Available Online: 18 April 2025 DOI: 10.54254/2977-5701/2025.22021

A study on the credit risk of commercial banks based on intelligent optimization algorithms to modify the KMV model

Chaojie Fan

School of Economics and Management, Southeast University, Nanjing, China

15365129798@163.com

Abstract. Credit risk is one of the main risks faced by commercial banks. Credit risk management includes risk identification, assessment, and early warning, among which risk assessment is fundamental and key. Currently, research on credit risk assessment in China is still in its developing stage, and the precision of measuring credit risk needs improvement. Among various evaluation methods, the Kealhofer, McQuown, and Vasicek model (KMV model) has shown good practical application and is relatively suitable for the national conditions of China. However, it still has some flaws. To address the issue of insufficient external validity in the default point parameter settings of the KMV model, the Particle Swarm Optimization (PSO) algorithm is used to optimize these parameters, and the Particle Swarm Optimization-Grey Wolf Optimization (PSO-GWO) algorithm is integrated to construct the Adaptive Particle Swarm Optimization-KMV Model (APSO-KMV model) and the PSO-GWO-KMV model. Based on an empirical study comparing real data from 5,234 companies, it was found that the original KMV model had an Area Under the Curve (AUC) value of 0.7362, accuracy of 0.2610, and binary cross-entropy loss of 0.7006; the PSO-KMV model had a short-term debt coefficient α of 0.0496, a long-term debt coefficient β of 0.2508, an AUC value of 0.9994, accuracy of 0.9996, and binary cross-entropy loss of 4.1990; the PSO-GWO-KMV model had a coefficient α of 0.0496 and a value β of 0.2690, an AUC value of 0.9987, accuracy of 0.7603, and binary cross-entropy loss of 4.0804. The optimized KMV model showed a significant improvement in predictive accuracy.

Keywords: commercial banks, credit risk, KMV model, intelligent optimization algorithms

1. Introduction

Credit risk is one of the oldest and most far-reaching financial risks. In modern society, as the business of commercial banks evolves, credit risk in commercial banks is divided into narrow and broad senses. Broadly, credit risk includes the credit risk of commercial banks, investment risk, and the risks associated with the commercial banks themselves [1]. Narrowly, credit risk refers solely to credit risk. This paper focuses on this type of credit risk. Credit risk management is the core business of commercial banks, including risk identification, assessment, and management. Identification is the preliminary work of assessment, while management is the subsequent result of assessment. Credit risk assessment is the core and key of credit risk management. To effectively study commercial bank credit risk, it is crucial to conduct research on credit risk assessment.

Scholars both domestically and internationally have made substantial progress in the study of credit risk assessment methods. From the early expert system method based on empirical judgment and the analytic hierarchy process to statistical analysis-based models such as the Z-score model, Logistic model, and KMV model, and then to the use of big data era models like Support Vector Machine (SVM), neural networks, and XGBoost models, there is a clear trend of moving from subjective to objective, and from empirical to intelligent methods. Based on existing research, this paper selects the KMV model as the main evaluation model and considers applying intelligent optimization algorithms to optimize the default point parameter setting of this model. The paper first introduces and reviews relevant theoretical methods, including the KMV model and the two main intelligent optimization algorithms, PSO and GWO. It then analyzes the strengths and weaknesses of each model and algorithm and, based on this, conducts fusion experiments to combine the strengths of the methods. Using the GWO algorithm as an example, the integrated PSO-GWO algorithm shows a significant improvement in search efficiency and convergence speed. Finally, the paper applies the optimized algorithm to modify the KMV model and optimizes its parameter settings.

2. Literature review

Currently, both academic and industry circles have developed a large number of credit risk assessment methods, which can be classified according to the models used [2]. Summarizing the relevant literature, credit risk assessment models can be divided into the following four categories: classical models, statistical models, modern models, and artificial intelligence models.

Classical models are essentially scoring systems based on subjective judgment by experts, including expert judgment methods represented by the 5C method, credit scoring methods, and the five-level loan classification method commonly adopted by Chinese commercial banks [3]. These evaluation models are highly subjective and require a significant number of experts. Moreover, due to the late start of private enterprises in China, factors such as quality, capital, and collateral in the 5C method differ from those of large foreign enterprises, which may lead to misjudgments in the scoring system.

Statistical models are based on multivariate statistical analysis, with the basic idea of categorizing and summarizing the patterns in historical samples and establishing relevant discriminant formulas to classify new samples [4]. Scholars in foreign countries began research in this area earlier. Beaver selected 30 financial ratio indicators to establish a univariate decision model and predicted financial crises for 158 companies [5]. Ohlson used Logit regression analysis to study bankruptcy enterprise samples, distinguishing three types of predictor variables: positive, negative, and indeterminate, to analyze the bankruptcy enterprise samples [6]. In China, research on statistical methods for financial risk warning models began in the late 1990s. Chen Jing used Beaver and Altman’s statistical prediction models for empirical analysis of ST companies in the domestic securities market and found that the decision model composed of multiple financial ratios such as asset-liability ratio and current ratio had a predictive capability up to three years in advance [7]. Statistical models have good explanatory power and applicability in corporate default prediction, but due to strict mathematical assumptions, real-world data often fail to meet the requirements. In addition, the construction of the indicator system and the selection of the model’s applicability still require further research and optimization.

In the corporate world, as the financial system becomes increasingly complex, many modern models for credit risk evaluation have gradually emerged. In 1997, J.P. Morgan introduced the CreditMetrics model, aiming to provide financial institutions with a comprehensive framework for evaluating and managing credit risk. Building on this, in 2004, Mark Kealhofer and Richard McQuown proposed the KMV model to address the limitations of traditional credit scoring models in handling extreme credit risk events (such as defaults). The KMV model predicts the default probability by calculating a company’s default point using its market value and stock price volatility. Subsequently, the Basel Committee on Banking Supervision, Derbali and Hallara [8], among others, conducted in-depth comparative analyses on the accuracy, effectiveness, and applicability of several modern credit risk measurement models such as CreditMetrics, CreditRisk+, and KMV, enhancing the objectivity and dynamics of these models, making them capable of reflecting the real market conditions of most countries, and further aligning with the regulatory requirements of various countries.

With the development of information technology, a large number of artificial intelligence models have been applied to risk assessment and management optimization. Currently, widely used models include Support Vector Machines (SVM) and Backpropagation (BP) neural networks. Support vector machines, a supervised learning algorithm first proposed by Vapnik and Chervonenkis in 1963, were further developed in the 1990s. Compared to traditional statistical methods, SVM does not rely on prior knowledge of the problem, offering excellent generalization ability [9]. Liu Min and Lin Chengde established a commercial bank credit risk assessment model based on the general learning algorithm SVM, confirming the effectiveness and superiority of this method for risk assessment [10]. BP neural networks, a type of multilayer feedforward neural network, are primarily used to construct credit scoring models to predict the probability of borrower default. Li and Chen used the ratio of listed companies failing to repay loans on time as a measure of credit risk, combining independent sample t-tests and principal component analysis to construct a commercial bank credit risk identification model based on BP neural network technology. Empirical results showed that the BP neural network model has strong identification capability for commercial bank credit risk and can achieve compatibility between memory and generalization abilities [11].

Overall, classical models are relatively subjective, and the related research has been quite extensive. Machine learning methods in statistical and artificial intelligence models often face the dilemma of “choosing between interpretability and precision.” Among modern models, although the KMV model still has some flaws, it is relatively well-suited to the national conditions of China. With the gradual improvement of the equity trading system in China, the stock prices of enterprises can more accurately reflect their operational status. As listed companies in China’s banking industry, nationwide joint-stock commercial banks have relatively transparent and publicly available information. The data required by the KMV model is easily accessible, and the model is simple to operate and practical. Therefore, China is basically equipped with the practical application conditions for the KMV model, and it can be well-suited for evaluating the credit risk of nationwide joint-stock commercial banks. It is worth noting that, although the KMV model can output the distance to default, due to the small sample size of defaults in China, the model cannot output the default probability for bonds.

The development of intelligent optimization algorithms has provided new ideas for the credit risk assessment of commercial banks. Intelligent optimization algorithms are a class of optimization methods that simulate the operating mechanisms of human intelligence, biological groups, and natural phenomena. The numerous adaptive optimization phenomena in nature inspire strategies for solving complex optimization problems across various disciplines such as management science, computer science, and economics. With a wide range of types and flexibility, intelligent optimization algorithms have significant potential in the field of credit risk assessment. Therefore, selecting relevant intelligent optimization algorithms to modify the KMV model to more accurately measure the credit risk of China's commercial banks is a feasible and effective approach for better measuring the credit risk of China's commercial banks.

In recent years, the research direction of using intelligent optimization algorithms to modify the KMV model for better credit risk measurement has gained significant attention both domestically and internationally. Lee used the Genetic Algorithm (GA) to encode variables and redefine the optimal default point of the KMV model. The results showed that the GA-KMV model had a higher accuracy than the KMV model, improving the default prediction performance [12]. Zhang et al. applied the Particle Swarm Optimization (PSO) algorithm, Maximum Likelihood Estimation (MLE) method, and Fuzzy Clustering (FC) to modify the KMV model, obtaining the optimal default point and effective market price for non-tradable shares using the PSO-KMV model, and using FC to classify liabilities into different clusters. This hybrid KMV model significantly improved the performance of the KMV model [13]. Ye constructed a GWO-KMV-XGBoost hybrid model, using the traditional KMV credit evaluation model as the main framework, adding GWO and XGBoost models, and applying it to the credit bond evaluation of listed companies in China. The use of these three models effectively solved the problems of few default samples, sample imbalance, and prediction accuracy, significantly improving the model's prediction accuracy and enhancing its interpretability [14].

3. Model construction

3.1. KMV model

The KMV model is developed based on the Black-Scholes (BS) option pricing theory. The core idea is that a company's default probability depends on the difference between its liabilities and asset value. The basic concept is that if, at the maturity of the debt, the value of liabilities exceeds the value of assets, the company is considered to have defaulted. The default distance is used to represent how far the company's asset value is from the default point. The derivation process consists of three steps: estimating the company's asset value V, asset volatility σ_V, and default point DP; calculating the default distance DD; and outputting the default probability EDF. Firstly, the asset value V and volatility σ_V can be solved using the following system of Equation (1):

$$\begin{cases} E_T = V_T\Phi(d_1) - De^{-rT}\Phi(d_2) \\ \sigma_E E_T = \Phi(d_1)V_T\sigma_V \end{cases} \tag{1}$$

Where T is the debt maturity date, and $E_T = max(0, V_T - D_T)$ is the company's equity value. E_T is treated as a call option. According to the BS formula, it is assumed that the company's asset value V_T follows a geometric Brownian motion, i.e., $dV_T = \mu V_T dT - \sigma_V dW_T$, where μ represents the expected asset growth rate, and W_T represents the volatility. By combining the BS formula, the company's equity value satisfies: $E_T = V_T\Phi(d_1) - De^{-rT}\Phi(d_2)$. Where $d_1 = \frac{ln\left(\frac{V_T}{D}\right)+\left(r+\frac{\sigma^2 V}{2}\right)T}{\sigma_V\sqrt{T}}$ and $d_2 = d_1 - \sigma_V\sqrt{T}$ are calculated based on the BS formula, r is the cumulative standard normal distribution, and $\Phi(X)$ represents the debt value, which is the amount of debt due at maturity. According to Itô's Lemma, the company's stock price E_T also follows a geometric Brownian motion, i.e., $\sigma_E E_T = \frac{\partial E_T}{\partial V_T}V_T\sigma_V$, where $\frac{\partial E_T}{\partial V_T} = \Phi(d_1)$. Both stock prices E_T and their volatility σ_E are publicly available data.

Regarding the calculation of the default point, based on extensive case analysis, KMV assumes that the default point consists of the company's short-term debt (with a maturity of less than one year) plus half of its long-term debt, i.e., $DP = STD + 0.5LTD$.

Next, the default distance is defined as the distance between the company's asset value and the default point, and it is standardized as follows: $DD = \frac{V-DP}{V\sigma_V}$.

Finally, by assuming that the company's asset market value follows a normal distribution, the default probability can be estimated as Equation (2):

$$EDF = P[E(V) \leq DP] = \Phi\left(-\frac{V - DP}{V\sigma_V}\right) = \Phi(-DD) \tag{2}$$

Clearly, the default distance and the expected default probability are negatively correlated. The KMV model links the asset value to the stock price, enabling it to reasonably predict future risks. As a dynamic model, it can provide real-time updates of the company's default probability. However, it is important to note that the default point parameters in the KMV model, specifically the coefficients for short-term and long-term debt, are calculated based on foreign data. Therefore, the external validity of these parameters should be questioned. Using intelligent optimization algorithms to adjust these parameters is a reasonable and feasible approach.

3.2. Introduction to intelligent optimization algorithms

3.2.1. PSO and APSO algorithms

Inspired by the intelligent behavior of bird flocks, in 1995, American scholars Eberhart and Kennedy proposed the Particle Swarm Optimization (PSO) algorithm [15]. During the foraging process, each bird searches in a direction it has determined and, while searching, records and shares the best foraging locations it has found with the flock. By combining individual memory with shared experiences from the flock, the birds eventually find the position in the forest with the most food, which corresponds to the global optimum solution for the problem at hand. Based on biological principles, the PSO algorithm is expressed as follows:

Let the position of the i th particle be denoted as $X_{id} = (x_{i1}, x_{i1}, \dots, x_{iD})$, the velocity of the i th particle as $V_{id} = (v_{i1}, v_{i1}, \dots, v_{iD})$, the individual best solution of the ith particle as $P_{id,pbest} = (p_{i1}, p_{i1}, \dots, p_{iD})$, the global best solution found by the swarm as $P_{d,gbest} = (p_{1,gbest}, p_{2,gbest}, \dots, p_{3,gbest})$, the fitness value of the best position found by the ith particle as f_p, and the fitness value of the best position found by the swarm as f_g. The velocity update formula is then given by: $v_{id}^{t+1} = wv_{id}^t + c_1r_1\left(p_{id,pbest}^t - x_{id}^t\right) + c_2r_2\left(p_{d,gbest}^t - x_{id}^t\right)$. The position update formula is: $x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$. The velocity update formula consists of three components. The first is the inertia part wv_{id}^t, which is composed of the inertia weight and the particle′s own velocity, representing the particle′s trust in its previous state of motion. The second is the cognitive part $c_1r_1\left(p_{id,pbest}^t - x_{id}^t\right)$, which represents the particle's own thinking, i.e., the distance and direction between the particle's current position and its individual best position. The third is the social part $c_2r_2\left(p_{d,gbest}^t - x_{id}^t\right)$, which represents the information sharing and cooperation between particles, i.e., the experience from other superior particles in the swarm, and can be understood as the distance and direction between the particle's current position and the swarm's best position.

To make the PSO algorithm achieve a balance between seeking the global optimum and the local optimum as much as possible, and to maintain a balance between convergence speed and search effectiveness, the key lies in optimizing parameter settings, including the inertia weight w and learning factors c_1 and c_2.

The inertia weight w represents the influence of the velocity of the previous generation of particles on the velocity of the current particles, or in other words, the particle's trust in its current state of motion. The larger the value of w, the stronger the ability to explore new areas and the better the global search ability, but the weaker the local search ability, and vice versa. In solving practical optimization problems, it is often desirable to first perform a global search to rapidly converge the search space to a certain area, then use a local fine search to obtain a high-precision solution. To address this need, an adaptive adjustment strategy can be introduced, where the value of w is linearly reduced as the iterations progress. The learning factor, also known as the acceleration coefficient or acceleration factor, is represented by c_1 and c_2. c_1 represents the weight of the particle′s next action coming from its own experience, which accelerates the particle towards its individual best position $P_{id,pbest}$. c_2 represents the weight of the particle's next action coming from the experience of other particles, which accelerates the particle towards the swarm's global best position $p_{d,gbest}^t$. When $c_1 = 0$, PSO degenerates into a selfish particle swarm algorithm, losing diversity within the swarm, and is prone to getting stuck in a local optimum. When $c_2 = 0$, PSO degenerates into a self-cognitive particle swarm algorithm with no information sharing, leading to slow convergence. When neither c_1 nor c_2 equals 0, it becomes a fully functional particle swarm algorithm, which is better at maintaining a balance between convergence speed and search effectiveness and is the optimal choice. The adaptive particle swarm optimization algorithm (APSO) with optimized parameter settings is expressed as Equation (3), (4) and (5):

$$v_{id}^{t+1} = wv_{id}^t + c_1r_1\left(p_{id,pbest}^t - x_{id}^t\right) + c_2r_2\left(p_{d,gbest}^t - x_{id}^t\right), c_1 \neq 0, c_2 \neq 0 \tag{3}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{4}$$

$$w = w_{max} - (w_{max} - w_{min})\frac{iter}{iter_{max}} \tag{5}$$

Where w_{max} is the maximum inertia weight, w_{min} is the minimum inertia weight. $iter$ represents the current iteration number, and $iter_{max}$ represents the maximum number of iterations.

3.2.2. Grey wolf optimization algorithm

The Grey Wolf Optimization (GWO) algorithm was first introduced in 2014 [16], achieving a balance of accuracy and speed by simulating the collective hunting behavior of grey wolves. In the design of the GWO algorithm, an effective solution corresponds to an individual grey wolf in the pack, and the degree of the solution's fitness corresponds to the wolf's rank in the social hierarchy, which ranges from $\alpha, \beta, \delta, \omega$; the optimal solution is considered as α. In the GWO algorithm, the hunting process is guided by the first three ranks, with ω wolves following these three types of wolves. Similar to the wolf pack hunting process, the solving process in GWO includes encirclement, hunting, and attack.

In the encirclement process, let t represent the iteration number, the position vector of the prey be $\vec{X}_p$, and the position vector of the grey wolf be $\vec{X}$, then the distance between them is $\vec{D} = \vec{C} \cdot \vec{X}_P(t) - \vec{X}(t)$, and the wolf's position is updated as $\vec{X}(t+1) = \vec{X}_P(t) - \vec{A} \cdot \vec{D}$. Here, $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \vec{C} = 2\vec{r}_2$ and $\vec{r}_1$ and $\vec{r}_2$ are taken from random numbers within the range [0,1]. The randomness plays a crucial role in avoiding local optima, especially in the final iterations where the global optimal solution is sought. The convergence factor $\vec{a}$ is defined as $\vec{a} = 2 - \frac{2t}{T_{max}}$, where T_{max} is the maximum number of iterations.

Once the prey is encircled, the α wolf leads the pack in guiding the hunting process. The prey's position is actually the optimal solution, and the grey wolves track and approach the prey in the hunting process to arrive at the optimal solution. Assuming that α, β, δ wolves have better knowledge of the prey's potential location, other grey wolves update their positions based on the optimal wolves, gradually getting closer to the prey. The distance between the remaining grey wolves and α, β, δ wolves is denoted as $\vec{D}_{\alpha,\beta,\delta} = |\vec{C} \cdot \vec{X}_{\alpha,\beta,\delta} - \vec{X}|$. The random vector $\vec{C}$is taken from random numbers within [0,2]; the larger the magnitude, the more the wolves move towards the leader or prey, reflecting a higher level of trust, which promotes exploration in the algorithm. The smaller the magnitude, the more cautious and slight the wolves' movement will be, indicating that the algorithm will search for the optimal solution within the current region. Under the guidance of higher-ranked wolves, the ω wolves advance towards α, β, δ with steps and directions represented as $\vec{X}_{1,2,3} = \vec{X}_{\alpha,\beta,\delta} - \vec{A}_{1,2,3} \cdot \vec{D}_{\alpha,\beta,\delta}$, with the final position being $\vec{X}_{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$.

The grey wolves complete the hunting process through an attack. The process of decreasing the value of a corresponds to the process of approaching the prey, with the fluctuation range of A decreasing as a decreases. When $|\vec{A}| < 1$, the wolves attack the prey (falling into local optima). When $|\vec{A}| > 1$, the grey wolves separate from the prey and seek a more suitable target (global optimum). In the algorithm, this can be reflected as follows: after several iterations, the selected α, β, δ wolves (the best three solutions) remain almost unchanged, or their fitness values change very little, indicating that the algorithm has fallen into a local optimum. Additionally, if the algorithm reaches the preset maximum number of iterations or the improvement in the solution (such as an increase in fitness value) is below a threshold, the prey is considered captured, meaning the algorithm has found a satisfactory solution.

3.3. Fusion model construction

3.3.1. PSO-GWO algorithm

The PSO algorithm tends to converge quickly and can effectively search for the global optimal solution, but in certain cases, it may become trapped in local optima. In contrast, the GWO algorithm demonstrates better exploration capabilities, enabling it to explore different regions of the search space. Combining these two algorithms can increase population diversity, improve the comprehensiveness of the search, reduce the risk of getting trapped in local optima [17], and enhance the algorithm's search efficiency and convergence speed. Furthermore, the social ranking mechanism and hunting strategy of the GWO algorithm provide strong adaptability, while the speed and position update rules of PSO can enhance the stability and robustness of the algorithm. This combination is capable of addressing a broader range of optimization problems, including high-dimensional, multi-modal, and dynamic optimization issues.

Therefore, it is worth considering using the GWO algorithm's exploration capability to quickly locate potential optimal regions in the early stages and then utilizing PSO's fast convergence feature to refine the search in the later stages, rapidly approaching the global optimum. The process for constructing the PSO-GWO fusion algorithm is as follows:

Initialization of α, β, δ positions and scores. The social ranking setup of GWO is retained to avoid premature convergence of the algorithm, promote effective information sharing, and enable other search agents to quickly learn and approach better solutions. Let the number of search agents be N, the search space dimension be dim, and the initial positions and velocities of the search agents be $\vec{X}_i^0$ and $\vec{V}_i^0$, respectively. $i = 1,2,\ldots,N$. Initialize three optimal solutions: $\vec{X}_{\alpha,\beta,\delta} = \vec{0}_{1\times dim}$, with corresponding fitness values $\alpha\_score, \beta\_score, \delta\_score$. For minimization problems, set the initial scores to infinity, and for maximization problems, set the initial scores to negative infinity.

Introducing PSO's velocity and position update mechanism. The position update of each search agent $\vec{X}_i$ is influenced not only by the guidance from $\vec{X}_{\alpha,\beta,\delta}$ (GWO) but also by its own velocity (PSO), providing the search agent with better dynamic adaptability, allowing it to adjust its search behavior based on the current search environment. The velocity update formula for each search agent i in the j -th dimension is given by:

$$\vec{V}^{t+1}{}_{i,j} = w \cdot \left[\vec{V}^{t}{}_{i,j} + \vec{C}_1\vec{r}_1\left(\vec{X}_\alpha - \vec{X}_{i,j}^t\right) + \vec{C}_2\vec{r}_2\left(\vec{X}_\beta - \vec{X}_{i,j}^t\right) + \vec{C}_3\vec{r}_3\left(\vec{X}_\delta - \vec{X}_{i,j}^t\right)\right] \tag{6}$$

where $w = \frac{1+\zeta}{2}$,$\zeta \in [0,1]$ and $w \in [0.5,1]$. The inertia weight w is used to adjust the velocity update, helping to balance the global and local search capabilities. Additionally, the individual best position and the global best position are replaced by $\vec{X}_{\alpha,\beta,\delta}$, which helps to avoid the search process from being trapped in local optima, as it encourages exploration of different regions indicated by the current best three solutions. The position update formula is $\vec{X}_{i,j}^{t+1} = \vec{X}_{i,j}^t + \vec{V}_{i,j}^{t+1}$, which incorporates the influence of velocity while converging toward $\vec{X}_{\alpha,\beta,\delta}$. The distance formulas between the other wolves and the α, β, δ wolves are denoted as $\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_{\alpha_j} - w \cdot \vec{X}_{i,j}|$, $\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_{\beta_j} - w \cdot \vec{X}_{i,j}|$, and $\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_{\delta_j} - w \cdot \vec{X}_{i,j}|$.

According to the above iteration mechanism, in each iteration t, the optimal three solutions α, β, δ are selected from the population based on the current positions $\vec{X}_i^t$ and fitness functions $f(\vec{X}_i^t)$ of all search agents. The position corresponding to the best fitness is $\vec{X}_\alpha$, $\alpha\_score = min\{f(\vec{X}_i^t)\}$. This process is repeated until the maximum number of iterations is reached or the fitness value converges to a certain threshold.

The fused PSO-GWO algorithm is then compared with the original GWO algorithm in the Matlab environment. Both algorithms are tested on various fitness functions, including convex functions, non-convex functions, non-differentiable functions, non-separable functions, and multi-modal functions. After testing, it was found that the fused algorithm converges more quickly, and while enhancing the global search ability of the algorithm, it also reduces the risk of being trapped in local optima.

3.3.2. APSO-KMV model

First, consider attempting to use the APSO algorithm to update the coefficients $\alpha$and βbefore LTD and STD, i.e., $DP = \alpha STD + \beta LTD$. The default probability output by the KMV model is unified as the Equation (7):

$$EDF = \Phi(-DD) = \Phi\left(-\frac{V - DP}{V\sigma_V}\right) = \Phi\left\{-\left[\frac{E + De^{-rT}\Phi(d_2) - \alpha STD\Phi(d_1) - \beta LTD\Phi(d_1)}{E\sigma_E}\right]\right\}, d_1 = \frac{ln\left(\frac{V}{D}\right) + \left(r + \frac{\sigma^2{}_V}{2}\right)T}{\sigma_V\sqrt{T}}, d_2 = d_1 - \sigma_V\sqrt{T} \tag{7}$$

Next, use the APSO algorithm to adjust this. In many financial risk management studies, the values of $\alpha$and $\beta$are typically neither too large nor too small, generally ranging between 0.01 and 0.5, which can encompass the financial characteristics of most companies. Therefore, based on past experience, the initial range of the parameters is set as $\alpha, \beta \in [0.01,0.5]$, a range that provides sufficient flexibility while avoiding extreme parameter values, ensuring the model’s interpretability and stability. To facilitate the training of the classification model, the third quartile is introduced as the threshold. The reason for this is that in default risk prediction, default probabilities often have a right-skewed distribution, with a few companies having a high default risk. Using the third quartile as a threshold can effectively distinguish between the majority of low-risk companies and the few high-risk companies, thereby preventing the model from being overly biased toward the few extremely high-risk companies. The prediction of default categories is determined by comparing the EDF and the threshold (see Equation 8):

$$predicted_{classes} = \begin{cases} 1, EDF \geq Threshold \\ 0, otherwise \end{cases} \tag{8}$$

The goal of the algorithm is to maximize the AUC value, so the objective function is in the form of maximize maximizeAUC(α, β). Since the PSO algorithm is usually designed to minimize a function value, the negative value of the AUC is returned as the objective in the function to achieve maximization. The characteristic of the algorithm is that the larger the particle swarm, the broader the search space, but the computational cost will also increase. To achieve a good balance between accuracy and computational overhead, the number of particles is set to 200, while the maximum number of iterations is set to 200, ensuring that the PSO can fully explore the parameter space while avoiding excessive iterations that may lead to overfitting or excessive time costs. Finally, to prevent the algorithm from converging too quickly or getting trapped in a local optimum in the early stages, an improved adaptive inertia weight algorithm is introduced to dynamically adjust the weight value to enhance the efficiency and stability of the optimization process. After the model outputs the default probability, accuracy and cross-entropy are selected as evaluation metrics (see Equation 9 and 10):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$cross_{entropy} = -\frac{1}{N}\sum_{i=1}^{N}[y_i\, log(EDF_i) + (1 - y_i)\, log(1 - EDF_i)] \tag{10}$$

Where y_i (0,1) is the actual default status, EDF_i is the predicted default probability, and N is the total number of samples.

3.3.3. PSO-GWO-KMV model

Based on the PSO-KMV model, the GWO algorithm is added for optimization. The first step of the model is to use the PSO algorithm to output the global optimal parameters α_{PSO} and β_{PSO}. The second step is to use α_{PSO} and β_{PSO} as the initial points and apply GWO for local optimization to obtain the final parameters $\alpha *$ and $\beta *$.

Initialization settings: The number of search agents is $n_{agents} = 20$, the dimensionality is $dim = 2$, the initial positions are $X_i^0 \sim U([0.01,0.01],[0.5,0.5]), i = 1,2,\dots,n_{agents}$, the optimal solution is $\vec{X}_{\alpha,\beta,\delta}$, and the initial score is infinity.

Iterative update: In each iteration $t = 1,2,\dots,max\_iter$, the following steps are performed: Evaluate the fitness, and for each search agent i, calculate the objective function $f(x_i^t) = -AUC(x_i^t)$, where AUC is based on EDF and the actual default labels; Update α, β, δ. The mechanism is (see Equation 11):

$$\begin{aligned} &if\ f(x_i^t) < \alpha_{score}, \text{ then } \alpha_{score} = f(x_i^t), \vec{X}_\alpha = x_i^t \\ &\text{else if } f(x_i^t) < \beta_{score}, \text{ then } \beta_{score} = f(x_i^t), \vec{X}_\beta = x_i^t \\ &\text{else if } f(x_i^t) < \delta\_score, \text{ then } \delta\_score = f(x_i^t), \vec{X}_\delta = x_i^t \end{aligned} \tag{11}$$

Update the search agent position: For each agent i and dimension $j (j = 1,2)$, $\vec{X}_1 = \vec{X}_{\alpha_j} - \vec{A} \cdot \vec{D}_\alpha$, $\vec{X}_2 = \vec{X}_{\beta_j} - \vec{A} \cdot \vec{D}_\beta$, $\vec{X}_3 = \vec{X}_{\delta_j} - \vec{A} \cdot \vec{D}_\delta$, where $\vec{D}_{\alpha,\beta,\delta} = |\vec{C} \cdot \vec{X}_{\alpha,\beta,\delta} - \vec{X}_{i,j}^t|$, $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \vec{C} = 2\vec{r}_2$, $\vec{a} = 2 - \frac{2t}{max\_iter}$, $r_1, r_2 \sim U(0,1)$. Then, $\vec{X}_{i,j}^{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$, the position is clipped as $\vec{X}_{i,j}^{t+1} = max(\, min(\vec{X}_{i,j}^{t+1}, 0.5), 0.01)$ to ensure the values are within the allowed bounds. The algorithm stops after reaching the maximum number of iterations.

4. Empirical analysis

4.1. Data sources and description

The data used in this experiment comes from the CSMAR database. The dataset includes financial indicators such as the market value of the company, debt market value, equity market value, equity volatility, and the risk-free interest rate. The assumption is made that the debt maturity time T=1 year, and the risk-free interest rate is selected as the 10-year government bond yield of 1.81%.

4.2. Data preprocessing

The following steps were taken to preprocess the raw data:

Missing Value Handling: Rows with missing values were removed, and data from 5,234 companies were ultimately used.

Normalization: To make different variables comparable and suitable for model training and parameter optimization, normalization was applied. Given that the absolute values of variables such as market value and debt market value play a significant role in calculating the default probability in the KMV model, normalization is more reasonable than standardization. The formula for normalization is: $X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$. Standardization converts data to a distribution with a mean of 0 and a standard deviation of 1, while normalization preserves the relative magnitude relationships between variables, which is more helpful for the calculation of default distance in the KMV model. Additionally, normalization ensures that companies of different sizes are comparable in terms of risk evaluation within the KMV model. For default judgment, after calculating the default probability using the KMV model, the third quartile is used as the threshold, i.e., $\text{Threshold} = 2.23\text{E} - 11$. This approach helps the model better capture the distribution characteristics of default risk and provides more reasonable label division standards for the classification model. Specifically, companies with a default probability less than this threshold are considered non-default, labeled as 0; those with a default probability greater than or equal to this value are considered default, labeled as 1.

4.3. Calculation results and comparison

4.3.1. Comparison between the APSO-KMV model and the original KMV model

As the number of iterations increases, the outputs of the APSO-KMV model converge to α =0.0496 and β =0.2508. After optimization, the values of α and β were tested on the test set, and the AUC value improved from 0.7362 to 0.9994. By plotting the ROC curve, it was found that the optimized model significantly improved the true positive rate while maintaining a low false positive rate. This indicates that the PSO-optimized model has stronger discriminatory power in classifying whether a company defaults. Especially in the region with low false positive rates, the optimized model can more accurately distinguish between defaulting and non-defaulting companies, thereby reducing the false alarm rate in practical applications. The loss function $\lambda(\alpha_2|w_1)$ is defined as the cost of Type I error (predicting a defaulting borrower as normal), and $\lambda(\alpha_1|w_2)$ is defined as the cost of Type II error (predicting a normal borrower as default). Research shows that the cost of Type I error is much higher than that of Type II error, so it is stipulated that $-1 < \lambda(\alpha_1|w_2) < \lambda(\alpha_2|w_1) < 0$. Under this premise, the optimized model effectively reduces the Type I error rate, which is significant for prediction models in this context.

After comparing the accuracy and binary cross-entropy loss, it was found that the accuracy of the APSO-KMV model improved from 0.2610 before optimization to 0.9996, while the binary cross-entropy loss increased from 0.7006 to 4.1990. From the results, it can be seen that the accuracy of the PSO-optimized model is close to 100%, indicating excellent data fitting. In contrast, the accuracy of the original parameter model is relatively low, suggesting that the original parameters are not effective for predicting corporate defaults in Chinese data. Moreover, the cross-entropy loss of the APSO-optimized model is much higher than that of the original parameter model, which may be due to the model overfitting the data. Therefore, in practical applications, regularization techniques may need to be introduced to balance the model’s generalization performance.

4.3.2. Comparison of PSO-GWO-KMV model with APSO-KMV and KMV models

To ensure accuracy while reducing binary cross-entropy loss, a hybrid strategy combining PSO and GWO was introduced to optimize key parameters in the KMV model. Additionally, the decision threshold was further improved through automatic search to enhance the model’s accuracy and reliability. Furthermore, to address the overfitting issue in the PSO-KMV model, a regularization method was incorporated. The threshold optimization method involves setting the search range as Threshold ∈ $[0.1,0.9]$, with a step size of 0.01, while maintaining the classification rule. The objective function is $Threshold *= arg \max_{Threshold} AUC, AUC = roc\_auc\_score(y, EDF)$, and the optimal threshold and corresponding $Threshold *$ are output. The objective function with regularization added is $f(x_i^t) = -(AUC(x_i^t) - \lambda(\alpha^2 + \beta^2))$, where λ is the regularization coefficient, defaulted to 0.01. By running the PSO-GWO algorithm, the parameters of the PSO-GWO-KMV model were compared with those of other models (see Table 1):

Table 1. Comparison of three major model indicators

Model Comparison		PSO-GWO-KMV	PSO-KMV	KMV
Model Parameters	α	0.0496	0.0496	1
	β	0.2690	0.2508	0.5
AUC		0.9987	0.9994	0.7362
Accuracy		0.7603	0.9996	0.2610
Binary Cross-Entropy Loss		4.0717	4.1990	0.7006

The optimized model output values for α were essentially consistent, with only slight adjustments made to β. From the perspective of key indicators in the PSO-GWO-KMV model, the AUC (Area Under the Curve) remains close to 1, indicating that the model has a strong ability to distinguish default risks. In terms of accuracy and binary cross-entropy loss, the PSO-GWO-KMV model has not fully resolved the overfitting problem observed in the PSO-KMV model. The model’s accuracy suggests that approximately 76% of default predictions are correct on the current dataset, which is still much higher than the original KMV model but lower than the PSO-KMV model. The binary cross-entropy value remains relatively high, indicating that the model’s performance in handling extreme cases is still not ideal, though extreme situations are not common in practical applications. The model still has room for further optimization when processing complex financial data.

5. Conclusion and suggestions

Banks play a dominant role in China’s financial system, and credit risk management is the core business of commercial bank operations. This research focuses on the narrow sense of credit risk in commercial banks, specifically assessing the credit risk of enterprises that have business dealings with commercial banks. The KMV model is a mature model in the industry for evaluating the credit risk of listed companies, but its default point parameters are based on calculations from foreign databases, which may lack external validity. To address this, this paper uses Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) algorithms to correct the KMV model and output optimized parameters. The study found that the combined APSO-KMV model and PSO-GWO-KMV model provide a more accurate measurement of default risk in listed companies compared to the original KMV model.

Based on the empirical research findings, the adjusted KMV model’s calculation of the default distance and default probability can serve as reference indicators for commercial banks when assessing the credit risk of listed companies, assisting managers in making loan decisions. Of course, aside from listed companies, there are also many non-listed companies that apply for loans from banks. However, since the KMV model uses stock prices as a substitute for company asset values, it is not suitable for evaluating non-listed companies. To address this, various intelligent optimization algorithms can be utilized to analyze the implicit relationships between the asset values of non-listed companies and their publicly disclosed financial data. Additionally, the growth rate of enterprise asset value has been neglected. Therefore, it is worth considering the integration of various dividend growth models to improve the evaluation model and more accurately measure risk.

References

[1] Xia, H. F. (2009). *Research on credit risk measurement and management of commercial banks* (pp. 1-130). Zhejiang University Press.
[2] Caouette, J. B., Altman, E. I., & Narayanan, P. (1998). *Managing credit risk: The next great financial challenge*. John Wiley & Sons.
[3] Hale, R. (1983). *Credit analysis: A complete guide*. New York: McGraw-Hill.
[4] Cao, S., & Liu, T. (2003). Research on enterprise credit evaluation based on neural networks. *Control Engineering, 5,* 404-406.
[5] Beaver, W. H. (1966). Financial ratios as predictors of failures. *Journal of Accounting Research, 6,* 71-102.
[6] Ohlson, J. S. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research, 18,* 109-131.
[7] Chen, J. (1999). Empirical analysis of the prediction of financial deterioration of listed companies. *Accounting Research, 4,* 32-39.
[8] Derbali, A. H., & Hallara, S. (2012). The current models of credit portfolio management: A comparative theoretical analysis. *International Journal of Management and Business Research, 2*(4), 271-292.
[9] Xiao, B. Q., Bai, W., & Yao, Y. (2016). Research on credit evaluation of small and micro enterprises based on LS-SVM. *Auditing and Economic Research, 31*(6), 102-111.
[10] Liu, M., & Lin, C. D. (2005). Credit risk assessment model of commercial banks based on support vector machine. *Journal of Xiamen University (Natural Science Edition), 1*, 29-32.
[11] Li, M., & Chen, L. Q. (2007). Empirical analysis of credit risk identification in commercial banks based on BP neural network. *Nanjing Social Sciences, 1*, 18-29.
[12] Breiman, L. (1986). *Classification and regression trees*. Wadsworth Statistics/Probability.
[13] Wang, Q. (2010). Credit risk assessment of supply chain financial models based on decision trees. *New Finance, 4*, 38-41.
[14] Xiao, B. Q., Bai, W., & Yao, Y. (2016). Research on credit evaluation of small and micro enterprises based on LS-SVM. *Auditing and Economic Research, 31*(6), 102-111.
[15] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). IEEE. https://doi.org/10.1109/ICNN.1995.488968
[16] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61.
[17] Qiu, S. M., Zhang, B., Du, X. L. (2024). A particle swarm position update-based grey wolf optimization algorithm for K-Means text classification method. CN202010129423.9 [Patent No. CN111368077A].