{S} \in \mathcal{F}([m], t)} R(\mathcal{S})$, and $\min_{\mathcal{S} \in \mathcal{F}([m], t)} r(\mathcal{S})$.

Before starting the search, we ask:

Question: What do(es) the intended set(s) look like?

A usual conjecture is that, in the intended set(s), the total appearances of every feature should be equal among all feature subsets, if $tk \ mod \ m = 0$. If tk is not divisible by m, their total appearances should be almost equal: at most differ by one.

Table 1: Table of Feature Appearances and Repetitions

Feature Number	Appearance	Repetition
1	a_1	R_1
2	a_2	R_2
$\cdots$	$\cdots$	$\cdots$
m	a_m	R_m

To investigate this, we create a table to count feature appearances and repetitions, as illustrated in Table 1.

Based on this table, we give a lemma:

Lemma 1. *Let $a_1, a_2, \cdots, a_m$ be the feature appearance count in the optimal set S_o such that $S_o \in \text{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$. Then $a_i > 0$, $\forall i, 1 \leq i \leq m$.*

Proof. We prove by contradiction. Suppose $\exists \sigma, 1 \leq \sigma \leq m$, $a_\sigma = 0$, there exists feature σ that never appears in any feature subset. As $tk > m$, there exist $a_\sigma = 0$ and $a_j \geq 2$ according to the pigeonhole principle. By using feature σ to replace feature j from any one of feature subsets containing it (since feature σ never appears before, the replacement will not create redundancy in feature subsets), we obtain the new feature appearance count $a'_\sigma = 1$ and new repetition count does not change ($R'_\sigma = R_\sigma = 0$) because repetition regarding feature σ cannot be caused by only one appearance among all feature subsets; we also obtain the new feature appearance count $a'_j = a_j - 1 > 0$ and new repetition count $R'_j = R_j - a_j + 1$. After the replacement, the repetition count regarding feature j gets smaller because $R'_j < R_j$. Meanwhile, the repetition counts regarding other features do not change. This means that the replacement reduces the total repetition counts by $(a_j - 1)$, resulting in a contradiction to the optimal set $S_o \in \text{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$. □

Using this lemma, we show that the conjecture is true:

Theorem 2. *Let $a_i, 1 \leq i \leq m$ be the feature appearance count in the set $\mathcal{S}_o$. In the case of tk mod $m = 0$, then $S_o \in \text{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$ if and only if $a_i = a_j$ always holds for any i, j, $1 \leq i \leq m$, $1 \leq j \leq m$. In the case of tk mod $m \neq 0$, then $S_o \in \text{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$ if and only if $a_i - a_j \leq 1$ always holds for any i, j, $1 \leq i \leq m$, $1 \leq j \leq m$.*

Proof. Based on Lemma 1 ($a_i > 0$, $\forall i, 1 \leq i \leq m$), we can derive

$$R_i = \sum_{j=1}^{a_i - 1} j = \frac{a_i^2 - a_i}{2}$$

because: each feature can at most appear once in each feature subset, so the repetitions regarding a feature can be computed by considering every two subsets containing this feature. In this case, we can use a_i to compute $R(\mathcal{S})$ by the definition in formula (1):

$$R(\mathcal{S}) = \sum_{i=1}^{m} R_i = \sum_{i=1}^{m} \frac{a_i^2 - a_i}{2}. \tag{5}$$

We first consider the case of tk mod $m = 0$:

Table 2: Table of Feature Appearances and Repetitions Represented by a_μ and Δa_i

Feature Number	Appearance	Repetition
1	$a_\mu + \Delta a_1$	$\frac{1}{2}[(a_\mu + \Delta a_1)^2 - (a_\mu + \Delta a_1)]$
2	$a_\mu + \Delta a_2$	$\frac{1}{2}[(a_\mu + \Delta a_2)^2 - (a_\mu + \Delta a_2)]$
$\cdots$	$\cdots$	$\cdots$
m	$a_\mu + \Delta a_m$	$\frac{1}{2}[(a_\mu + \Delta a_m)^2 - (a_\mu + \Delta a_m)]$

Case 1. *tk mod $m = 0$.* We first prove necessity. Given $S_o \in \text{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$, when $a_i = a_j$ for any i, j, $1 \le i \le m$, $1 \le j \le m$, we have $a_i = a_\mu$ for any $i,$, $1 \le i \le m$, where $a_\mu = \frac{tk}{m}$ is the average appearance of all features. Hence, based on the equality (5), the total repetition index can be given by $\frac{m}{2}(a_\mu^2 - a_\mu)$, since every feature has the same appearances and repetitions. If there is $a_i \ne a_j$, $1 \le i \le m$, $1 \le j \le m$, then $\exists \sigma, 1 \le \sigma \le m, a_\sigma \ne a_\mu$. We denote by $\Delta a_i = a_i - a_\mu$ the difference between a_i and a_μ. The repetitions regarding each feature can also be represented by a_μ and Δa_i, as shown in Table 2.

Since a_μ is the average appearance of all features, we have

$$\sum_{i=1}^{m} \Delta a_i = \sum_{i=1}^{m} a_i - \sum_{i=1}^{m} a_\mu = 0.$$

Thus, the total repetition index is

$$\sum_{i=1}^{m} \frac{1}{2}[(a_\mu + \Delta a_i)^2 - (a_\mu + \Delta a_i)] = \frac{m}{2}(a_\mu^2 - a_\mu) + \frac{1}{2}\sum_{i=1}^{m} {\Delta a_i}^2$$

which is larger than $\frac{m}{2}(a_\mu^2 - a_\mu)$, because we can derive $\sum_{i=1}^{m} {\Delta a_i}^2 = \sum_{i=1}^{m} (a_i - a_\mu)^2 > 0$ by relying on $\exists \sigma, 1 \le \sigma \le m, a_\sigma \ne a_\mu$. This means that $\exists \sigma, 1 \le \sigma \le m, a_\sigma \ne a_\mu$, contradicts to the condition $S_o \in \text{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$. So in the case of *$tk$ mod $m = 0$*, it is true that, if S_o is optimal, $a_i = a_j$ always holds for any i, j, $1 \le i \le m$, $1 \le j \le m$. This finishes the proof of necessity, also giving a corollary:

Corollary 3. *If tk mod $m = 0$, then* $\min R(\mathcal{S}) = \frac{m}{2}(a_\mu^2 - a_\mu)$.

This corollary can be used to prove sufficiency. Given $a_i = a_j$ for any i, j, $1 \le i \le m$, $1 \le j \le m$, we have $a_i = a_\mu = \frac{tk}{m}$ for any i, $1 \le i \le m$. Hence, we can derive

$$R(\mathcal{S}_o) = \sum_{i=1}^{m} R_i = \sum_{i=1}^{m} \frac{a_i^2 - a_i}{2} = \frac{m}{2}(a_\mu^2 - a_\mu) = \min R(\mathcal{S})$$

which is equivalent to $S_o \in \text{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$. This finishes the proof of sufficiency.

Case 2. *tk mod $m \ne 0$.* We first prove necessity by contradiction. Suppose there exists $a_\sigma - a_\phi \ge 2, 1 \le \sigma \le m, 1 \le \phi \le m$, so subsets containing feature σ are, by at least two, more than subsets containing feature ϕ. Hence, among all subsets containing feature σ, we can find one subset containing feature σ and use feature ϕ to replace feature σ, without making the replaced subset identical with any other subset containing feature ϕ. After the replacement, the repetitions regarding feature σ decrease by $(a_\sigma - 1)$, while the repetitions regarding feature ϕ increase by a_ϕ, according to the equality (5). The total repetition change of the replacement is $(a_\phi - a_\sigma + 1) < 0$. This is a contradiction to $S_o \in \text{argmin}_{\mathcal{S} \in \mathcal{F}([m],t)} R(\mathcal{S})$. Thus, in the case of $tk \ mod \ m \neq 0$, if S_o is optimal, then $a_i - a_j \leq 1$ always holds for any i, j, $1 \leq i \leq m$, $1 \leq j \leq m$. This finishes the proof of necessity, also giving a corollary:

Corollary 4. *If $tk \ mod \ m = \epsilon > 0$, then* $\min R(\mathcal{S}) = \frac{m}{2}(\lfloor a_\mu \rfloor^2 - \lfloor a_\mu \rfloor) + \epsilon \lfloor a_\mu \rfloor$

where $a_\mu = \frac{tk}{m}$ is not an integer because $tk \ mod \ m \neq 0$. There are ϵ features (each with appearance $(\lfloor a_\mu \rfloor + 1)$ and repetition $\frac{1}{2}(\lfloor a_\mu \rfloor^2 + \lfloor a_\mu \rfloor)$) that appear one more time than the rest $m - \epsilon$ features (each with appearance $\lfloor a_\mu \rfloor$ and repetition $\frac{1}{2}(\lfloor a_\mu \rfloor^2 - \lfloor a_\mu \rfloor)$).

With this corollary, we prove sufficiency. If $tk \ mod \ m = \epsilon > 0$ and $a_i - a_j \leq 1$ always holds for any i, j, $1 \leq i \leq m$, $1 \leq j \leq m$, there exist ϵ features that appear one more time than the rest $m - \epsilon$ features. Each of the ϵ features with more appearances $(\lfloor a_\mu \rfloor + 1)$ has repetition $\frac{1}{2}(\lfloor a_\mu \rfloor^2 + \lfloor a_\mu \rfloor)$); each of the $m - \epsilon$ features with less appearances $(\lfloor a_\mu \rfloor)$ has repetition $\frac{1}{2}(\lfloor a_\mu \rfloor^2 - \lfloor a_\mu \rfloor)$). Then, the total repetition can be given by

$$R(\mathcal{S}_o) = \frac{\epsilon}{2}(\lfloor a_\mu \rfloor^2 + \lfloor a_\mu \rfloor) + \frac{m-\epsilon}{2}(\lfloor a_\mu \rfloor^2 - \lfloor a_\mu \rfloor) = \frac{m}{2}(\lfloor a_\mu \rfloor^2 - \lfloor a_\mu \rfloor) + \epsilon \lfloor a_\mu \rfloor = \min R(\mathcal{S})$$

which is equivalent to $S_o \in \text{argmin}_{\mathcal{S} \in \mathcal{F}([m],t)} R(\mathcal{S})$. This finishes the proof of sufficiency. □

Regarding the repetition ratio $r(\mathcal{S})$, Theorem 2, with the equality (2) and (5), can generate a corollary:

Corollary 5. *Let $a_1, a_2, \cdots, a_m$ be the feature appearance counts in the set S_o. Then $S_o \in \text{argmin}_{\mathcal{S} \in \mathcal{F}([m],t)} R(\mathcal{S})$ if and only if $S_o \in \text{argmin}_{\mathcal{S} \in \mathcal{F}([m],t)} r(\mathcal{S})$. In this case,*

$$\min r(\mathcal{S}) = \sum_{i=1}^{m} \frac{a_i^2 - a_i}{|\mathcal{S}_o|^2 - |\mathcal{S}_o|}.$$

With these properties demonstrating the intended set(s), we design feature distribution algorithms that have related characteristics.

4.3 Arranging Features into Trees

4.3.1 Diagonal Distribution Algorithm

Since a k-family (= collection of k-subsets) with total repetition index 0 has a small size $\lfloor \frac{m}{k} \rfloor$, we aim to find a larger k-family with the lowest total repetition index and maximal pairwise repetition index 1 for practical machine learning tasks. Hence, we propose a feature distribution approach using a procedure with several rounds. According to Theorem 2, if each feature equally appears in each round, then the lowest total repetition index can be achieved. This feature distribution approach chooses a group of k-subsets inside $m = k^2$ according to the following rules.

Rule 1. The 1st round (= the starting round) consists of all r_1 k-subsets without repetition. Obviously, $r_1 = [\frac{m}{k}]$. The collection consisting of all k-subsets chosen in the 1st round is denoted by $\mathcal{F}_1$.

Rule 2. The j-th round consists of all r_j k-subsets with at most j pairwise repetitions to each subset in the present or earlier rounds. The collection consisting of all k-subsets chosen in the j-th round is denoted by $\mathcal{F}_j$.

The collection $\mathcal{F}$ we search for is the union of all $\mathcal{F}_i$'s, namely, $\mathcal{F} = \bigcup_{j=1}^{L} \mathcal{F}_j$, where L is the number of rounds.

Usually it is hard to determine the size of each $\mathcal{F}_j$ for $j \geq 2$, since a serious problem occurs from the 2nd round: $r_j =?$

Example 1. *Let's check the simplest case when $m = 4$ and $k = 2$. In this case, one choice of k-subsets in the 1st round are 12 and 34 (using ab to stand for the set $\{a, b\}$) according to Rule 1; now, according to Rule 2, each subset in the 2nd round consists of $k = 2$ numbers from exactly 2 different subsets in the 1st round. Thus, the 2nd round consists of $r_2 = 2 \times 2$ subsets, which are all the subsets of 1234 not appearing in the 1st round, so they are 13, 24, 14, 23. So there are totally $2 + 4 = 6$ subsets with total repetition index $R(\mathcal{S}) = 12$ and repetition ratio $r(\mathcal{S}) = \frac{R(\mathcal{S})}{C_6^2} = 12/15 = 0.8$.*

The general setup of Example 1 is as follows. Let $m = k^2$. Before select any k-subset, we arrange all k^2 numbers into a $k \times k$ matrix A (called to be *the feature matrix*), namely,

$$A = \left(\begin{array}{cccc} a_{11} & a_{12} & ... & a_{1k} \\ a_{21} & a_{22} & ... & a_{2k} \\ ... & ... & ... & \\ a_{k1} & a_{k2} & ... & a_{kk} \end{array} \right) \tag{6}$$

Then choose the first k rows of A as the k-subsets in the first round.

Subsets in the 2nd round are chosen "diagonally" into $l \leq k$ groups, each group consisting of k subsets, as follows.

Denote by T_{1j}^{2l} the k-subset of the l-th group in the 2nd round with "leader" (= the first element of this subset) $(1j)$. Then the 1st group in the 2nd round consists of the following subsets

$$T_{1j}^{2l} = \{1j, 2j, 3j, ..., kj\}, 1 \leq j \leq k$$

which are exactly all columns of A.

When $l \geq 2$, the l-th group consists of the following k-subset chosen diagonally

$$T_{1j}^{2l} = \{1j, 2(j + l - 1), 3(j + 2(l - 1)), ..., k(j + (k - 1)(l - 1))\} \tag{7}$$

where the addition is of course in $\mathbb{Z}/k\mathbb{Z}$, but we use $\{1, 2, ..., k\}$ (replacing 0 with k) instead of $\{0, 1, ..., k - 1\}$ as the indices of the feature matrix A are so.

Algorithm 1, Diagonal Distribution Algorithm (DDA), summarises how our approach arranges features into trees. The 1st round is implemented in line 2-6. The rest of rounds are implemented in line 7-13. Since the operations of formula (7) (line 9) involves k elements, the time complexity of this algorithm is $\mathcal{O}(k^3)$ for all cases including the worst and best case. Similarly, since the chosen k-family $\mathcal{F}$ has cardinality $k + k^2$, the space complexity of this algorithm is also $\mathcal{O}(k^3)$ for all cases. Since DDA does not process data instances, both the time and space complexity of DDA are inconsiderable, compared to training $k + k^2$ trees.

Algorithm 1 Diagonal Distribution Algorithm

```
Input: Number of features per tree $k$; Feature matrix $A = (a_{ij})$, where $a_{ij}$ or $ij$ represents
    the feature number $k(i-1)+j$;
Output: $k$-family $\mathcal{F}$;
let $\mathcal{F}$ be $\emptyset$;
for $j = 1$ to $k$ do
    $T^{1j}_{j1} = \{j1, j2, j3, ..., jk\}$, where each element is from $A$;
end for
$\mathcal{F}_1 = \bigcup_{j=1}^{k}\{T^{1j}_{j1}\}$;
$\mathcal{F} \leftarrow \mathcal{F} \bigcup \mathcal{F}_1$;
for $j = 1$ to $k$ do
    for $l = 1$ to $k$ do
        $T^{2l}_{1j} = \{1j, 2(j+l-1), 3(j+2(l-1)), ..., k(j+(k-1)(l-1))\}$, where each element
        is from $A$ and all algebraic operations are in $\mathbb{Z}/k\mathbb{Z}$ (replacing 0 with $k$);
    end for
end for
$\mathcal{F}_{2+} = \bigcup_{l=1}^{k}\bigcup_{j=1}^{k}\{T^{2l}_{1j}\}$;
$\mathcal{F} \leftarrow \mathcal{F} \bigcup \mathcal{F}_{2+}$;
return $\mathcal{F}$
```

4.4 Total Repetition Index of Arranged Forests with DDA

As stated in Algorithm 1 (DDA), each element of A appears exactly once in

$$T^{1j}_{j1} = \{j1, j2, j3, ..., jk\}, 1 \le j \le k.$$

While in $T^{2l}_{1j} = \{1j, 2(j+l-1), 3(j+2(l-1)), ..., k(j+(k-1)(l-1))\}, 1 \le j \le k, 1 \le l \le k$ (all algebraic operations are in $\mathbb{Z}/k\mathbb{Z}$, replacing 0 with k), there are elements appearing more than once when k is not a prime number. For example, when $k = 6$, $T^{2l}_{11}, 1 \le l \le 6$ contains two of 33 because $[j+2(l-1)]\ mod\ 6 = 3$ at $j = 1, l = 2$ and $j = 1, l = 5$. However, regardless of k being prime or not, each element of A still appears the same number of times in sets of $\mathcal{F}_{2+} = \bigcup_{l=1}^{k}\bigcup_{j=1}^{k}\{T^{2l}_{1j}\}$, which can be illustrated by the following theorem.

Theorem 6. *Each element tu of A appears exactly k times in sets of $\bigcup_{l=1}^{k}\bigcup_{j=1}^{k}\{T^{2l}_{1j}\}$, $T^{2l}_{1j} = \{1j, 2(j+l-1), 3(j+2(l-1)), ..., k(j+(k-1)(l-1))\}, 1 \le j \le k, 1 \le l \le k$ (all algebraic operations are in $\mathbb{Z}/k\mathbb{Z}$, replacing 0 with k).*

Proof. According to Algorithm 1 (DDA), for each element $tu, 1 \le t \le k, 1 \le u \le k$ in $T^{2l}_{1j} = \{1j, 2(j+l-1), 3(j+2(l-1)), ..., k(j+(k-1)(l-1))\}, 1 \le j \le k, 1 \le l \le k$,

$$u = j + (t-1)l - (t-1)\ (mod\ k), 1 \le j \le k, 1 \le l \le k$$

where all algebraic operations are in $\mathbb{Z}/k\mathbb{Z}$, replacing 0 with k. Then, the appearances of a_{tu} can be computed by counting the pairs of j and l which satisfy

$$(u - j + t - 1) = (t-1)l\ (mod\ k), 1 \le j \le k, 1 \le l \le k. \tag{8}$$

Set $t-1=\theta, j-u=x, l-1=y$, the above equation is

$$x+\theta y = 0 \ (mod\ k), 0 \leq \theta \leq k-1. \tag{9}$$

If $\theta = 0$, then $j = u$ ($x = 0$) and any $l, 1 \leq l \leq k$ are the k solutions of equality (8). If $\theta \neq 0$, based on the systems of linear congruences in number theory, a group of solutions can be derived from the solution $x_0 = 0$ and $y_0 = 0$ by

$$\begin{cases} x = x_0 + \dfrac{\theta\gamma}{(1,\theta)} \\ y = y_0 - \dfrac{\gamma}{(1,\theta)} \end{cases} \tag{10}$$

where $\gamma \in \mathbb{Z}$ and $(1,\theta)$ is the greatest common divisor of the coefficients. Since $(1,\theta) = 1$ and y can be any integer, $l = y+1$ (in $\mathbb{Z}/k\mathbb{Z}$, replacing 0 with k) can be any integer in $[1,k]$. For each l, j can be given by $j = \theta\gamma + u$ (in $\mathbb{Z}/k\mathbb{Z}$, replacing 0 with k). Hence, there are always k pairs of j and l that can satisfy equality (8). The number of the pairs also represents appearances of each element tu of A in sets of $\bigcup_{l=1}^{k}\bigcup_{j=1}^{k}\{T_{1j}^{2l}\}$. □

Theorem 6 and 2 show that DDA can always give a k-family $\mathcal{F}_{\mathcal{DDA}}$ with the lowest total repetition index, as summarised in the following corollary.

Corollary 7. *Let $\mathcal{F}_{\mathcal{DDA}}$ be the output of DDA. Then $\mathcal{F}_{\mathcal{DDA}} \in \operatorname{argmin}_{\mathcal{S}\in\mathcal{F}([m],t)} R(\mathcal{S})$.*

In addition, considering that each element of A appears exactly $(k+1)$ times in all selected sets (inside $\mathcal{F}_1 = \bigcup_{j=1}^{k}\{T_{j1}^{1j}\}$ together with $\mathcal{F}_{2+} = \bigcup_{l=1}^{k}\bigcup_{j=1}^{k}\{T_{1j}^{2l}\}$), Corollary 3 can give the total repetition index of DDA's output $\mathcal{F}_{\mathcal{DDA}}$:

$$R(\mathcal{F}_{\mathcal{DDA}}) = \frac{k^4+k^3}{2}. \tag{11}$$

4.5 Pairwise Repetition Index of Arranged Forests with DDA

Unlike the total repetition index, the pairwise repetition index cannot be computed by considering feature appearances. We must take into account the mutual repetitions of elements in every pair of subsets. Say the feature matrix A has an element $tu, 1 \leq t \leq k, 1 \leq u \leq k$ in $T_{1j}^{2l} = \{1j, 2(j+l-1), 3(j+2(l-1)), ..., k(j+(k-1)(l-1))\}, 1 \leq j \leq k, 1 \leq l \leq k$. If tu is also in another subset $T_{1j'}^{2l'}$, then, according to the equality (8), we have

$$\begin{cases} u-j+t-1 = (t-1)l \ (mod\ k) \\ u-j'+t-1 = (t-1)l' \ (mod\ k) \end{cases} \tag{12}$$

where at least one condition in $j' \neq j$ and $l' \neq l$ holds, so that the two subsets are different. There may be more than one pair of j' and l' that can satisfy the equation. Thus, it is challenging to count the pairwise repetition index regarding tu.

However, we are able to derive the maximal pairwise repetition index in certain cases. When k is a prime number, we show that the pairwise repetition index of DDA's output is at most one: $|x \bigcap y| \leq 1$ for all $x, y \in \mathcal{F}$ with $x \neq y$.

Theorem 8. *Let $k = p$ be a prime. Then the maximal pairwise repetition index of DDA's output is 1. Indeed, $|T_{1i}^{2l} \bigcap T_{1j}^{2q}| = 1 - \delta_{lq}$ where δ is the Kronecker delta.*

Proof. In the first round, each of the selected subsets $T_{j1}^{1j} = \{j1, j2, j3, ..., jk\}, 1 \leq j \leq k$ represents a unique row. So the pairwise repetition index across them is always 0. From the 2nd round or later, for any selected subset, there is exactly one element on each row. Thus, any subset with a subset of the first round has the pairwise repetition index at most 1. Then, whether the maximal pairwise repetition is more than 1 depends on the relation between subsets from the 2nd round or later.

From the 2nd round or later, we count shared elements from any two k-subsets T_{1i}^{2l} and T_{1j}^{2q} (i.e., $1 \leq l \leq k$, $1 \leq q \leq k$). Because of the loops in Algorithm 1 (DDA), $l = q$ and $i = j$ cannot be both true. Suppose there is a shared element $tu \in T_{1i}^{2l} \bigcap T_{1j}^{2q}$.

When $l \neq q, i = j$, only the first element is shared because $\{2(i + l - 1), 3(i + 2(l - 1)), ..., k(i + (k - 1)(l - 1))\}$ and $\{2(i + q - 1), 3(i + 2(q - 1)), ..., k(i + (k - 1)(q - 1))\}$ (in $\mathbb{Z}/k\mathbb{Z}$, replacing 0 with k) are totally different. When $l \neq q, i \neq j$, we have $u = i + (t-1)(l-1) = j + (t-1)(q-1)$ and $t = (i-j)(q-l)^{-1} + 1$ (in $\mathbb{Z}/k\mathbb{Z}$, replacing 0 with k). The multiplicative inverse $(q - l)^{-1}$ exists in $\mathbb{Z}/k\mathbb{Z}$ because k is a prime. So t only has one solution and u can be obtained correspondingly. There is exactly one shared element. Thus, when $l \neq q$, $|T_{1i}^{2l} \bigcap T_{1j}^{2q}| = 1$.

When $l = q, i \neq j$, $u = i + (t - 1)(l - 1) = j + (t - 1)(q - 1)$ has no solution (i.e., no shared element). In this case, $|T_{1i}^{2l} \bigcap T_{1j}^{2q}| = 0$.

Overall, the maximal pairwise repetition is 1 and $|T_{1i}^{2l} \bigcap T_{1j}^{2q}| = 1 - \delta_{lq}$. □

By Theorem 8, in the case $m = p^2$ with p a prime, the similarity matrix S_{ls} of each pair $\mathcal{F}_l$ and $\mathcal{F}_s$ (with $l \neq s$) is exactly the all one matrix $J = ee^T$, where $e = (1\ 1\ \cdots\ 1)^T$ is the all one vector. Thus the family $\mathcal{F}$ has total repetition index

$$R(\mathcal{C}) = \sum_{j=1}^{p} p^2 j = \frac{p^4 + p^3}{2} \tag{13}$$

which is exactly the result stated in Theorem 2.

4.6 Estimation on the Total Repetition Index of Random Forests

Random Forests randomly distributes a subset of features into each tree. In this section we attempt to quantify feature overlap between trees of Random Forests by using total repetition index. It is equivalent to answering the following question:

- If one selects a k-family (=collection of k-subsets) at random, what is the total repetition index?

Below we use a pure combinatorial method to solve this problem, in particular, the resulting total repetition index of any k-family is via sampling without replacement (i.e., any two trees use different subsets of features).

Note that existing Random Forests in some machine learning libraries may sample features with replacement (i.e., allowing some trees to use the same subset of features). Also, when the intended tree number is larger than C_m^k (however, C_m^k is often large in real-world scenarios, e.g., $C_{16}^4 = 1820$, so the computational cost of building these trees can be very high), some trees have to sample features with replacement, resulting in higher repetition. Hence, considering sampling without replacement is sufficient for most real-world scenarios.

Since each number appears uniformly in the k-subsets(there are totally C_m^k such subsets), so each number appears exactly in $\frac{kC_m^k}{m}$ such subsets, yielding a probability $p(a \in A) = \frac{k}{m}$ for any fixed number $a \in [m]$ and any randomly chosen k-subset A.

Given any k-subset A, what is the probability of $A \bigcap B = \emptyset$ for B a randomly chosen k-subset? Since

$$A \bigcap B = \emptyset \iff |A \bigcap B| = 0 \iff \forall b \in B, b \notin A,$$

we have

$$p(A \bigcap B = \emptyset) = p(\forall b \in B, b \notin A) = \left(1 - \frac{k}{m}\right)^k. \tag{14}$$

Similarly, for each $1 \leq \alpha \leq k-1$, the probability $p_\alpha = p(|A \bigcap B| = \alpha)$ of B intersecting A at exactly α points is

$$p_\alpha = p(|A \bigcap B| = \alpha) = C_k^\alpha \left(\frac{k}{m}\right)^\alpha \left(1 - \frac{k}{m}\right)^{k-\alpha}. \tag{15}$$

Suppose $\mathcal{F}$ is a randomly chosen k-family, what is the total repetition index of $\mathcal{F}$? For each $A \in \mathcal{F}$, any other subset in $\mathcal{F}$ intersects A at exactly α points in probability p_α. Thus, from (15), the expectation $E(r_\alpha(\mathcal{F}))$ of the repetition index α of $\mathcal{F}$ is

$$E(r_\alpha(\mathcal{F})) = \alpha \left(\sum_{j=1}^{|\mathcal{F}|-1} j\right) p_\alpha = \alpha \frac{|\mathcal{F}|(|\mathcal{F}|-1)}{2} C_k^\alpha \left(\frac{k}{m}\right)^\alpha \left(1 - \frac{k}{m}\right)^{k-\alpha}. \tag{16}$$

Therefore, the totally repetition index of a randomly chosen k-family $\mathcal{F}$ has the following expectation

$$R(\mathcal{F}) = \sum_{\alpha=1}^{k-1} E(r_\alpha(\mathcal{F})) = \frac{|\mathcal{F}|(|\mathcal{F}|-1)}{2} \sum_{\alpha=1}^{k-1} \alpha p_\alpha. \tag{17}$$

It is very difficult to get a good estimation result of the above formula (17) in general. However, we can get some helpful bounds as m (equivalently, k) goes to infinity. To see this, it suffices to estimate the factor $\sum_{\alpha=1}^{k-1} \alpha p_\alpha$. By the well-known fact that the largest term of a Bernoulli distribution $B(m, q)$ is always around $(m+1)q$. In our case, $m = k$ and $q = \frac{1}{k}$, so the largest terms are the first some terms. In fact, the first 3 terms are enough for our purpose:

$$\begin{aligned}
&\sum_{\alpha=1}^{k-1} \alpha P_\alpha \geq P_1 + 2P_2 + 3P_3 \\
&= C_k^1 q(1-q)^{k-1} + 2C_k^2 q^2 (1-q)^{k-2} + 3C_k^3 q^3 (1-q)^{k-3} \\
&= 3(1 - \frac{4}{3k}) \left(1 - \frac{1}{k}\right)^{k-2}
\end{aligned}$$

Therefore the estimation for the expectation R of the total repetition of the random algorithm is

$$R(\mathcal{F}) \geq \frac{3|\mathcal{F}|(|\mathcal{F}|-1)}{2}\left(1-\frac{1}{k}\right)^{k-2}(1-\frac{4}{3k}). \tag{18}$$

When $k \geq 6$, the above formula yields the following estimation

$$R(\mathcal{F}) = \sum_{\alpha=1}^{k-1} E(r_\alpha(\mathcal{F})) \geq \frac{3|\mathcal{F}|(|\mathcal{F}|-1)}{2e} \tag{19}$$

where e is the base of the natural logarithm.

If $k \leq 5$, the exact result can be obtained as below.

4.6.1 Comparison between prime DDA and random algorithm.

From DDA, we obtain such a family with exactly $k+k^2$ elements (the first k subsets being disjoint) with total repetition index $\frac{1}{2}(k^4+k^3) = \frac{m^2}{2}(1+\frac{1}{\sqrt{m}})$ by the equality (11), which is equivalent to $\frac{m^2}{2}$, and of the infinity rank $O(m^2)$.

How about the random algorithm? According to the formula (19), a randomly chosen k-family $\mathcal{F}$ with $|\mathcal{F}| = k^2+k$ elements will have the total repetition index:

$$R(\mathcal{F}) = \sum_{\alpha=0}^{k-1} r_\alpha(\mathcal{F}) \geq \frac{3(k^2+k)(k^2+k-1)}{2e}. \tag{20}$$

So the infinity rank of the total repetition index of the random algorithm is at least $O(m^2)$, with a constant multiple larger than $\frac{3}{2e} > 0.5518$.

Hence, in getting low total repetition, DDA is better than the random algorithm.

Let's check some smaller cases. When $k = 5$, DDA gives a k-family with $k^2+k = 30$ subsets and repetition index $\frac{k^4+k^3}{2} = 375$, while the random algorithm yielding a k-family with 30 subsets will produce the total repetition index (cf. formula (17)) at least

$$\frac{(k^2+k)(k^2+k-1)}{2}\sum_{\alpha=1}^{k-1}\alpha p_\alpha = 15 \times 29 \times \frac{4^2 \times 951}{5^6} = 435 \times 0.973824 = 423.61344$$

which is much larger than that of DDA. The cases of $k < 5$ are similar.

When $m = k^2$ with k a large integer, $\left(1-\frac{1}{k}\right)^k$ goes to $\frac{1}{e} \approx 0.367$, so the total repetition index in the DDA formula (11) is in the same infinity rank to the factor $|\mathcal{F}|(|\mathcal{F}|-1) = k^4+k^3-2k^2-k+1$. Compared with the total repetition index $\frac{k^4+k^3}{2}$ in DDA formula (11), the expected total repetition index of the random algorithm almost doubles that of DDA for large k.

Hence, theoretically, DDA can achieve lower total repetition indices than the random algorithm.

4.7 Modified Diagonal Distribution Algorithm

Besides DDA, we propose another algorithm that can arrange features with the lowest total repetition and low maximal pairwise repetition index, when $m = k^2$, $k + 1 = p$ where $p > 4$ is a prime number. The main idea of DDA can be generalised to the case $k + 1 = p$ via a mild modification of the case that k is itself a prime number. We call the modified algorithm Modified Diagonal Distribution Algorithm (MDDA). Its procedures are shown in Algorithm 2 (see Appendix A).

Suppose $k + 1 = p$ is a prime. Then enlarge the feature matrix A in (6) to an $(k + 1) \times (k + 1) = p \times p$ matrix $A(+)$ with $(p1, p2, ..., pp)$ as the p-th row respectively $(1p, 2p, ..., pp)^T$ as the p-th column:

$$A(+) = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1k} & a_{1p} \\ a_{21} & a_{22} & ... & a_{2k} & a_{2p} \\ ... & ... & ... & & \\ a_{k1} & a_{k2} & ... & a_{kk} & a_{kp} \\ a_{p1} & a_{p2} & ... & a_{pk} & a_{pp} \end{pmatrix} \tag{21}$$

Now proceed the 1st round as before and we get the first k rows (but deleting the element from last column) as the first k-family $\mathcal{F}_1$. Then we go on the 2nd round by first choosing the first k columns each deleting the element from the last row; then we use DDA and get a typical $p = k + 1$-subset as $\{1j_1, 2j_2, ..., pj_p\}$, with exactly one $j_t = p$. There are exactly two cases. Case 1 is $j_p = p$: then just remove this $pj_p = pp$, we get a true k-subset consisting of entries in A. Case 2 is $j_p < p$: then combine these two numbers tj_t and pj_p into a real one tj_p, we get a true k-subset consisting of entries in A. Altogether, we get a k-family of size $k + k + kp = k^2 + 3k$ from MDDA. Comparatively, DDA gives only $k^2 + k$.

Besides the size of output, there is another difference between DDA and MDDA. The former always produces a k-family with maximal pairwise repetition 1 (i.e. two subsets in $\mathcal{F}$ intersect at most at 1 point) when $k = p$ is prime, while the latter comes with maximal pairwise repetition at most 3 when $k = p - 1$, due to the combining operation (i.e., line 15-32 of Algorithm 2, see Appendix A).

4.7.1 Total and Pairwise Repetition Index of Arranged Forests with MDDA

Here, we analyse the total and pairwise repetition index of MDDA's output when $k = p-1$ ($p > 4$ is prime).

Let $m = k^2$ and $k = p - 1$ with p a prime number. MDDA produces a k-family $\mathcal{F}$ with $k^2 + 3k$ subsets. Those subsets selected in the first two rounds are the same as DDA, so they do not increase any repetition index of $\mathcal{F}$. However, those subsets selected from the 3rd round do increase the total repetition index, due to the combination operations, compared with DDA. Similar to the case of DDA, we show that each element $tu, 1 \leq t \leq k, 1 \leq u \leq k$ of matrix $A(+)$ appears the same number of times.

Theorem 9. *Each element $tu, 1 \leq t \leq k, 1 \leq u \leq k$ of matrix $A(+)$ appears exactly $k + 3$ times in the sets of the output $\mathcal{F}$ of MDDA, when $k = p - 1, p > 4$ is a prime.*

Proof. In the 1st round, each element tu appears exactly once in the sets $T_{j1}^{1j} = \{j1, j2, j3, ..., jk\}, 1 \leq j \leq k$. From the 2nd round, since the procedures of Algorithm 1

(DDA) and Algorithm 2 (MDDA) are identical before the combining operations of MDDA, each element tu appears exactly $k+1$ times in these procedures, as a consequence of Theorem 6. So far, each element tu appears exactly $k+2$ times before we consider the combining operations. Say element tu is generated in the combining operation on element $t(k{+}1) \in T_{1j}^{2l}$ and element $(k+1)u \in T_{1j}^{2l}$, then we have

$$\begin{cases} j + (t-1)(l-1) = 0 \ (mod\ k+1) \\ \qquad\qquad\qquad u = j + k(l-1) \ (mod\ k+1) \end{cases} \tag{22}$$

where all algebraic operations are in $\mathbb{Z}/p\mathbb{Z}$, replacing 0 with $k+1$. Hence, the appearances of each element tu in the combining operation can be counted by investigating the number of j, l pairs that satisfy the above equations. From the two equations, we deduce

$$u = (1-t)(l-1) + k(l-1) \ (mod\ k+1).$$

As stated before, $1 \le t \le k, 1 \le u \le k$, so that $(l-1) \neq 0$ and $(1-t+k) \neq 0$. Then $1 \le (1-t+k) < (k+1)$. As $k+1$ is a prime, by Fermat's little theorem, we have

$$(1-t+k)^k = 1 \ (mod\ k+1).$$

In addition, according to the properties of finite fields, we obtain the multiplicative inverse of $(1-t+k)$, $(1-t+k)^{-1} = (1-t+k)^{k-1}$, because

$$(1-t+k)(1-t+k)^{k-1} = 1 = (1-t+k)(1-t+k)^{-1} \ (mod\ k+1).$$

Thus, for each element tu, there is exact one valid value of l because

$$l = u(1-t+k)^{-1} + 1 = u(1-t+k)^{k-1} + 1 \ (mod\ k+1).$$

Also, there is exact one valid value of j because

$$j = u(1-t)(1-t+k)^{k-1} \ (mod\ k+1).$$

This means that, for each element tu, there is only exact one pair of j, l that satisfies the two equations (22). Overall, each element tu appears exactly $k+3$ times in the sets of the output $\mathcal{F}$ of MDDA, when $k = p-1, p > 4$ is a prime. □

By equations (22) and Theorem 2, we derive that MDDA can always give a k-family $\mathcal{F}_{\mathcal{MDDA}}$ with the lowest total repetition index when $k = p-1, p > 4$ is a prime, as summarised in the following corollary.

Corollary 10. *Let k-family $\mathcal{F}_{\mathcal{MDDA}}$ be the output of MDDA and $k = p-1$ where $p > 4$ is a prime. Then $\mathcal{F}_{\mathcal{MDDA}} \in \operatorname{argmin}_{\mathcal{S} \in \mathcal{F}([m],t)} R(\mathcal{S})$.*

Since each element appears exactly $(k+3)$ times in all selected sets, Corollary 3 can give the total repetition index of MDDA's output $\mathcal{F}_{\mathcal{MDDA}}$ when $k = p-1, p > 4$ is a prime:

$$R(\mathcal{F}_{\mathcal{MDDA}}) = \frac{k^4 + 5k^3 + 6k^2}{2}. \tag{23}$$

Thus, theoretically, when $k = p-1$ where $p > 4$ is a prime, the total repetition index of MDDA's output is lower than that of the random algorithm (showed in Section 4.6).

Also, we show that the combination operations can increase the pairwise repetition index of MDDA's output, compared to that of DDA's output.

Theorem 11. *The maximal pairwise repetition index of MDDA's output is at most 3 when $k = p - 1, p > 4$ is a prime.*

Proof. In the first round, the selected subsets $T^{1j}_{j1} = \{j1, j2, j3, ..., jk\}, 1 \leq j \leq k$ are the same as the case in DDA. All elements from each subset are in a unique row, and each element of a subset is in a unique column. The pairwise repetition index across them is always 0. The second round generates $\{1l, 2l, 3l, ..., kl\}, 1 \leq l \leq k$ where each element of a subset is in a unique row and all elements of each subset are in a unique column. Hence, any two subsets from the first two rounds have pairwise repetition index at most 1. From the 3rd round or later, after the combination operation, every selected subset has at most 2 elements in one row and at most 2 elements in one column. Thus, any subset from the 3rd round or later has at most 2 identical elements with any subset selected before the 3rd round.

Thus, whether the maximal pairwise repetition is more than 2 depends on the relation between subsets from the 3rd round or later. Consider two k-subsets from the 3rd round or later, say T^{2l}_{1i} and $T^{2s}_{1j},\ l, s \geq 2$.

By Theorem 8, $|T^{2l}_{1i} \bigcap T^{2s}_{1j}| = 1 - \delta_{ls}$ before the combining operation. After the combining operation, suppose tu is the new element of T^{2l}_{1i}. When

$$i + (t-1)(l-1) = j + (t-1)(s-1)\ (mod\ k+1),$$

tu is also an element of T^{2s}_{1j}. Likewise, the new element of T^{2s}_{1j} generated by the combining operation can also be an element of T^{2l}_{1i}. In summary, after the combining operation, $|T^{2l}_{1i} \bigcap T^{2s}_{1j}| \leq 3$, meaning that the maximal pairwise repetition index is at most 3. □

5. Experiments

5.1 Empirical Validation of Repetition

We first conduct empirical validation of the feature repetition for Arranged Forests (i.e., DDA and MDDA) and Random Forests (i.e., random sampling without replacement).

5.1.1 DDA and Random Sampling

We wrote a computer program to run DDA and the random algorithm (1000 different trials per k) with $k \in [3, 20]$. The empirical validation results (Figure 1) confirm that DDA is better to achieve a low total repetition index. The empirical validation results are in line with our theoretical result.

Also, regarding the maximal pairwise repetition index, the empirical validation results (Figure 2) confirm that, when k is a prime, the maximal pairwise repetition index of DDA's output is 1, lower than the lowest cases of the random algorithm.

5.1.2 MDDA and Random Sampling

With the same condition, we compared MDDA and the random algorithm (1000 different trials per k) with $k \in [3, 20]$. The empirical validation results (Figure 3) are in line with our theoretical results, also confirming that MDDA can achieve a lower total repetition index in the cases of $k = p - 1$ ($p > 4$ is a prime). However, results indicate that, compared to

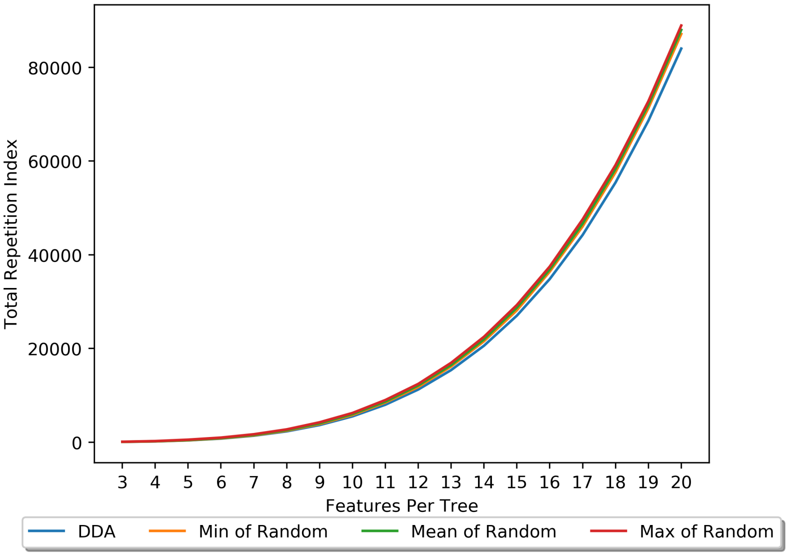


Figure 1: Empirical validation results of DDA and the random algorithm (1000 different trials) regarding the total repetition index.

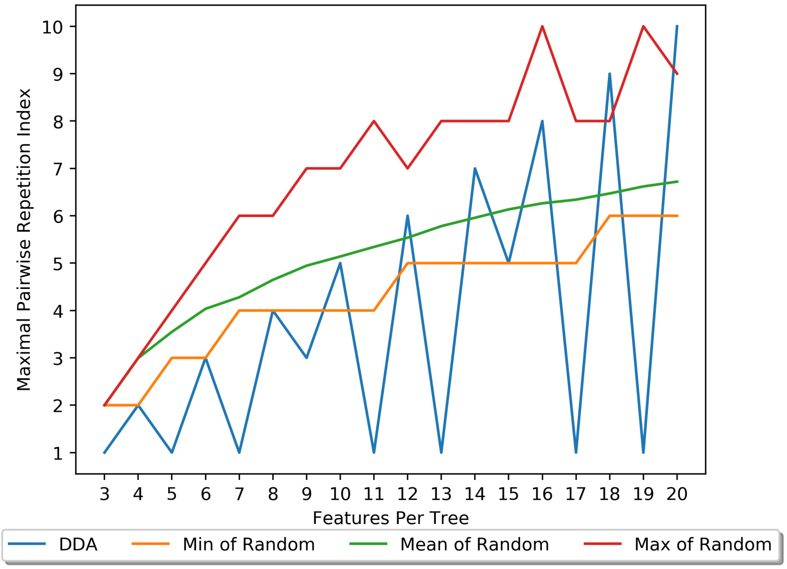


Figure 2: Empirical validation results of DDA and the random algorithm (1000 different trials) regarding the maximal pairwise repetition index.

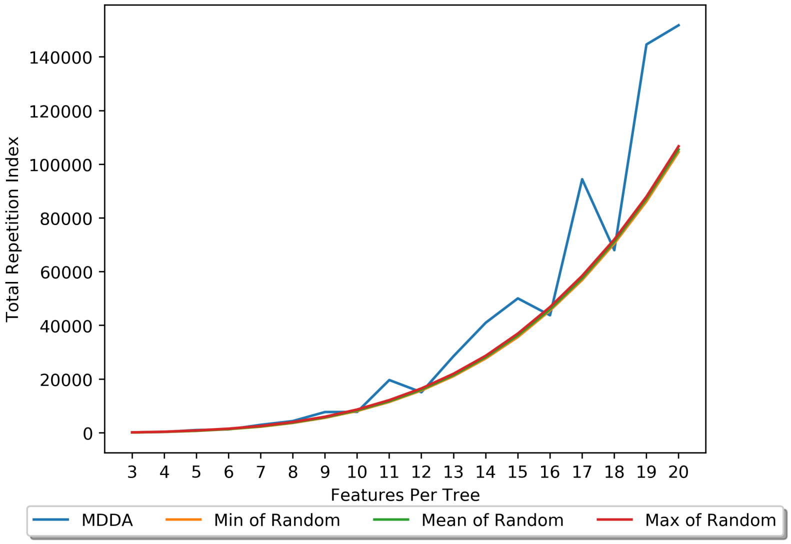


Figure 3: Empirical validation results of MDDA and the random algorithm (1000 different trials) regarding the total repetition index.

the random algorithm, MDDA may obtain a higher total repetition index in the cases that $k+1$ is a composite number.

Regarding the maximal pairwise repetition index, the empirical validation results (Figure 4) confirm that, when $k+1$ is a prime, the maximal pairwise repetition index of DDA's output is at most 3, no more than the lowest cases of the random algorithm. However, results indicate that, when $k+1$ is not a prime, the random algorithm may outperform MDDA in achieving a lower maximal pairwise repetition index.

5.2 Real-World Dataset and Experiment

We now compare Arranged Forests and Random Forests in a real-world prediction task. First, we introduce the real-world dataset and our experiment settings. Then, we construct Arranged Forests and Random Forests for this machine learning task and compute their repetition indices. Finally, we test the two kinds of models and compare their machine learning performance.

5.2.1 Real-World Dataset and Experiment Settings

On UCI Machine Learning Repository by Dua and Graff (2017), we selected a dataset for Parkinson's Disease Classification uploaded by Sakar et al. (2019). This dataset contains processed speech samples of 252 subjects (188 patients and 64 healthy individuals). Each subject performed three repetitions in data collection. Thus, the number of instances is 756. Besides the label of whether a subject has Parkinson's Disease, this dataset contains 753 features regarding time frequency, Wavelet Transform, tunable Q-factor wavelet transform (TQWT), Mel Frequency Cepstral Coefficients (MFCCs) and Vocal Fold.

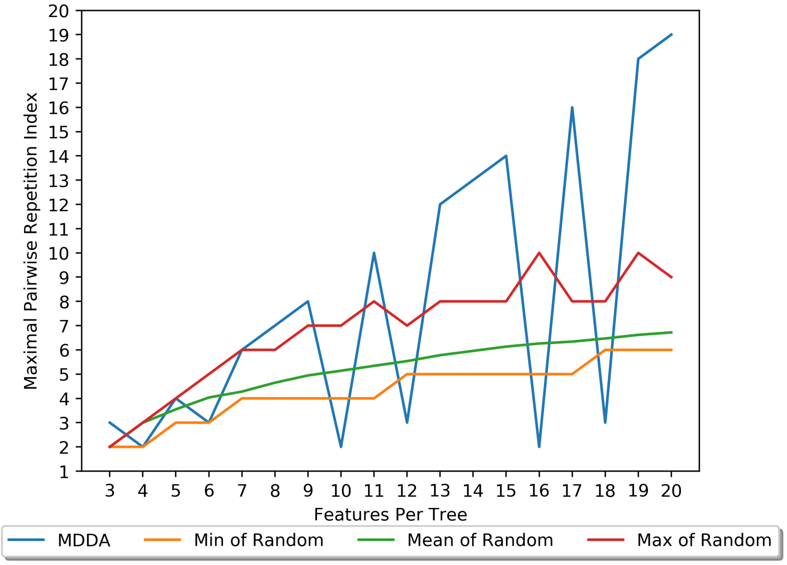


Figure 4: Empirical validation results of MDDA and the random algorithm (1000 different trials) regarding the maximal pairwise repetition index.

Table 3: Experimental results of total repetition indices of Arranged Forests and Random Forests ($Ntree = 552$, $Mtry = 23$). Arranged Forests was constructed by DDA. The results of Random Forests were based on 1000 different trials of random feature distributions.

Total Repetition Index	Random Forests	Arranged Forests(DDA)
Min	150534	146004
Mean	152056.497	146004
Max	153257	146004

We used 70% instances as the training set and 30% as the test set. Instances from one subject were not split into two sets. Since the current design and analysis of Arranged Forests is limited to the case $m = k^2$ (m is the total feature number and k is the feature number of each tree), we only used the first 529 features so that each tree has 23 features.

During training time, the bootstrap process and parameters of trees are identical in Arranged Forests and Random Forests. Also, the total number of trees $Ntree = 552$ (since DDA gives a 23-family with size $23 + 23^2 = 552$) are identical in Arranged Forests and Random Forests. Hence, the only difference is the feature distribution of each tree: Arranged Forests using DDA (23 is a prime); Random Forests using random sampling without replacement.

Overall, we created 1 model of Arranged Forests and 1000 models of Random Forests (1000 different random feature distributions).

Table 4: Experimental results of maximal pairwise repetition indices of Arranged Forests and Random Forests ($Ntree = 552$, $Mtry = 23$). Arranged Forests was constructed by DDA. The results of Random Forests were based on 1000 different trials of random feature distributions.

Maximal Pairwise Repetition Index	Random Forests	Arranged Forests(DDA)
Min	6	1
Mean	7.096	1
Max	11	1

5.2.2 Repetition Indices of Arranged Forests and Random Forests

Table 3 shows the experimental results of total repetition indices of Arranged Forests (DDA) and Random Forests. Among 1000 random trials for Random Forests, the minimal total repetition index is 150534; average 152056.497; maximum 153257. In contrast, the total repetition index of Arranged Forests is 146004, much lower than any case of Random Forests. According to the equality (13), when $k = p = 23$, the exact total repetition index of Arranged Forests (DDA) should be $\frac{23^4+23^3}{2} = 146004$ which is in line with the empirical result.

Table 4 shows the experimental results of maximal pairwise repetition indices of Arranged Forests (DDA) and Random Forests.

Among 1000 random trials for Random Forests, the maximal pairwise repetition ranges from 6 to 11, with an average 7.096. Comparatively, the maximal pairwise repetition of Arranged Forests (DDA) is only 1, which is in line with the statement of Theorem (8).

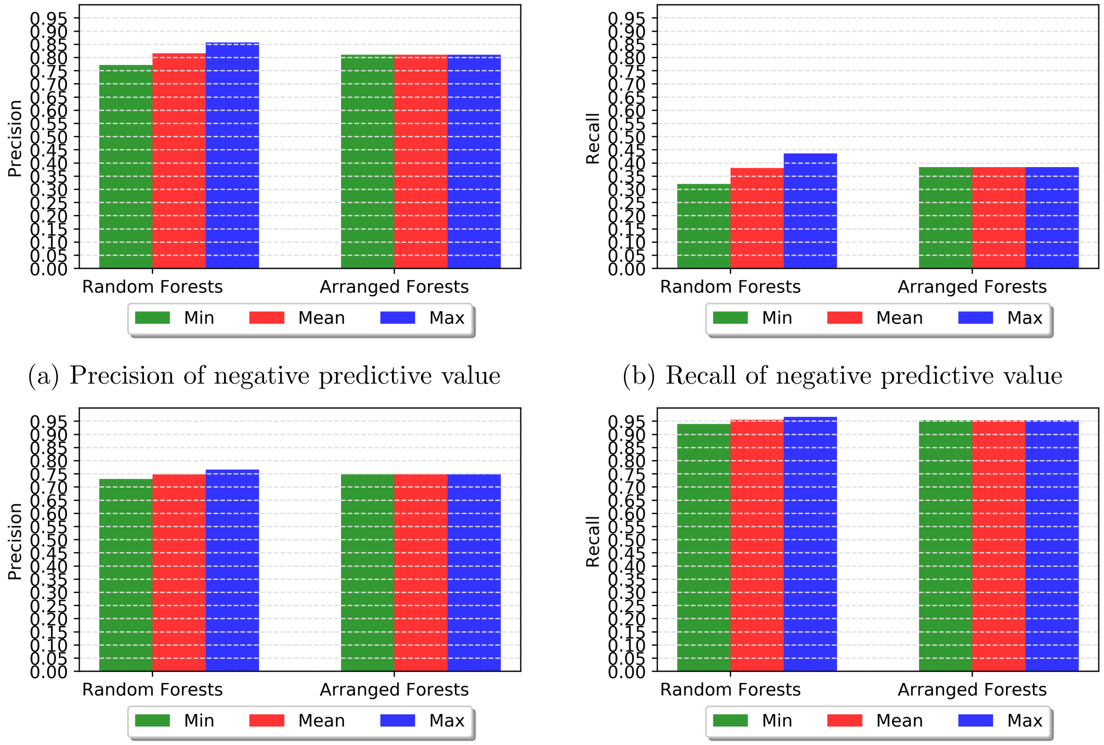
5.2.3 Classification Performance of Arranged Forests and Random Forests

Figure 5 summarises the classification performance results of Arranged Forests and Random Forests. For negative samples, Figure 5a shows the precision: Random Forests Min 0.771, Mean 0.816, Max 0.857; Arranged Forests 0.811. Figure 5b shows the recall of negatives: Random Forests Min 0.321, Mean 0.381, Max 0.436; Arranged Forests 0.385. For positives, Figure 5c shows the precision: Random Forests Min 0.731, Mean 0.748, Max 0.766; Arranged Forests 0.749. Figure 5d shows the recall of positives (this is the most important measure in medical scenarios): Random Forests Min 0.940, Mean 0.955, Max 0.967; Arranged Forests 0.953.

Regarding any of the four measures, we observed that Arranged Forests can approximately achieve the average performance of Random Forests and significantly outperform the bad models of Random Forests. We also noticed substantial differences among the best, average and worst cases of Random Forests.

6. Discussion and Conclusion

In this paper we aim to improve Random Forests (also other similar ensemble learning algorithms) by reducing feature overlap between random trees. The concern is that random trees heavily relying on some specific features may not generate good output. To quantify



(a) Precision of negative predictive value

(b) Recall of negative predictive value

(c) Precision of positive predictive value

(d) Recall of positive predictive value

Figure 5: Classification performance results of Arranged Forests and Random Forests.

overlap between a given number of feature sets, we proposed two measures: pairwise and total repetition index. We also proposed two feature distribution algorithms (DDA, MDDA) to construct decision trees with low feature repetition for different $Mtry$ parameters. And we call the models Arranged Forests, since their feature sets of trees are arranged, not randomly selected.

We provided mathematical calculation and properties regarding the two repetition indices of feature sets in Arranged Forests. We also gave a mathematical estimation for the total repetition index of feature sets in Random Forests. Through mathematical analysis and empirical validation, we showed that Arranged Forests with certain parameters can achieve lower repetition indices, compared to Random Forests.

Using a real-world dataset, we conducted experiments to empirically compare the predictive performance of Arranged Forests and Random Forests. Empirical results show that Arranged Forests can approximately achieve the average predictive performance of Random Forests and significantly outperform the bad models of Random Forests.

We observed that different feature distributions can significantly affect the predictive performance of Random Forests. Through low-overlap arrangement of features into trees, Arranged Forests can be an efficient replacement of Random Forests to obtain the average predictive performance and avoid the poor predictions caused by improper random feature combinations. Note that it consumes more resources to obtain the average predictive performance by using multiple Random Forests models. Thus, Arranged Forests is a suitable machine learning algorithm in the trend of green computing against climate change.

6.1 Limitations and Future Work

Our current design and analysis of Arranged Forests have some limitations. Future work can generalise Arranged Forests in more cases and better investigate the feature distribution problem of ensemble learning.

First, to train Arranged Forests, given a total number m of features, we require $m = k^2$ where k is a positive integer. Considering that the case $m \neq k^2$ is too complicated, future work may first investigate cases such as $m = 2^k$. Second, our analysis on the total repetition index of Arranged Forests constructed by MDDA only considers $k = p - 1$ (p is a prime) within the case $m = k^2$. Future work can attempt to mathematically compute the total repetition index of Arranged Forests constructed by MDDA in other conditions, maybe starting from $k = p + 1$. Also, future work can further investigate the properties regarding the pairwise repetition index of Arranged Forests.

In addition, beyond Arranged Forests, although we gave an estimation of the total repetition index of Random Forests, the mathematical calculation for the exact range is still a challenge. Researchers can make efforts to find better estimation about such randomness regarding the total and pairwise repetition index. Moreover, future work may propose new algorithms to arrange features with low total and pairwise repetition indices. Researchers can investigate how low the indices can be achieved by their algorithms when a certain number of trees/predictors are needed.

Acknowledgments

Yuehui Zhang is supported by NSFC under Grant Nos. 11671258 and 11771280, by NSF of Shanghai Municipal under Grant No. 17ZR1415400.

Appendix A.

In this appendix, we show the procedures of MDDA in Algorithm 2.

Algorithm 2 Modified Diagonal Distribution Algorithm

```
Input: Number of features per tree $k$; Feature matrix $A = (a_{ij})$, where $a_{ij}$ or $ij$ represents the
    feature number $k(i-1)+j$;
Output: $k$-family $\mathcal{F}$;
extend $A$ to $A(+)$ which has $k+1$ rows and columns.
set values of additional elements in $A(+)$ to be 0 (or arbitrary valid numbers). These values will
    not be used to represent features.
let $\mathcal{F}$ be $\emptyset$;
for $j = 1$ to $k$ do
    $T^{1j}_{j1} = \{j1, j2, j3, ..., jk\}$, where each element is from $A(+)$;
end for
$\mathcal{F}_1 = \bigcup_{j=1}^{k}\{T^{1j}_{j1}\}$;
$\mathcal{F} \leftarrow \mathcal{F} \bigcup \mathcal{F}_1$;
for $j = 1$ to $k+1$ do
    for $l = 1$ to $k+1$ do
        $T^{2l}_{1j} = \{1j, 2(j+l-1), 3(j+2(l-1)), ..., (k+1)(j+k(l-1))\}$, where each element is from
            $A(+)$ and all algebraic operations are in $\mathbb{Z}/(k+1)\mathbb{Z}$ (replacing 0 with $k+1$);
        if l is 1 and j is (k+1) then
            Continue;
        end if
        if l is 1 then
            remove all $(k+1)y_0$ from $T^{2l}_{1j}, 1 \leq y_0 \leq k+1$;
        end if
        for all $x_1y_1$ in $T^{2l}_{1j}$ do
            if $x_1y_1$ is $(k+1)(k+1)$ then
                remove $x_1y_1$ from $T^{2l}_{1j}$;
            else
                if $x_1$ is $(k+1)$ then
                    find $x_2(k+1)$ in $T^{2l}_{1j}$;
                    remove $(k+1)y_1$ and $x_2(k+1)$ from $T^{2l}_{1j}$;
                    add $x_2y_1$ into $T^{2l}_{1j}$;
                else if $y_1$ is $(k+1)$ then
                    find $(k+1)y_2$ in $T^{2l}_{1j}$;
                    remove $x_1(k+1)$ and $(k+1)y_2$ from $T^{2l}_{1j}$;
                    add $x_1y_2$ into $T^{2l}_{1j}$;
                end if
            end if
        end for
        $\mathcal{F} \leftarrow \mathcal{F} \bigcup \{T^{2l}_{1j}\}$;
    end for
end for
return $\mathcal{F}$
```

References

Béla Bollobás, Bhargav P Narayanan, and Andrei M Raigorodskii. On the stability of the erdős–ko–rado theorem. *Journal of Combinatorial Theory, Series A*, 137:64–78, 2016.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Particia A Carey and Anant P Godbole. Partial covering arrays and a generalized erdös-ko-rado property. *Journal of Combinatorial Designs*, 18(3):155–166, 2010.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam, 1982.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

Tin Kam Ho. Nearest neighbors in random subspaces. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 640–648. Springer, 1998a.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998b.

Barbara FF Huang and Paul C Boutros. The parameter sensitivity of random forests. *BMC bioinformatics*, 17(1):331, 2016.

Kaggle Inc. Kaggle, 2019. URL `https://www.kaggle.com/`.

Pallika Kanani and Prem Melville. Prediction-time active feature-value acquisition for cost-effective customer targeting. *Advances In Neural Information Processing Systems (NIPS)*, 2008.

Vrushali Y Kulkarni and Pradeep K Sinha. Pruning of random forest classifiers: A survey and future directions. *2012 International Conference on Data Science & Engineering (ICDSE)*, pages 64–68, 2012.

Feng Nan, Joseph Wang, and Venkatesh Saligrama. Feature-budgeted random forest. In *International Conference on Machine Learning*, pages 1983–1991, 2015.

Philipp Probst and Anne-Laure Boulesteix. To tune or not to tune the number of trees in random forest. *Journal of Machine Learning Research*, 18(181):1–18, 2018. URL `http://jmlr.org/papers/v18/17-269.html`.

C. Okan Sakar, Gorkem Serbes, Aysegul Gunduz, Hunkar C. Tunc, Hatice Nizam, Betul Erdogdu Sakar, Melih Tutuncu, Tarkan Aydin, M. Erdem Isenkul, and Hulya Apaydin. A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform. *Applied Soft Computing*, 74:255–263, 2019.

Emanuel Sperner. Ein satz über untermengen einer endlichen menge. *Mathematische Zeitschrift*, 27(1):544–548, 1928.

Stacey J Winham, Robert R Freimuth, and Joanna M Biernacka. A weighted random forests approach to improve predictive performance. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6(6):496–505, 2013.

Abraham J. Wyner, Matthew Olson, Justin Bleich, and David Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *Journal of Machine Learning Research*, 18(48):1–33, 2017. URL `http://jmlr.org/papers/v18/15-240.html`.