Technological Forecasting & Social Change 175 (2022) 121371

Contents lists available at ScienceDirect

Technological Forecasting & Social Change

journal homepage: www.elsevier.com/locate/techfore

Self-Adaptive bagging approach to credit rating

Ni He [*,a], Wang Yongqiao [a], Jiang Tao [b], Chen Zhaoyu [c]

[a] School of Finance, Zhejiang Gongshang University, Hangzhou, China
[b] School of Statistics and Mathematics, Hangzhou Commercial College, Zhejiang Gongshang University, Hangzhou, China
[c] Operational Administration, Industrial Securities Co., Ltd., 29 Jiefang East Road, Jianggan District, Hangzhou, China

ARTICLE INFO

Keywords:
Ensemble learning
Self-Organising learning
Decision tree
Credit rating

ABSTRACT

We propose an enhanced bagging strategy based on self adaptive learning. Recent work in computational biology has seen an increasing use of ensemble learning methods due to their unique advantages in dealing with small sample size, high-dimensionality, and complex data structures. The proposed strategy has a unique advantage in dealing with classification tasks (i.e., credit rating in this study) with data on a relatively small sample size but a large number of heterogeneously distributed features. The self-organising learning mechanism makes the traditional bagging strategy more efficient in terms of model structure. Each submodel in the bagging network becomes a self-adapting base learner in credit rating. In order to prove the applicability and to measure the performance of the proposed method, we carried out a validation using three different traditional algorithms over both artificial datasets and real market data. The results of the proposed algorithm are, on average, better in most of the comparative experiments, while maintaining a model structure that is less complex.

1. Introduction

Credit rating is a process of assessing risks associated with lending to an individual or an organisation. Credit rating, which aims to identify applicants with bad credit who may have a high possibility of defaulting, has become one of the primary ways for financial firms to assess their operational risks (Huang et al., 2007; Jiang and Packer, 2019). In the last few years, in both developed countries and developing countries, the market for credit-associated products has increased enormously. For example, peer-to-peer lending has not only been encouraged by governments, but has also been widely accepted by the general public. Some individuals and organisations have invested heavily in these products but have ended up with huge losses or even in bankruptcy. Therefore, the development of some reliable and *interpretable* approaches to assess the creditworthiness of these products is crucial. Today, a large amount of information about borrowers - such as past borrowing and repayment records, financial status, social network, behavioural patterns, levels of insolvency, and relative likelihood of being fraudulent - is stored and analysed. The challenge is evident: the number of the features are redundant; the usefulness of many features is unstable; and there is no consistent pattern (a group of combined features) to identify. Furthermore, investments with very high credit are surely associated with considerably lower returns, while investments with a clear record of a very bad credit history are easy to identify. These two kinds of investments are either unattractive or easily avoidable. Most investors are seeking opportunities with an acceptable credit level and a good payoff. Therefore, the main purpose of credit rating is to find a *boundary* between safe and unsafe investments. Such a *boundary* also represents a group of investments have the highest utility in terms of balanced higher returns and low variances.

The most popular traditional statistical approaches applied to credit rating, such as logistic regression, decision tree, are relatively interpretable. Traditional statistical techniques always require assumptions such as multivariate normality for independent variables (Huang et al., 2004). **However such assumptions on data distribution are not appropriate for credit-rating data**. Hardly any existing model can perfectly fit the data structure, data features and objective of classification (Yu et al., 2008). Therefore, many studies have used non-parametric machine learning approaches (Lessmann et al., 2015; Li et al., 2020; Tang et al., 2019), such as the support vector machine (Huang et al., 2004), genetic programming (Ong et al., 2005), fuzzy logistics (Sohn et al., 2016), rough set (Maldonado et al., 2020; Wang et al., 2012b), and artificial neural network (Malhotra and Malhotra, 2003). These techniques, although theoretically more appealing than parametric approaches, are still criticised for their inconsistent performance on data with irrelevant features or unbalanced sample size. The

* Corresponding author.
E-mail addresses: nihe@zjgsu.edu.cn (N. He), wangyq@zjgsu.edu.cn (W. Yongqiao), jtao@263.net (J. Tao), chenzhaoyu@xyzq.com.cn (C. Zhaoyu).

https://doi.org/10.1016/j.techfore.2021.121371
Received 3 December 2020; Received in revised form 13 August 2021; Accepted 16 November 2021
Available online 29 November 2021

data available for credit rating are often known for having an excessive number of features and lower interpretability. That is, some features may be relevant[1] or deterministic only in certain periods (not always effective, as being most of the traditional econometric approaches assume). Besides, some features may be relevant when they are combined. Many data-processing approaches, such as linear combination, eigen-decomposition, and nonlinear mapping, however, can *create* some more relevant features, but the newly created features are not easily interpretable. A feasible solution is to find a *mixture of experts model* (Jordan and Jacobs, 1994) with a dynamic structure in which the outputs of the individual experts are integrated to form an overall output. Such a mixture model is essentially a committee machine that can approximate data having a mixture of various distributions, or data heavily contaminated by significant noise (Ala'raj and Abbod, 2016; Jones et al., 2015). The mixture model has, in practice, become increasingly popular for classification in many real-world applications, such as fault detection, risk alarm, etc.

Ensemble learning (Polikar, 2006) is a typical mixture model that is particularly useful when modelling is too difficult to achieve with a single algorithm. It uses multiple learning algorithms to obtain a better estimation by majority vote. Ensemble methods are meta-algorithms that may combine several machine learning or statistical algorithms into one model in order to decrease variance and/or bias (Ribeiro and dos Santos Coelho, 2020). There are two major types of ensemble learning techniques: boosting and bagging (Bootstrap Aggregating). Boosting (Zhou et al., 2021) is a sequential learning algorithm that aims to exploit the dependence between the base learners. The overall performance will very likely have less bias than with a single base learner. Bagging (Asadi and Roshan, 2021) is a parallel learning algorithm that aims to exploit the independence between the base learners since the error can be dramatically reduced by averaging and, thus, reducing the variance. In order to enhance the performance (i.e., either reduce the variance or the bias), ensemble learning requires a lot of extra computation. Such a disadvantage would be very disappointing in the case of online learning, which means that could be an infinite number of learning samples. Therefore, a compromise solution is to apply ensemble learning with less complicated base learners (i.e., Each expert model should be as simple as possible - simple structure and limited functionality), and each base learner (or expert) will act as a classifier [2] The final decision should be a compromise agreed upon by the majority of base learners. It should provide stability and increase the accuracy of the machine learning algorithm that is used in the statistical classification and regression.

Most concurrent classification/clustering techniques look for the center of distribution - that is, seek to find the most typical pattern/sample within the training data. In the case of predicting creditworthiness, within the group of creditable samples, we need a typical representative/winner, and within the group of discreditable samples, we need a typical loser. In the testing phase, we test the creditworthiness of a sample by measuring its closeness [3] to both the typical winner and loser before we can judge whether the new sample belongs to (in the case of hard partition[4]) the creditable or discreditable group, or by how much (in the case of soft partition[5]) it belongs to either group. However, for traditional clustering or classification algorithms, there are three major defects that would negatively affect the performance of credit rating.

1. When measuring the closeness, traditional approaches have to define a distance matrix that will inevitably distort the original implications of the features or make the model less interpretable. For example, in the case of Euclidean distance, we may need to take the sum of the square root of a continuous value (i.e., annual income) and a binary value (i.e., gender). The result will have to lose both of their original implications.
2. For most real market applications, especially in the case of credit rating, it is more practically realistic to find a boundary rather than a center. It is very difficult, or even impossible, to find any persistent winner or loser. The distribution of samples in a feature space is not gathered in the *middle* but along the *boundaries*. An optimal solution is not achievable, while sub-optimal or relatively optimal solutions are of more interest to people. Similar to the support vector machine (Andrew, 2001), it is more practical to construct a hyperplane or set of hyperplanes in a high dimensional feature space. Intuitively, a good classification can be achieved only by identifying suitable support vectors (i.e., samples along the boundary).
3. In the case of traditional ensemble classification, after base learner development, the ensemble stage is used to combine all the learners' output to form a final classification or prediction. Such a process could enhance the diversity among base learners but at the cost of extra computation and less accuracy. Selective ensemble (Yu et al., 2008; Zhou et al., 2010) composes a tailor-made ensemble from the pool of base learners and can maximise predictive accuracy while optimising diversity (Lessmann et al., 2015; Partalas et al., 2010).

Thus in the paper, we propose ensemble learning with a number of tree-structured base learners to form a bagging (Bootstrap Aggregating) like algorithm. In order to enhance the efficiency of the proposed bagging algorithm, we incorporate a self-organising learning strategy (Yan et al., 2020) that can guide the development (i.e., learning process) of tree-structure base learners. The self-organising learning or self-organising map is a classic clustering algorithm and has achieved many successes in fraud detection (Olszewski, 2014) and credit-risk evaluation (Wang et al., 2020). The overlapping of training subsets (i.e., Boostraping step in a typical bagging algorithm) is controlled by a topological neighbourhood network. The proposed bagging model can be trained in a more organised manner with limited randomness and can be treated as a selective homogeneous ensemble classifier. After creating a pool of developed base learners, we search the space of available base learners for a proper learner subset that enters the ensemble step. Through both artificial and real market data, we are convinced that the proposed classification model is theoretically applicable and economically sensible.

The rest of this paper is organised as follows. Section 2 reviews previous related works and describes the proposed models. Artificial data and two real market examples are used to demonstrate the proposed model in Section 3. Discussions, conclusions and future works are discussed in Section 4.

2. Methodology

In order to achieve a somewhat complicated nonlinear mapping between features (either quantitative or qualitative) that may affect creditworthiness and the predicted credit rating, we use a generalised

[1] Here we use relevant or deterministic to represent a significant correlation between the credit status of an identity and some of its features/attributes.

[2] Credit rating is more likely to be done by classification and/or clustering, which are used to assign discrete scores (creditworthiness) to different individuals or organisations. Regression is also popular in the literature, but relatively more rare than classification and clustering.

[3] Some measure that could appropriately estimate the difference between samples, such as Euclidean distance, Cosine distance, or Jaccard similarity. More appropriate measures applied in this study would be detailed in later sections.

[4] In most traditional clustering models, any sample will be assigned to only one cluster.

[5] For some clustering approaches, such as the Gaussian Mixture Model, the sample could simultaneously belong to multi-clusters, each with a membership degree.

additive model that consists of a few predefined basis functions. Such additive models are good extensions of linear models, easily interpretable while also more flexible. The idea is known as committee machine, which is a type of artificial neural network using a *divide-and-conquer* strategy. That is, the responses of multiple neurons/sub-models/experts are combined/added into a single response (Haykin, 2009). Each expert should have different parameters that reflect alterations in the data characteristics. The combination of various experts yields a mixture model with dynamic nonlinear responding regimes.

2.1. Parallel ensemble learning

Ensemble learning is a kind of machine learning technique whereby multiple base/weaker learners are trained to solve a single/same problem (Chang et al., 2018; Polikar, 2006). The generalisation ability of an ensemble is usually much stronger than that of a single model, which makes ensemble learning attractive (Yang et al., 2010). The final output of the ensemble are the aggregated solutions from all base learners by (weighted) majority vote or by the Bayes' Law (Fragoso et al., 2018).

In the case of credit rating, as we know that the data have high dimension and that the relationships among features and creditability are unstable, we obviously need a parallel ensemble rather than a sequential ensemble (Hu et al., 2019). High accuracy in predicting credit risk is neither achievable nor necessary, while relatively stable prediction is more desirable. Bagging, is therefore, not only the most appropriate choice, but is also one of the most intuitive approaches, with better performance than other ensemble learning methods (Mu et al., 2018; Wang et al., 2011). Each training data subset is used to train a different base learner of the same type. Ordinary Bagging needs bootstrapped replicas of the training data: In order to reduce the variance of combined output, the training subsets for different base learners overlap heavily. Ordinary bagging can enhance the accuracy and lower the variance, it requires significantly more computation.

Given online environments with high data dimensions, there are two major issues: ensemble size and the number of training samples in each subset. Since the purpose of ensemble learning is to achieve variance reduction at a cost of reasonable computation, the ensemble size or the number of base learners should be relatively small. As long as the result of classification is stable, the ensemble size can be as small as possible. For the number of samples in each subset, as we should assume that each base learner has a simple structure, and the number of training samples in any subset should be small. A group of samples within a small local *area* should be well classified by a simple base learner (i.e., a decision tree unit). However, a group of samples with a complicated distribution may not be well classified even by a complicated decision tree. In absence of *priori* information, we need to empirically optimise the ensemble size and number of samples while creating a good balance of computational efficiency and model performance.

In Section 2.3, we will present a self-organised bootstrap strategy from the original training data. The self-organises bagging algorithm forms an ensemble in a relatively dependent fashion. It iteratively trains the base learners to avoid too much overlapping with the current trained ensemble. For each bootstrap sample group, there would be a major responsible trained base learner as well as some minor responding base learners. That is, each piece of data, especially for those close to the boundary of different clusters, would be allocated by the proposed algorithm to different base learners for training purposes. A typical ensemble learning network, such as random forest, consists of independent component base learners (i.e., decision tree). While in this work, component base learners are not independent.

2.2. Base learner

Let $\mathbf{x} = [x_1, x_2, \ldots, x_m] \in R^m$ be an m-dimensional vector representing the features/characteristics of a borrower, and let $y \in \{1, 0\}$ be a (binary) variable that distinguishes good ($y = 0$) and bad ($y = 1$) borrowers. The credit rating is to estimate the posterior probability $p(y = 1|\mathbf{x}_i)$ - how possible it is that a default event may be happen to borrower i. The supervised learning algorithm is used to estimate a class conditional probability $p(\mathbf{x}|y)$ by using existing label information y, and then to calculate posterior probabilities using Bayes' rule. Tree-based methods sequentially partition a dataset so as to separate good and bad borrowers through a number of binary splits. The process follows from minimising indicators of node impurity, such as the Gini index or information entropy.

CART(Classification and Regression Tree) is a popular and well-developed tree-based regression and classification method (Razi and Athappilly, 2005). In a terminal node n, for an associated sub region R_n with N_n observations, we have

$$\widehat{P}_{n,k} = \frac{1}{N_n} \sum_{\mathbf{x}_i \in R_n} I(y_i = k) \tag{1}$$

as the proportion of class $k \in \{0, 1\}$ samples in node n and $I(.) = 1$ when $y_i = k$. The samples in terminal node n should thus be classified as,

$$k^* = \underset{k}{\operatorname{argmax}} \widehat{P}_{n,k} \tag{2}$$

which is the majority vote in node n. In this study, we measure the impurity of nodes by the Gini index [6], which could also be interpreted as the training error, as we expect that, ideally, a successful node should contain samples of either good or bad borrowers.

$$\sum_{(i \neq j) \in K} \widehat{P}_{n,i} \widehat{P}_{n,j} = \sum_{k \in K} \widehat{P}_{n,k} \left(1 - \widehat{P}_{n,k}\right) \tag{3}$$

The splitting variable c and split point s can be obtained by solving,

$$\underset{c,s}{\operatorname{argmin}} \left[\min \sum_{\mathbf{x}_i^L \in R_L(c,s)} \left(y_i^L - k_L^*\right)^2 + \min \sum_{\mathbf{x}_i^R \in R_R(c,s)} \left(y_i^R - k_R^*\right)^2 \right] \tag{4}$$

and the estimated parameter c, s can split the region R into binary parts R_L and R_R.

Here, CART represents a binary split tree-based learning mechanisms. One of its advantages is its good interpretability on feature perseveration (Blanco-Justicia et al., 2020). The determination of the split (c, s) parameter can be done by searching through all possible features and their values, which is actually a greedy algorithm [7] However, the optimal depth of the tree should be adaptive and reach some balance. In order to satisfy the principle of parsimony, the depth of the tree should also be limited.

2.3. Self-organised bootstrap strategy

Self-organising learning is an unsupervised learning algorithm, which means that the process of learning can be self-guided. It differs from many other artificial neural network algorithms, as it uses competitive learning other than error-correction learning (i.e., stochastic gradient descent). Another unique feature is the neighbourhood function, which could preserve the topological properties of the input. Each neuron is associated by a base learner, and the learning process will not spoil the topology induced from the map (i.e., arranged *positions* on a pre-determined lattice).

For the original self-organising map, the weights of neurons are trained using competitive learning. In this study, we train the base learners instead of weights to fit the input. At the beginning, each base

[6] Information entropy and Gini index are, in general, similar to each other. They both serve as a measure of misclassification and both are differentiable. The algorithm we proposed can actually use either of these two approaches.

[7] The problem of identifying the most suitable feature and corresponding threshold can be very difficult when the values of some features are continuous.

learner has been initialised by training with a small number of random inputs. Before more organised training, each base learner (CART) has an initial parameter setting (the tree structure and the splitting parameters of each non-terminal node) and some historical record of past training inputs. For any training input (a piece of credit record) thereafter, the algorithm will first find it the most *suitable* base learner. The past inputs of the most suitable base learner should be very *close* or *similar* to the training input. Such a base learner is defined as the best matching base learner (BML). The parameter of the BML and base learners close[8] to it would be adjusted *towards* the training input. Therefore, the issues raised here are:

1. How to measure the similarity among different training samples.
2. How to define a training input
3. How to identify the BML.
4. How to implement self-adaptive updating.

2.3.1. Similarity measurement

The similarity measure is always very important and can be debatable. Traditionally, it can be Euclidean distance, Minkowski distance, Cosine similarity, Cross entropy, etc. In the feature space, input samples should be geographically close to each other. Since features can be either categorical or continuous, and any mapping (i.e., combining a few features to form a new one) may inevitably distort the economical meanings of original features, we treat features separately. Two input samples are regarded as close to each other if and only if all of their features are similar. For example, if sample j is similar to sample i, any feature $X_{j,m}$, $m \in M$ of sample j shall satisfy,

$$\| X_j - X_i\|^m_{j\in\Omega} < \Phi_{i,m}\left(\frac{I}{N}, max(\ \|X_j - X_i\|^m_{i,j\in\Omega}\right)\right) \tag{5}$$

where $\| X_j - X_i\|^m$ represent the *distance* between samples i and j along the feature m, I is the total number of training samples i within a neighbourhood space Ω and N is the total number of base learners. $\Phi_{i,m}(.)$ generating a threshold value that can control the number of samples around the *central* sample i, and these samples can be regarded as close to sample i. The number of closing samples shall be around I/N, therefore, $\Phi_{i,m}(.)$ should be a function of I/N and the maximum distance of any two samples along feature m.

In this study, we assume that every feature should have a stable correlation with the default probability; otherwise, we will eliminate that feature. Since features are very different from each other by their natures, the *similarity* (distance) may not be defined by the norm. Alternatively, we use difference in rank $\| X_j - X_i\|^m = \| R(X_j) - R(X_i)\ \|^m$ instead of distance in Equ. 5. In worse cases, we could further weighted the difference in rank by the correlation factor $\| X_j - X_i\|^m = f_m\| R(X_j) - R(X_i)\ \|^m$.

2.3.2. Training input

Since the purpose of the proposed algorithm is to find a separator, the input block should be on the boundary, which means the input samples should belong to at least two different classes. Like the idea of the Support Vector Machine, only support vectors matter. A training input contains a group of inputs - i.e., a group of borrowers - that share some similarities in terms of a number of the features.

The procedure to form a valid training input is as follows,

1. Randomly choose a sample as the assembly point, and find its neighbour samples according to the similarity measure described above. The number of samples in a training input should be around I/N [9], where N is the pre-determined number of base learners in the initial stage.
2. Validation of input is done by testing whether its Gini index is less than an empirical threshold.

$$\sum_{k\in K} P(y_{un}=k)(1-P(y_{un}=k))\langle\Psi\left(\sum_{k\in K} P\left(y_{Ref}=k\right)\left(1-P\left(y_{Ref}=k\right)\right)\right) \tag{6}$$

where y_{un} are the labels of the group of samples belonging to a training input pending validation, and the right-hand side of Equ. 6 is a reference Gini index estimated by a random collection of samples with their labels denoted as y_{Ref}. $\Psi(.)$ generates a threshold value proportional to the reference Gini index.

Therefore, a validated training input should be a collection of samples that lie on the boundary. These samples are similar to each other in terms of their features x, but different in their labels y.

2.3.3. Best matching base learner (BML)

The best matching base learner (BML) should be the one that can produce the best performance. Therefore, for a given training input that contains a number of data samples, the output (estimated labels) of the BML should be close to the actual (true) labels of the input. Since the samples of the given input should have a different label y, for input i, we can define a probability $P(y_i = k)$ for different k. The closeness of the output label to the actual label can be,

$$E_{k\in K}(\ \|E_{i\in input}(\widehat{P_{n^*}}(y_i=k)) - E_{i\in input}(P(y_i=k))\|\ ) < \epsilon_t^o \tag{7}$$

where $E_{k\in K}(.)$ is an average across different classes, and $E_{i\in input}(.)$ is an average over samples in the training input. $\widehat{P_{n^*}}(y_i)$ is the estimation of y_i by the base learner n^*. ϵ_t is a threshold value that can be set by trial-and-error and can also vary with time t. A large ϵ, which means a large (or less pure) training input, can be useful at the beginning of model learning, and a smaller ϵ can be useful in the fine-tuning stage as the training process converges (i.e., reaches a local minimum).

Alternatively, the BML can be found by comparing features, as the features X_i of samples in a given input i should be similar to the features $E_{j\in\Delta^*}(X_j)$ of samples of the historical inputs Δ^* of the BML. Likewise, there could be another threshold to identify the BML.

$$\| X_i - E_{j\in\Delta^*}(X_j)\| < \epsilon_t^h \tag{8}$$

Again, the distance measure $\| . \|$ can be replaced by the rank or correlation-weighted rank when features are very different in their natures. The operation would be similar to the one described in the section on the *similarity measure*.

For any training input, the corresponding BML has the strongest reason to be updated towards *what the current training input wants*, while its neighbours can also be updated to certain degrees. The update process is inspired by self-organised learning.

2.3.4. Self-adaptive updating

The update formula for a neuron with reference weight w_i in a self-organising map is as follows,

$$\triangle w_i(t) = h_i(\sigma,k,t)\eta(t)\ \|\ x(t) - w_i(t)\ \| \tag{9}$$

where $\eta(t)$ is a time-adaptive learning rate, $\| x(t) - w_i(t)\ \|$ is a distance (similarity) measure between input $x(t)$ and neuron status $w_i(t)$, and

[8] Here close means short distance on the pre-determined lattice defined by the self-organising map. These base learners are neighbours of the BML.

[9] In this stage, we don't want either too many or too few data points in an input. Too many data points makes the split more difficult while too few data will slow down the process of training. Therefore, in our experiments, we take the number of samples between $0.9 * I/N$ and $1.1 * I/N$.

$h_i(\sigma, k, t)$ is the neighbourhood function with its parameter θ_i representing the topological structure of a pre-defined lattice. The value of $h_i(\sigma, k, t)$ determines the magnitude of changes that neuron i and its neighbours should make in line with $\| x(t) - w_i(t) \|$. The formula shows that the updating process of neuron weight is recursively implemented. Each input will affect a winning neuron, together with its neighbours, the number of affected neighbours is determined by the neighbour function $h_i(\sigma, k, t)$.

The traditional training of CART is not self-memorisable; that is, the estimation of its parameters is based only on concurrent input. In order to build a self-memorisable recursive learning process, we form a synthetic input using the neighbourhood function. During that training process, each base learner (CART) has a set of parameters and an input by which the CART determines its parameters. Thus, for a new input, there should be a BML whose previous input is mostly closed to that new input. The neighbor function can be used to measure how close the new input $\{X, y\}$ is to each previous input of base learner i (including the BML and its neighbors). The synthetic input can be formed by using an assemble percentage P_i, and the process is shown in Fig. 1.

$$P_i = h_i(\sigma, k, t) = k_i(t) exp\left(\frac{\| X - E_{i \in \Delta^s}(X_i) \|^2}{\sigma_i^2(t)}\right) \tag{10}$$

where $\| X - E_{i \in \Delta^s}(X_i) \|^2$ is a similarity measure between the previous input $E_{i \in \Delta^s}(X_i)$ of base learner i and the new input X, $\sigma_i(t)$ is a time-variant parameter that can control the effectiveness of the neighbourhood function - the scope of influence should decrease over time, like a standard self-organising map algorithm, and the $k_i(t)$ is also a monotonically decreasing learning rate. The assembly parameter P_i of base learner i is the most critical part of the proposed algorithm, as the self-organising learning mechanism has been solely realised by it. At the beginning, P_i used to have a relatively large value due to large $k_i(t)$. $P_i > P_j$ if input X is closer to base learner i than to base learner j. The closeness should be measured by a well defined distance, as detailed in Equ. 7. When the value of features are quantifiable and comparable, we can simply use Euclidean distance to quantify the closeness or similarity between a previous input and a new input.

Guided by such a self-organised learning pattern, the training process of each base CART within an ensemble bagging framework is no longer independent. Each CART i would be virtually placed on a pre-defined grid, and its parameters would be updated only by purposely assembling (by P_i) training inputs, especially in a later stage of training. Self-organising learning is an unsupervised approach, while the proposed self-adapted bagging is a supervised learning approach that reduces the potential abundance in the bootstrap strategy and maintains the competence of bagging CART in classifying.

2.3.5. Model pruning

The general purpose of this study is to find a more efficient ensemble learning approach when dealing with financial classification problems. In order to prevent the case of underfitting, the training process starts with an abundant number of base learners, while the *idle* learners are expected to be gradually removed. The definition of the idle learner is dependent on massive experiments. In principle, according to the self-organised bootstrap strategy, each learner should receive inputs in the training process. If any learner is dominated by its neighbours - meaning that both the learner and its close neighbours cannot receive enough training - the learner's performance on out-of-sample data will be unsatisfactory due to improper training.

Such a pruning operation ensures a relatively low possibility of the overfitting issue. Since there is very limited information on how many initial base learners there should be to start with, the number of idle learners that should be removed varies. Every pruning is a rebalancing of the whole model, so after each removal, there should be very little impact on the performance of the model as a whole. If a more sparse/pruned model can perform as well as its predecessor (before pruning), or if the degraded performance can be recovered easily, such pruning is regarded as efficient. Therefore, the model pruning can be executed as follows,

Find the idle unit At each iteration, every base learner can receive an assembly percentages P according to the closeness of the input to its previous inputs. If any base learner consecutively receives trivial $P \approx 0$ input, the base learner is considered an idle unit that is actually repeating itself during the updating process.

Attempt to remove The idle unit will be removed temporarily if it cannot receive an adequate number of inputs for the last epoch. The threshold number s can be the sum of consecutively received P in the last epoch and should be set empirically.

Validated removal The overall performance might get worse after removal of any idle base learner, but should improved after a few iterations if the previous contribution of that idle base learner can be compensated by other active base learners (i.e., its neighbours). Therefore, we implement a validation of unit removal by checking the performance recovery of the whole model. If, within next epoch, the performance can be recovered or even improved, the removal of a base learner is validated, otherwise, we keep that idle base learner.

After removal of any base learner, the SOM lattice should be revised accordingly. In this study, in order to minimise topological constraints, we use one-dimensional SOM, which also makes the lattice correction much easier. An analogy for the revision process is removing a pearl from a necklace and then reconnecting it. The neighbourhood function will also be changed accordingly.

2.3.6. Model ensemble

In the aggregation step, all possible base learners within the grid should be used when making predictions, with each learner weighted by the possibility of being the right learner. The possibility should be a posterior probability that is generated by: the prior, starting knowledge or assumptions, and the likelihood - the probability of the data given the base learner (Domingos, 2000). The idea is that the generated posterior is based on the prior, which could be the possibility of being the right base learner. In this work, the right base learner can be defined as the learner that had the best performance in the past. Self-organised

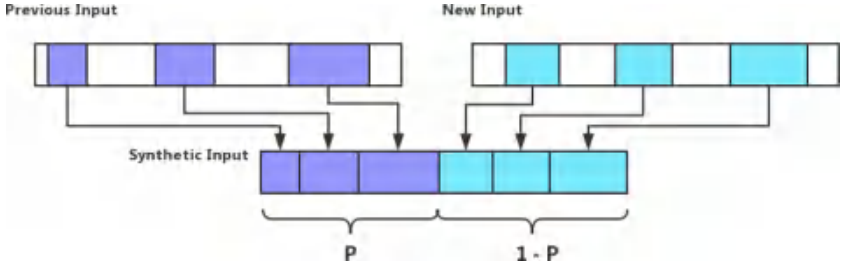

Fig. 1. A Synthetic input of a certain CART is produced by its previous input and a new input: the assemble percentage P is produced by the neighborhood function $h_i(\sigma, k, t)$.

learning makes use of the most appropriate (right) base learner, along with its neighbours. For each base learner, the inverse average Gini index on past data is used for weighting the outcome.

$$\widehat{y}(t) \quad = \frac{1}{N_\Omega}\sum_{n\in\Omega} w_n(t)\widehat{y}_n(t)w_n(t) = \frac{\sum_{i=1}^{T}\left(1/Gini_{t-i,n}\right)}{\sum_{n\in\Omega}\sum_{i=1}^{T}\left(1/Gini_{t-i,n}\right)} \tag{11}$$

where N_Ω is the total number of base learners in a neighbourhood space Ω, $\widehat{y}_n(t)$ represents the estimated label y by base learner n at time t, while $\widehat{y}(t)$ is the final estimation of the model, which is the average of responding base learners in the neighbourhood space Ω. $w_n(t)$ is the weight of base learner n at time t and is proportional to its historical performance (look back to T periods since time t). The performance is measured by purity of final split groups/ terminal leaves, so is calculated by the inverse of Gini index. Here window size T and neighbourhood space Ω are adjustable. For a converged model, T should not be sensitive, as the outputs of that base learner should be very stable, while Ω would be sensitive and should be consistent with the previous setting in training stage.

2.3.7. Performance measures

The performance of the proposed algorithm can be measured by accuracy in classification together with model complexity. The higher utility can be achieved only by high accuracy in model classification and low complexity in terms of a small number of base learners. The aim of the proposed algorithm is to create a parsimonious model with good explanatory ability which could distinguish well between entities with different credit qualities.

The accuracy is measured by the classification error. For a series of testing input samples, each sample will receive an average output from the trained model $\widehat{y}$ and has an actual label y. Intuitively, we can use Kullback-Leibler divergence $D_{KL}(y \parallel \widehat{y})$ to measure the difference between distributions of estimated $\widehat{y}$ and actual y.

$$D_{KL}(y \parallel \widehat{y}) = \sum_{i=1}^{T_w} P(y)\log\left(\frac{P(y)}{P(\widehat{y})}\right) \tag{12}$$

where $P(y_i), i = 1, 2, \ldots, T_w$ is the distribution of the actual label y in the training or testing group, while $P(\widehat{y})$ is the distribution of its estimation. A small Kullback-Leibler divergence shows better classification.

Low complexity is measured by a regulation/penalty term that can be an indicator of the number of nodes and the number of branches. The penalty equivalent is added to the ratio of the total number of splitting N_{total} to the average splitting $N_{average}$. The total splittings includes branches of every node. The minimisation objective is as follows,

$$D_{KL}(y \parallel \widehat{y}) + \lambda\frac{N_{total}}{N_{average}} \tag{13}$$

where λ is the turning factor that controls the strength of the regularisation term. Such regularisation tends to shrinks the number of splitting towards zero, but it can reduce the variance to a great extent, which will result in a better trade-off.

3. Data and experiments

We applied the proposed model to both artificial data and real market data collected from a peer-to-peer lending platform. Real market model input has various features that are closely related to the model output, which, in this study, is some credit measure such as *charge off, default, fully paid, deferred*, or *late*. The results of the proposed algorithm have been compared with those of other popular and relevant approaches, such as *Logistic Regression, Classification and Regression Trees*, and *Gradient Boosting Decision Tree*. The results show that the proposed model can provide parsimonious solutions and may outperform traditional ensemble models in terms of both modelling accuracy and computation costs.

3.1. Artificial data

In order to provide a better visual illustration, we created a two-dimensional data sets that contains **2000** data points with a non-linearly separative boundary. The data only has 2 different classes and 2 attributes (features). The proposed algorithm was trained (by randomly sampled 1600 out of 2000 data points) to identify the true boundary that splits 2 classes of data (one class in blue and the other in red). Each base learner has been limited to have a maximum of 4 terminator leaves, which means at most two splits, though more leaves can result in higher fitting accuracy in training data. We aim to build a parsimonious model with a smaller number of base learners. Out of sample testing was conducted with the rest of the 400 data points.

The initial model began with 3/4 base learners (trees) and gradually grew up to 6 base learners. After 10 independent trials, we can see the proposed algorithm can properly find a suitable network of base learners at a reasonable computational cost (i.e., number of iterations) and with a good fit either in or out of sample. The complexity of the learned ensemble models has been well controlled. The statistics of 10 trials are detailed in Table 1, and one of the learning results (highlighted in Table 1) is shown in Fig. 2. The small dots in blue and red represent the training data in 2 different classes (i.e., red class and blue class). Results in Fig. 2 are shown by the rectangles with regions marked in either red or blue. The dots *above* the areas in red/blue were predicted to be red/blue, and the dots highlighted in red/blue represent their actual classes. Different base learners have different structures; this can be seen by the fact that base learners 2 and 4 has 3 terminal leaves, and other base learners have 2 terminal leaves. The structure of the base learner should closely represent the sharp boundary separating 2 classes. Another important factor is the overlap between neighbouring base learners. For instance, the *right part* of base learner 1 is overlaps with the *left part* of base learner 2 in Fig. 2. This means that in the estimating stage, data

Table 1
Resulting statistics of repeated trials on artificial datasets .

No.	Start	Num. of terminal leaves	In-sample Accuracy	Out-of-sample Accuracy	Iterations	Over-lapping
1	3	[2 3 3 2 3 4]	94.61%	93.32%	2200	24.82%
2	3	[2 2 2 3 2 3]	95.40%	92.01%	2400	33.12%
3	3	[2 3 3 3 3 2]	91.43%	90.01%	2100	28.21%
4	3	[2 3 2 3 2 2]	94.70%	93.54%	2400	24.98%
5	3	[2 2 3 3 4 4]	93.12%	91.43%	2400	28.43%
6	4	[2 4 4 2 2 3]	94.44%	92.22%	2300	26.13%
7	4	[2 4 3 2 2 2]	95.34%	92.24%	2400	20.42%
8	4	[2 3 2 2 4 2]	94.32%	91.34%	2200	21.42%
9	4	[3 2 2 3 4 3]	93.31%	92.32%	2100	24.34%
10	4	[2 2 3 4 3 3]	94.23%	92.43%	2300	21.43%

[1] The *Start* means the number of base learners at the beginning of training and *Num. of terminal leaves* means the size of each subtree after the learning is completed, i.e., [2 2 3 4 3 3] means that there are 6 base learners with 2,2,3,4,3,3 leaves, respectively. [2] The overlapping measures indicate how many inputs have been processed by multiple base learners. During the training process, the level of overlapping has been controlled by the model parameter, while the values in this table show the resulting level of collaborations among neighbouring base learners.

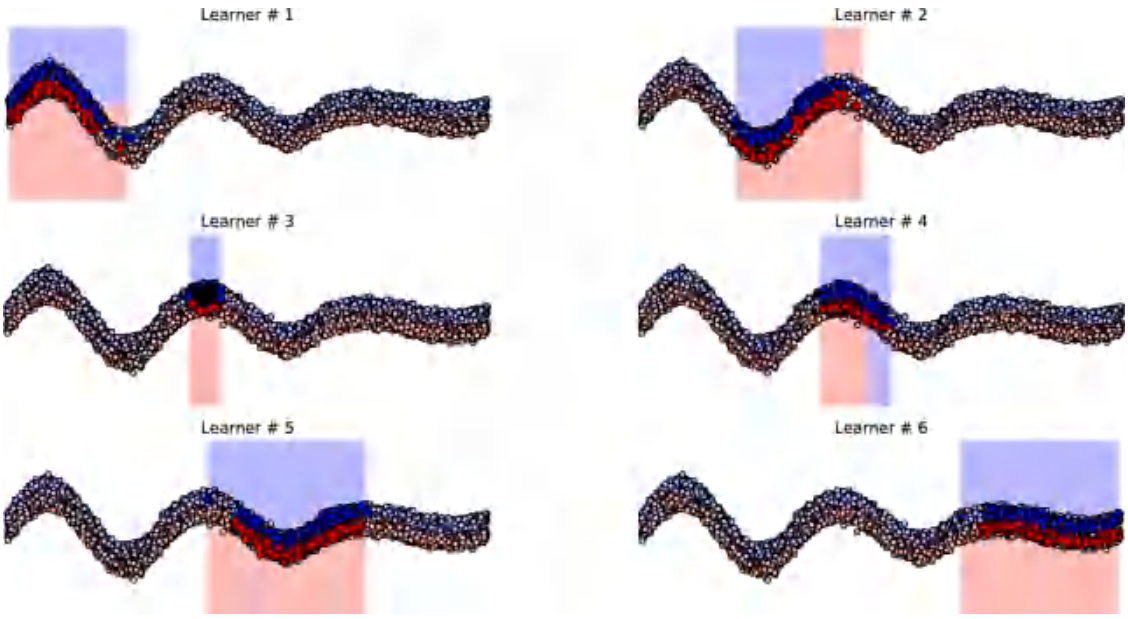

Fig. 2. The illustrations of classification capability of base learners in artificial testing data after training.

from that overlapping area should be jointly processed by both base learner 1 and base learner 2 through the ensemble process proposed in Section 2.3.6. The amount of overlap can be controlled by the adaptive learning process proposed in Section 2.3.4, and the final amount is shown in Table 1. A roughly 25% overlap at the final stage of learning shows that, for about 100 input samples, only 75 are estimated by the closest base learner, while 25 of them are jointly estimated by more than one base learners - i.e., the closest base learner with one of its neighbours in the case with artificial data.

3.2. Australian credit approval

The samples represent positive and negative instances of credit card applications. The dataset has been widely used by machine learning applications (Mekic et al., 2014) and has 690 instances with 14 attributes. Among these 14 attributes, there are 6 numerical and 8 categorical attributes. Since there is no description available for attributes due to confidentiality, we can not select features according to their economic implications. Therefore, we keep only a few attributes that have a relatively strong correlation with the class attribute (with binary values '+' or '-').

Fig. 3 shows the correlation between some instances[10] of different classes. For example, in the subplot $X1$, attribute $X1$ of data instances has values a or b, and each data instance has a label class + or - . The blue shape represents the distribution of the number of instances that take value a and b of attribute $X1$ and value + of their label class. The orange shapes are similar but take the value - of their class label. Since the blue shapes and the orange shapes represent different classes, the shape of blue distribution and orange distribution should be different. However, nearly all of the 8 categorical attributes ($X8$ is slightly different from the others) have similar distributions of both blue and orange shapes. We, therefore, choose only attribute $X8$. Similar testing is done on 6 numerical attributes. The results in Fig. 4 show that attributes $X3$,$X7$,$X10$, and $X14$ demonstrate different distributions on their + and - classes.

The proposed algorithm is again tested on all 690 instances with 5 discriminative attributes. The experiments are conducted by Random Forest, GBDT (Gradient Boosting Decision Tree), C4.5, Logistic Regression. The out of sample performance of the proposed algorithm, along with other benchmark models is shown in Table 2. We compare the out of sample performance of the proposed algorithm with the traditional models by confusion matrix, which is a popular way to visualise the performance of a supervised statistical classification model. If we divide the actual samples into the *true* group and the *false* group, and their estimations into *positive* and *negative* groups, then the accuracy of the estimation is defined by

$$\frac{TP+TN}{TP+FP+FN+TN} \tag{14}$$

Meanwhile, the True Positive Rate and True Negative Rate are defined by

$$\frac{TP}{TP+FP} \text{ and } \frac{TN}{TN+FN} \tag{15}$$

True Positive Recall and True Negative Recall are defined by

$$\frac{TP}{TP+FN} \text{ and } \frac{TN}{TN+FP} \tag{16}$$

where TP means true positive; TN means true negative; FP means false positive; FN means false negative.

The results show that the proposed self-adaptive bagging method outperformed other ensemble and traditional approaches in terms of accuracy in finding the right credit category of the testing Australian credit data. Both true positive recall and negative recall also indicate that the proposed model is slightly better than other models. C4.5 achieved the highest true positive precision in testing data. On average, the proposed approach is better for this classic classification task. The attributes are not very relevant to the credit classes, as Fig. 3 and Fig. 4 show. The quality of attributes would have a negative effect on the overall performance; therefore, we applied the model to the LendingClub data because it could provide more attributes and samples.

3.3. Lendingclub historical loan data

LendingClub is a US peer-to-peer (P2P hereafter) lending company and is the world's largest P2P lending platform. It claims that $47.24 billion in loans were originated up to the first quarter of 2019 and that it has helped more than 1.5 million borrowers since 2007. LendingClub provides complete loan data for all loans issued up to 2019 Q1. We

[10] We randomly pick 2/3 of instances for in-sample testing to show the correlation.

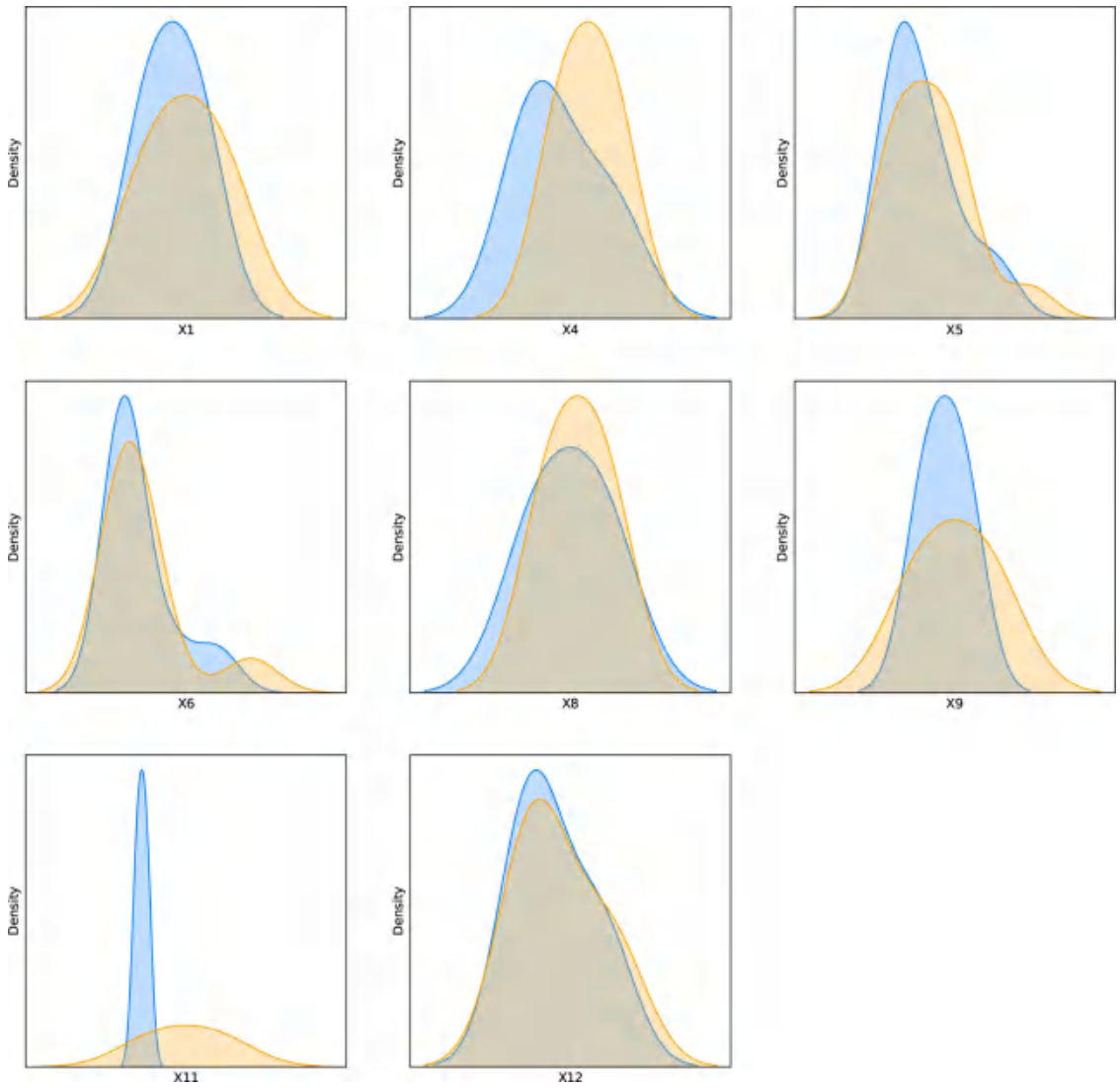

Fig. 3. The blue shape represents the distribution of the number of instances belonging to the "+" class, and orange shape represents the distribution of the number of instances belonging to the "-" class. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

collected borrowers' data up to **2016 Q3**, as there were several major circumstance changes afterwards.

- As of July 2017, LendingClub discontinued issuance of F&G grade Notes
- From September 2017, it started to price credit risk by the 5th generation credit model, which is said to be more predictive than the FICO score and is 24% better at differentiating the likelihood of a borrower charging off[11].
- Since the scandal that led to the 2016 departure of its founder and then-CEO Renaud Laplanche, many of its investors have either stopped investing in loans or have invested at reduced levels.

The data we collected include a total of 151 different attributes. We use *loan status* as the discriminative variable and focus only on the *Fully Paid* and *Charge Off* groups, which refer to those of creditworthy (good) borrowers and risky borrowers. We eliminate borrowers with other loan statuses, such as *current, default, late, issued*, and *in grace period*, as these borrowers cannot yet be judged as either safe or risky. The proposed credit model is built on the data up to **2016 Q3** and validated on the most recent data.

In order to focus on the features that are predictive in assessing credit risk (i.e., categorising safe and risky borrowers), we test all other 150 empirically customised attributes against the *loan status*. We found that 18 out of 150 attributes are relatively more correlated with the *loan status* of 2 types of borrowers, as Fig. 5 shows. The red dotted lines in Fig. 5 show some clear tendencies of correlations between the values of attributes and the loan status.

The proposed algorithm is tested on these sampled P2P data, and the results are promising. In order to make a fair comparison with other popular algorithms/models, we conduct experiments with the same data by Random Forest, GBDT(Gradient Boosting Decision Tree), C4.5, and Logistic regression. The results of classification based on 5 different approaches are shown in Table 3. We could find that the proposed algorithm is efficient, as it almost outperforms all other models when the complexity of the models is limited.

The structure of 16 base learners, as well as some of the parameters, are shown in Fig. 6. It can be seen that due to the effects of one-dimensional neighbourhood function, the structures are similar for the consecutive (i.e., neighbouring) trees but different if their topological distances become large. Meanwhile, since we impose the pruning and self-organising learning approach, the structure of base learners is not

[11] The power of data and the next generation credit model https://blog.lendingclub.com/lendingclubs-next-generation-credit-model/

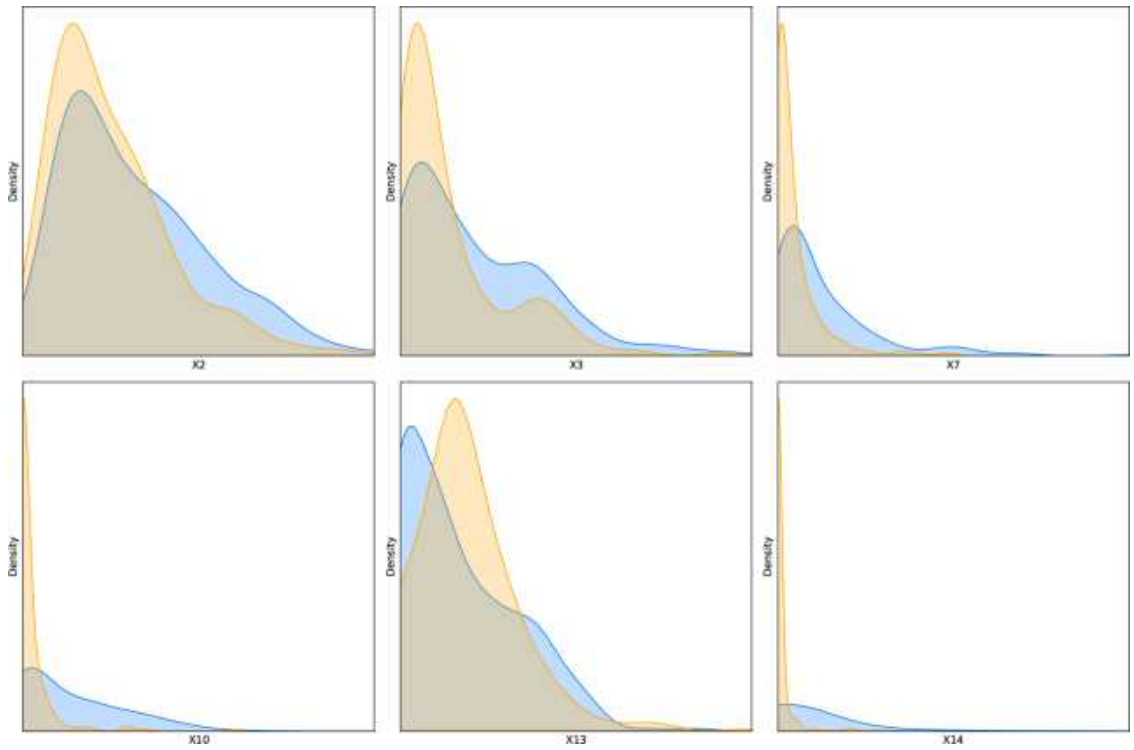

Fig. 4. The blue shape represents the distribution of the number of instances belonging to the '+. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)" class and orange shape represents the distribution of the number of instances belonging to the '-' class

Table 2
Resulting Statistics of Experiment on Australian Credit Approval Data .

	Accuracy	Precision (Pos)	Recall (Pos)	Precision (Neg)	Recall (Neg)
SomTree	**0.803**	0.796	**0.832**	**0.782**	**0.821**
GBDT	0.795	0.778	0.829	0.735	0.654
RF	0.784	0.768	0.801	0.702	0.784
C4.5	0.799	**0.802**	0.695	0.683	0.793
LR	0.793	0.768	0.799	0.773	0.814

[1] In the Table, *SomTree* refers to the proposed self-adaptive bagging method; *GBDT* refers to Gradient Boosting Decision Tree; *RF* refers to Random Forest; *C4.5* refers to Decision Tree C4.5; *LR* refers to Logistic Regression. [2] The precision rate including the true positive precision *Precision(Pos)* and true negative precision *Precision(Neg)*, together with *Accuracy* can indicates how the candidate model can find the right category for the data sample. The recall rate including the true positive recall *Recall(Pos)* and true negative recall *Recall(Neg)* indicates the sensitivity of the candidate model.

very complicated (i.e., for most boosting approaches, the trained trees tend to have a large number of layers and more leaves, which will be more likely to cause overfitting issues and, thus, deteriorate the out of sample performance.)

The results in Table 2 and Table 3 showed a slightly different level of performance for different data samples. The correlation between sample creditworthiness and the value of attributes plays an important role, though the proposed approach can be highly adaptive or customised.

The experiments on real market data demonstrate the effectiveness of the proposed model. It provides not only comprehensively good performance in credit rating, but also a flexible model that can be used to control the complexity and reduce the computational cost[12]

4. Discussion, conclusion and future works

4.1. Discussion

In this paper, a self-adaptive ensemble learning model was developed to provide an alternative for handling credit date due to its good generalisation capability. The major contribution of this work is to build a boundary detection model by integrating self-organising learning into ensemble learning (i.e., bagging).

As a supervised algorithm, compared to any *single* model, empirical ensemble learning is better at learning from training data, making predictions, and promoting larger diversity (Wang et al., 2012a). The term ensemble is reserved for methods that can search through a solution space to find a suitable one. The ensemble process is quite similar to the genetic algorithm that begins with a random population and gradually converges to some local optimal solution. The major challenge is to achieve high averaged accuracy while preventing extensive computation. As long as the final classification is less dependent on the peculiarities of any subset of training data, we can combine as few base learners as possible to achieve an interpretable outcome (Blanco-Justicia et al., 2020; Rao et al., 2020). In other words, we purposely create diversity to reduce the variance; however, too much diversity is computationally expensive and unnecessary. In fact, a small ensemble size and a small number of samples in each subset cannot be achieved simultaneously. In this study, we achieved a good trade-off between performance and efficiency by conducing excessive testing on the data.

To further enhance the ability of the classic parallel bagging

[12] The speed of the calculations among different approaches has not been recorded in this study. The efficiency in calculation is represented by the number of branches of each base learner or the overall complexity of the model structure.

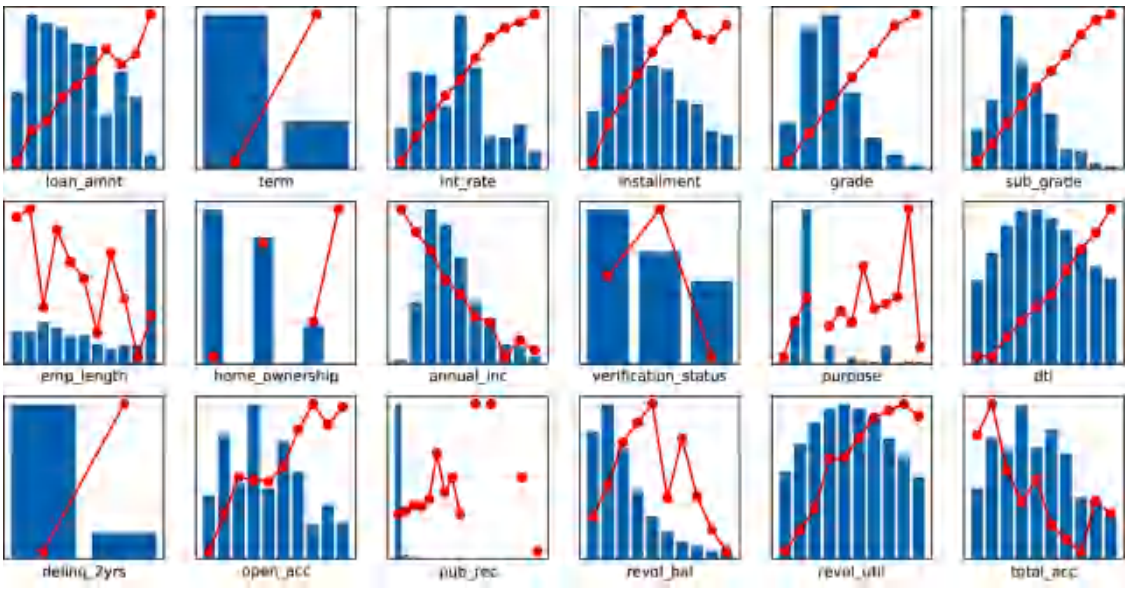

Fig. 5. Correlation between loan status and other attributes.

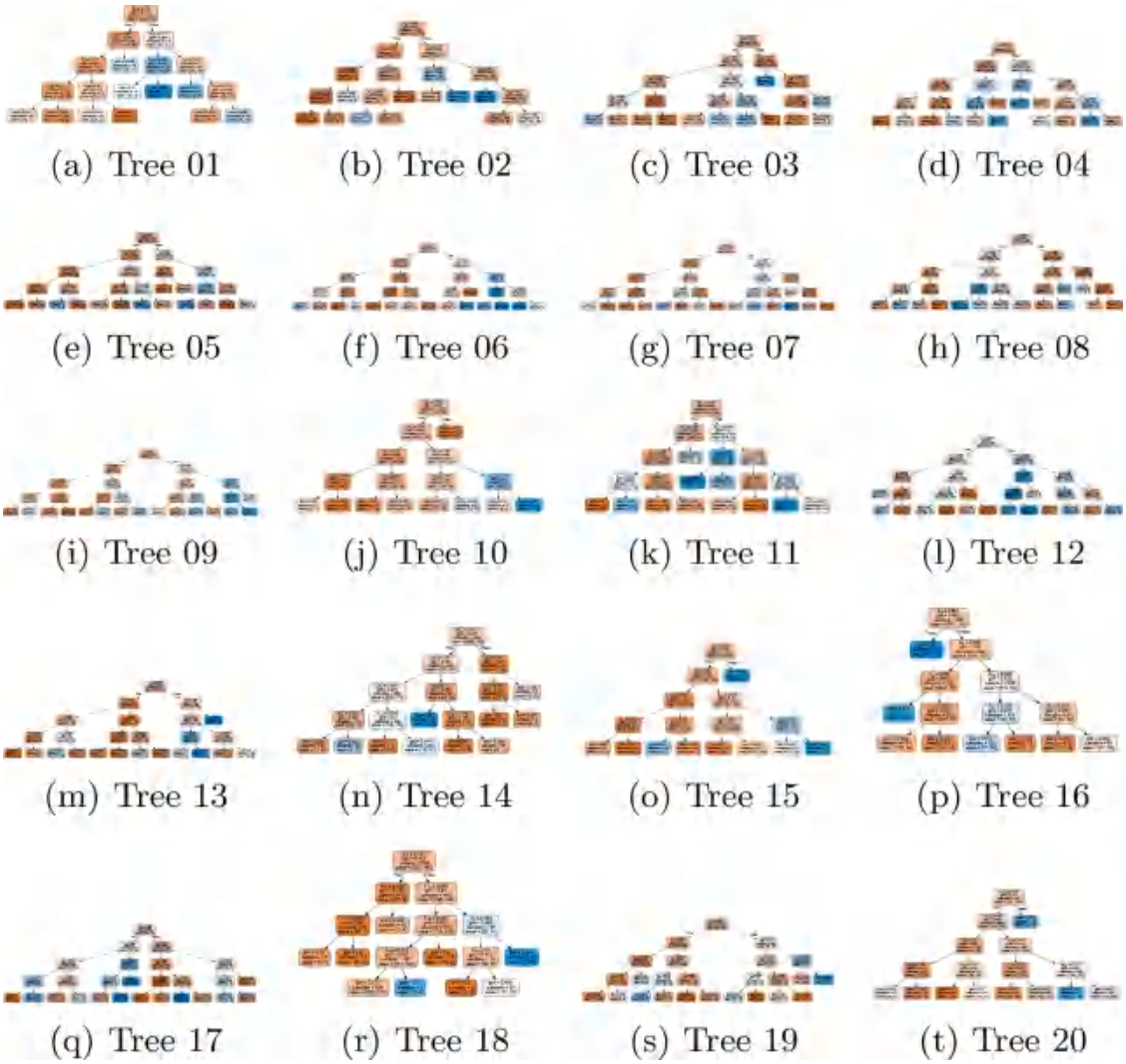

Fig. 6. Training Results.

Table 3
Resulting Statistics of Experiment on LendingClub Loan Data .

	Accuracy	Precision (Pos)	Recall (Pos)	Precision (Neg)	Recall (Neg)
SomTree	**0.804**	0.821	**0.893**	**0.796**	**0.793**
GBDT	0.800	0.786	0.847	0.741	0.684
RF	0.784	**0.897**	0.702	0.685	0.612
DT	0.796	0.801	0.876	0.699	0.579
LR	0.775	0.812	0.806	0.761	0.638

algorithm, we used self-organising learning to make bagging unparalleled. Motivated by the fact that different information is handled by different parts of the cerebral cortex in the human brain (Haykin, 1999), self-organising learning allows the different parts (i.e., neurons/base learners) of the network to process different inputs. Only a group of *relevant* base learners will participate in credit rating.

We had three challenges in apply self-organising learning to the ensemble algorithm. **(A)** Incomparable Features: As features are all relevant to creditworthiness, they become incomparable. We known that features are significantly correlated with credit, but we don't know by how much. We used the Pareto rule to develop a ranking rather than a quantified distance/difference among different entities using their features. **(B)** Find the Best Matching Learner: The estimated parameters of a base learner are highly dependent on recent input, as CART follows a unique greedy learning process in finding its optimal parameters. The reason to find the BML is to find a given input with the most appropriate base learner. The BML has been defined as the one with the closest foundation/background of its recent inputs, among others. **(C)** Define an Appropriate Neighbourhood Function: In this study, we changed the original one off learning to a self-memorisable iterative process, and we used the neighbourhood function to form a synthetic input. That is, we trained the base learners with selected inputs. The selection process is defined by a neighbourhood function. Such a combination of self-organising learning and a bagging algorithm reduced the potential abundance in the bootstrap strategy and maintained the competence of the bagging CART in classifying.

The proposed model has two validation steps. Pruning is used as a penalty for model complexity. Pruning occurs only during the later stage of training, especially when some idle units have been dominated by their powerful neighbours. Such situations might occur when neighbourhood function is too weak to activate such idle units. The other validation is actually an early stop mechanism of the learning process. It can stop the learning process from going too far or from getting trapped into some local suboptimal. Although the self-organising learning can avoid local minima, further control can improve comprehensive performance of the model.

4.2. Conclusion and future works

Credit rating has been one of the most important application areas of classification and various algorithms such as traditional statistical models and non-parametric methods based on machine learning, have been developed for better credit approval processes. The common assumption about balanced proportions among different kinds of data, as well as the i.i.d assumption about data distribution, are always violated in credit-rating data. Traditional statistical models often have limitations in handling credit data due to those strong model assumptions.

Since the decision tree is easily understandable and interpretable by business people (Sadatrasoul et al., 2013), we connect a number of base learners (classification and regression tree) by a self-organised bagging ensemble. The experimental results in the paper shows that the proposed algorithm can obtain better classification accuracy than other traditional approaches.

Future studies should aim at developing more efficient and cost-sensitive learning methods by improving the selection of interpretive features and the sampling approaches that could partially overcome the class imbalance problem. Other relevant variables, such as customer behaviour, should also be investigated to improve the credit rating accuracies in future studies. In this study, we have assumed that all features are relevant and that the sample size is more than adequate. However, we did encounter credit rating problems with a small number of entities but a large number of features. In order to build up a more generalised algorithm, the model needs to be more adaptable in feature selection, as well as in base learner tuning. Regarding performance measurement, the KullbackLeibler Divergence may not be the most appropriate measure. The symmetric Jensen-Shannon Divergence or the KolmogorovSmirnov test may better be used to assist supervised ensemble learning.

CRediT authorship contribution statement

Ni He: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft. **Wang Yongqiao:** Investigation, Validation. **Jiang Tao:** Funding acquisition, Writing – review & editing. **Chen Zhaoyu:** Data curation, Software.

CRediT authorship contribution statement

Ni He: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft. **Wang Yongqiao:** Investigation, Validation. **Jiang Tao:** Funding acquisition, Writing – review & editing. **Chen Zhaoyu:** Data curation, Software.

Acknowledgements

This research was supported by the Natural Science Foundation of Zhejiang Province under Grant Number (LY19G010001), the National Natural Science Foundation of China under Grant Number (71671166), the School of Finance and the Tailong Finance School of Zhejiang Gongshang University. Meanwhile, the authors would like to thank the data collection services provided by Hangzhou Woma Info. Tech. Co. Ltd.

References

Ala'raj, M., Abbod, M.F., 2016. Classifiers consensus system approach for credit scoring. Knowledge-based systems 104, 89–105.
Andrew, A.M., 2001. An introduction to support vector machines and other kernel-based learning methods. Kybernetes 32, 1–28.
Asadi, S., Roshan, S.E., 2021. A bi-objective optimization method to produce a near-optimal number of classifiers and increase diversity in bagging. Knowledge-based systems 213, 106656.
Blanco-Justicia, A., Domingo-Ferrer, J., Martanez, S., Scnchez, D., 2020. Machine learning explainability via microaggregation and shallow decision trees. Knowledge-based systems 194, 105532.
Chang, Y.-C., Chang, K.-H., Wu, G.-J., 2018. Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. Applied soft computing 73, 914–920.
Domingos, P., 2000. Bayesian averaging of classifiers and the overfitting problem. ICML2000: Proceedings of the Seventeenth International Conference on Machine Learning.
Fragoso, T.M., Bertoli, W., Louzada, F., 2018. Bayesian model averaging: a systematic review and conceptual classification. International statistical review 86, 1–28.
Haykin, S., 1999. Neural Networks: A Comprehensive Foundation (3rd Edition). Prentice-Hall.
Haykin, S., 2009. Neural Networks and Learning Machines. Pearson Prentice Hall.
Hu, X., Huang, H., Pan, Z., Shi, J., 2019. Information asymmetry and credit rating: aquasi-natural experiment from china. Journal of Banking & Finance 106, 132–152.
Huang, C.L., Chen, M.C., Wang, C.J., 2007. Credit scoring with a data mining approach based on support vector machines. Expert systems with applications 33 (4), 847–856.
Huang, Z., Chen, H., Hsu, C.J., Chen, W.H., Wu, S., 2004. Credit rating analysis with support vector machines and neural networks: a market comparative study. Decision support systems 37, 543–558.
Jiang, X., Packer, F., 2019. Credit ratings of chinese firms by domestic and global agencies: assessing the determinants and impact. Journal of Banking & Finance 105, 178–193.

Jones, S., Johnstone, D., Wilson, R., 2015. An empirical evaluation of the performance of binary classifiers in the prediction of credit ratings changes. Journal of Banking & Finance 56, 72–85.
Jordan, M.J., Jacobs, R.A., 1994. Hierarchical mixtures of experts and the EM algorithm. Neural computation 6, 181–214.
Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C., et al., 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. European journal of operational research 247, 124–136.
Li, J.-P., Mirza, N., Rahat, B., Xiong, D., 2020. Machine learning and credit ratings prediction in the age of fourth industrial revolution. Technological forecasting and social change 161, 120309.
Maldonado, S., Peters, G., Weber, R., 2020. Credit scoring using three-way decisions with probabilistic rough sets. Information sciences 507, 700–714.
Malhotra, R., Malhotra, D.K., 2003. Evaluating consumer loans using neural networks. Omega 31 (2), 83–96.
Mekic, E., Mekic, E., Ilgun, E., 2014. Application of ANN in australian credit card approval. European Researcher 69, 334–342.
Mu, Y., Liu, X., Wang, L., 2018. A pearson's correlation coefficient based decision tree and its parallel implementation. Information sciences 435, 40–58.
Olszewski, D., 2014. Fraud detection using self-organizing map visualizing the user profiles. Knowledge-based systems 70, 324–334.
Ong, C.S., Huang, J.J., Tzeng, G.H., 2005. Building credit scoring models using genetic programming. Expert systems with applications 29, 41–47.
Partalas, I., Tsoumakas, G., Vlahavas, I., 2010. An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. Machine learning 81, 257–269.
Polikar, R., 2006. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 6, 21–45.
Rao, C., Liu, M., Goh, M., Wen, J., 2020. 2-Stage modified random forest model for credit risk assessment of p2p network lending to ×three ruralsØ borrowers. Applied soft computing 95, 106570.
Razi, M.A., Athappilly, K., 2005. A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. Expert systems with applications 29 (1), 65–74.
Ribeiro, M.H.D.M., dos Santos Coelho, L., 2020. Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. Applied soft computing 86, 105837.
Sadatrasoul, S., Gholamian, M.R., Siami, M., Hajimohammadi, Z., 2013. Credit scoring in banks and financial institutions via data mining techniques: a literature review. Journal of AI and Data Mining 1 (2), 119–129.
Sohn, S.Y., Kim, D.H., Yoon, J.H., 2016. Technology credit scoring model with fuzzy logistic regression. Applied soft computing 43, 150–158.
Tang, L., Cai, F., Ouyang, Y., 2019. Applying a nonparametric random forest algorithm to assess the credit risk of the energy industry in china. Technological forecasting and social change 144, 563–572.
Wang, G., Hao, J., Ma, J., Jiang, H., 2011. A comparative assessment of ensemble learning for credit scoring. Expert systems with applications 38 (1), 223–230.
Wang, G., Ma, J., Huang, L., Xu, K., 2012. Two credit scoring models based on dual strategy ensemble trees. Knowledge-based systems 26, 61–68.
Wang, J., Hedar, A.R., Wang, S., Ma, J., 2012. Rough set and scatter search metaheuristic based feature selection for credit scoring. Expert systems with applications 39 (6), 6123–6128.
Wang, L., Chen, Y., Jiang, H., Yao, J., 2020. Imbalanced credit risk evaluation based on multiple sampling, multiple kernel fuzzy self-organizing map and local accuracy ensemble. Applied soft computing 91, 106262.
Yan, J., Liu, J., Tseng, F.-M., 2020. An evaluation system based on the self-organizing system framework of smart cities: a case study of smart transportation systems in china. Technological forecasting and social change 153, 119371.
Yang, P., Hwa Yang, Y., Bing, B.Z., Albert, Y.Z., 2010. A review of ensemble methods in bioinformatics. Current bioinformatics 5, 296–308.
Yu, L., Wang, S., Lai, K.K., 2008. Credit risk assessment with a multistage neural network ensemble learning approach. Expert systems with applications 34 (2), 1434–1444.
Zhou, L., Fujita, H., Ding, H., Ma, R., 2021. Credit risk modeling on data with two timestamps in peer-to-peer lending by gradient boosting. Applied soft computing 110, 107672.
Zhou, L., Lai, K.K., Yu, L., 2010. Least squares support vector machines ensemble models for credit scoring. Expert systems with applications 37 (1), 127–133.

He Ni received his PhD degree in Electrical Engineering and Electronics from the University of Manchester, England. He is a professor in School of Finance Zhejiang Gongshang University. His research interests include data mining and machine learning and has published more than 10 peer-reviewed journal papers including Neurocomputing, Applied Soft Computing, International Journal of Technology Management.

Yongqiao Wang received the Ph.D. degree in Management Sciences and Engineering from Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS), Beijing in 2005. He is currently a full professor in School of Finance, Zhejiang Gongshang University. His research interests include machine learning, data mining and financial risk management. He has co-authored more than 10 papers in journals including European Journal of Operational Research, IEEE Transactions on Neural Networks and Learning Systems and IEEE Transactions on Fuzzy Systems.

Tao Jiang received his PhD degree in probability theory and mathematical statistics from University of Science and Technology of China. He is currently the president of Zhejiang Gongshang University Hangzhou College of Commerce, a professor in School of statistics and Mathematics of Zhejiang Gongshang University. His main research fields include quantitative risk management, stochastic model research in actuarial science, statistical inference in biomedical field, and design of randomized trials, and has published more than 50 peer-reviewed journals papers including Insurance: Mathematics and Economics, Statistics in Medicine, Science China and Journal of Computational and Applied Mathematics.

Zhaoyu Chen received his MSc degree in finance from School of Finance in Zhejiang Gongshang University. He is currently a security analyst in Industrial Securities Co., Ltd. and his research interests include machine learning techniques in financial risk management.