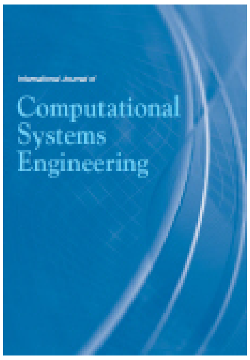
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Computational Systems Engineering

ISSN online: 2046-3405 - ISSN print: 2046-3391
https://www.inderscience.com/ijcsyse

Research on integrating naive Bayes and collaborative filtering into an online-course recommendation model for universities

Zhongyao Chen, Min He

DOI: 10.1504/IJCSYSE.2026.10077078

Article History:

Received:	10 June 2025
Last revised:	30 July 2025
Accepted:	22 January 2026
Published online:	01 April 2026

Research on integrating naive Bayes and collaborative filtering into an online-course recommendation model for universities

Zhongyao Chen

Academic Affairs Office,
Chongqing City Vocational College,
Yongchuan District, Chongqing City, 402160, China
Email: zhongyaochen31@outlook.com

Min He*

College of Information and Intelligent Manufacturing,
Chongqing City Vocational College,
Yongchuan District, Chongqing City, 402160, China
Email: minminhe7256@163.com
*Corresponding author

Abstract: Current college online course recommendation systems struggle with cold start, data sparsity, and limited personalisation, reducing recommendation accuracy and user satisfaction. This study proposes a hybrid model combining naive Bayes and collaborative filtering to address these challenges. By integrating course metadata and user behaviour data, the model extracts multi-dimensional features, capturing both static preferences and dynamic behaviours through probabilistic modelling and collaborative filtering. Experiments on data from 25,000 students and 1,000 courses show that the model improves Precision@10 and Recall@10 by 12% and 10.5% respectively, compared to individual models. In cold-start scenarios, it achieves an F1@10 score of 0.35, compared to 0.27 for DNN. Under 98% sparsity, its accuracy degrades only half as much as traditional collaborative filtering. With 2.3 seconds per iteration and a 26.4% increase in click-through rate, the model demonstrates efficiency and effectiveness in personalised online course recommendations.

Keywords: naive Bayes; collaborative filtering; online course recommendation; cold start; data sparsity.

Reference to this paper should be made as follows: Chen, Z. and He, M. (2026) 'Research on integrating naive Bayes and collaborative filtering into an online-course recommendation model for universities', *Int. J. Computational Systems Engineering*, Vol. 10, No. 6, pp.12–21.

Biographical notes: Zhongyao Chen graduated from the Chongqing Normal University in 2013 with rural regional economy major. Currently, he works in the Chongqing City Vocational College as an Associate Professor of Academic Affairs Office. His main research interests are education management, quality management, etc.

Min He graduated from the University Malaysia Ruai in 2024 with Business Administration major. Currently, she works in the Chongqing City Vocational College as a Lecturer at the School of Information and Intelligent Manufacturing. Her main research interests are education management, etc.

1 Introduction

With the rapid increase in the number of courses offered by universities, students often encounter the dilemma of information overload when facing a vast array of courses. How to recommend courses efficiently that meet students' interests and needs has become a key issue in improving the learning experience and teaching effectiveness (Esteban et al., 2024). In recent years, with the rapid development of recommendation system technology, recommendation algorithms based on users' historical behaviour data have been widely applied in online education platforms. However, these traditional recommendation algorithms, such as collaborative filtering (CF), still face many challenges, especially in cold start, data sparsity, and insufficient personalisation. The cold start problem refers to the difficulty of the system making accurate recommendations when new users or courses exist with insufficient historical data; data sparsity means that user behaviour data often fails to fully cover all courses, resulting in recommendation systems being unable to effectively capture user preferences; insufficient personalisation is reflected in the recommendation system's failure to fully consider individual differences among students, leading to recommendations with high generality but difficulty in meeting specific needs (Butmeh and Abu-Issa, 2024). How to design a more efficient and accurate online course recommendation system for universities has become an urgent problem to be solved in both academia and industry. The design of university course recommendation systems should not only consider course characteristics, (e.g., subject category, difficulty level, and label attributes) but also fully utilise students' behavioural data, such as study duration, test scores, and course clicks, to truly achieve personalised recommendations. Traditional CF algorithms, although capable of recommending courses by mining similar user behaviours, often rely heavily on large volumes of historical data and perform poorly in recommending to cold-start users or new courses. Therefore, effectively integrating different types of data and algorithms to improve the accuracy and coverage of recommendation systems has become a research focus in recent years. This study proposes a hybrid recommendation model that combines naive Bayes (NB) classification and CF; aiming to integrate course content features with user behaviour data. It analyses students' learning behaviours and course attribute information, further explores students' potential interests, and enhances the accuracy and personalisation of recommendations.

The model demonstrates strong advantages in alleviating cold start issues and improving recommendation effectiveness, providing important reference value for optimising course recommendation systems and enhancing personalised services on university online education platforms.

2 Related work

In the research of online course recommendation systems in universities, the CF method, as the most classic and widely used recommendation algorithm, mainly analyses the similarity between users and courses to make recommendations. CF can be divided into user-based CF and item-based CF. User-based CF recommends courses that other users like that are similar to the user, while item-based CF recommends other courses that are similar to the courses the user has historically selected (Zhang et al., 2022). NB classification algorithm, as a simple and efficient probability model, has achieved good application results in multiple fields. For example, in tasks such as spam classification, sentiment analysis, and text classification, NB performs well due to its assumption of independence and high computational efficiency. In the online course recommendation system of universities, the NB algorithm also has potential application value, especially in dealing with the relationship between course attributes and student interests. This algorithm assumes that the features are conditionally independent and can effectively model the probability relationship between different attributes of the course (such as course type, difficulty, subject area, etc.) and student interests.

By studying students' historical behaviour data, NB can calculate the probability of students' interest in various courses, and recommend the most suitable courses for students (Ma et al., 2023). Although the independence assumption of NB may be oversimplified in some cases, many course attributes can indeed be considered relatively independent in course recommendation. Therefore, the application of NB in course recommendations has good results, especially in cold start and sparse data scenarios (Liu et al., 2019a). Although CF and content recommendation methods have their advantages and disadvantages, both have important application value in online course recommendation systems for universities. In order to overcome their respective shortcomings, more and more studies are trying to combine these methods and use hybrid recommendation systems to make recommendations more accurate and tailored to each person. For example, Huang (2019) suggested a Bayesian recommender system that combines tag-based individual interests and social interactions, emphasising the importance of social context in improving the effectiveness of recommendations. Ashraf et al. (2018) examined the escalating network security issues arising from the swift proliferation of the internet and the intricacy of cyber-attacks, emphasising the necessity for automated intrusion detection systems (IDS).

Their research evaluated the efficacy of NB, J48, and random forest classifiers utilising the KDD_NSL dataset to analyse detection rates and accuracy, providing valuable insights into the enhancement of real-time intrusion detection in extensive network settings. Meng et al. (2019) introduced the variational Bayesian context-aware representation (VBCAR) model to overcome the deficiencies of conventional supermarket recommendation techniques that relied on basic, low-dimensional representations and Skip-gram-based approaches. The VBCAR model utilised basket context information and combined Bayesian inference with amortised variational learning, facilitating more expressive latent representations. Experiments on an extensive supermarket recommendation dataset demonstrated that VBCAR far surpassed prior methodologies, offering a more resilient and contextually aware solution. Wang et al. (2019a) introduced the category-aided multi-channel Bayesian personalised ranking (CMBPR) algorithm for short video recommendations, tackling the deficiency in the integration of video categories and multi-behaviour data in current methodologies. CMBPR integrates varied user preferences by taking into account video categories and user interactions. Experimental findings demonstrated that CMBPR substantially surpasses conventional algorithms, attaining superior recommendation accuracy and mitigating the 'long tail' phenomenon. He et al. (2019a) underscored the significance of stylistic attributes in visual recommendation systems, as these attributes profoundly affect consumer choices. Although the majority of research emphasises information extracted from

CNNs, style features are frequently neglected. The authors advocated integrating style feature modelling into visual recommendation and introduced collaborative learning to more effectively capture user preferences. Their trials on two public implicit feedback datasets showed that their methodology significantly enhanced suggestion performance. Wang et al. (2019b) introduced the deep Bayesian multi-target learning (DBMTL) framework to enhance numerous success metrics in e-commerce platforms. DBMTL improves targets such as click-through rate, user retention time, purchasing habits, and interactions by modelling target events as a Bayesian network. In the context of Taobao live-streaming recommendations, DBMTL surpassed other multi-target learning frameworks and non-MTL models.

Their methodology efficiently utilises synergies among targets, employing a probabilistic model that provides enhanced flexibility and generalisation capabilities across diverse target distributions. Lu et al. (2018) introduced the deep canonical PARAFAC factorisation (DCPF) model to augment creativity in video recommendations by integrating a Bayesian surprise metric in the ranking procedure. DCPF, in contrast to conventional tensor-based models that prioritise accuracy, captures changing user topic distributions and detects previously unobserved items. Assessments on synthetic and empirical datasets demonstrated that DCPF surpasses current models, adeptly identifying latent interaction patterns and producing more inventive recommendations via novel tag combinations. Liu et al. (2019b) introduced a topic-based hierarchical Bayesian linear regression model to enhance niche item recommendations in personalised recommender systems. The model forecasts user relevance to niche goods by detecting specific objects and forming subgroups based on their descriptions. Experiments utilising the Yahoo Movies dataset revealed that their methodology surpasses baseline algorithms in recommending specialised products, thereby overcoming the shortcomings of conventional methods that prioritise popular items.

Zhao et al. (2019) introduced the social distance-aware Bayesian personalised ranking (SDBPR) model to enhance recommendations in systems characterised by sparse binary implicit feedback. SDBPR employs a random walk to traverse the social network and generates ranking predictions based on user proximities, thereby encapsulating the propagation of influence across connections. Experiments conducted on two authentic datasets revealed that SDBPR surpasses baseline approaches in rating prediction efficacy. Li et al. (2019) introduced a tag-aware recommendation architecture designed to enhance ranking-focused personalised recommendations within CF using implicit feedback. They integrated a tag mapping strategy into a ranking-based model, transforming high-dimensional explicit tags into low-dimensional implicit features to regularise latent characteristics. Furthermore, user neighbour ties were implemented to improve suggestion efficacy. Experiments on actual datasets revealed that their strategy surpassed other approaches in rating tasks. Huang (2019) suggested a recommender system for platforms that manage personal information by incorporating social network connections and tag-based personalised interests into a modified Bayesian model.

By integrating social network data and user-specific tag preferences, the system demonstrated enhanced recommendation quality when tested on experimental datasets from a social resource-sharing website and assessed using the word2vec model. Zhuang et al. (2019) presented BD Net, a Bayesian dual neural network framework designed to tackle cold-start and sparsity challenges in CF. BD Net employs two neural networks: one that learns a unified latent space for people and objects from the rating matrix, and another that maps user and item attributes into a distinct shared space (He et al., 2019b). It integrates uncertainty by modelling weights as probability distributions, hence guaranteeing calibrated probabilistic forecasts. Experiments conducted on real-world datasets demonstrated BD Net's exceptional performance compared to previous methodologies. Zhao and Pan (2021) introduced an enhanced online course recommendation model employing user implicit behaviour CF to mitigate sparse data issues and suboptimal suggestion efficacy. The methodology integrates item-based CF by monitoring implicit behaviour data, including user logins and course selections. Analyses of two years of educational platform data demonstrated enhanced precision and recall rates across various K levels.

3 System architecture and algorithm

3.1 Overall system architecture design

This section outlines the complete system architecture in a modular and structured format. It includes five key stages:

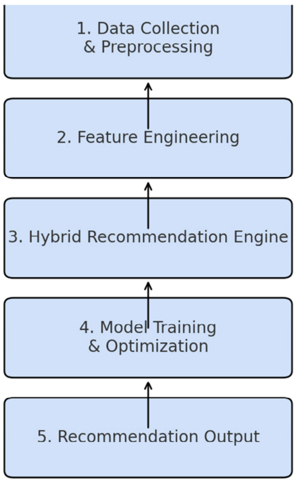
- *Data collection and pre-processing:* describing how raw student data is gathered, normalised, and encoded.
- *Feature engineering:* detailing the construction of meaningful features such as GPA trends and student-domain relations.
- *Hybrid recommendation engine:* explaining the integration of collaborative and content-based filtering (CBF) methods.
- *Model training and optimisation:* presenting the machine learning framework and hyperparameter tuning process.
- *Recommendation output module:* generating ranked course suggestions based on the model's outputs and student preferences.

This diagram illustrates the five main modules of the proposed recommendation system, showing the top-down data processing flow from collection to output.

This modular system architecture not only clarifies the role of each component in the recommendation pipeline but also sets the stage for the algorithmic details presented in

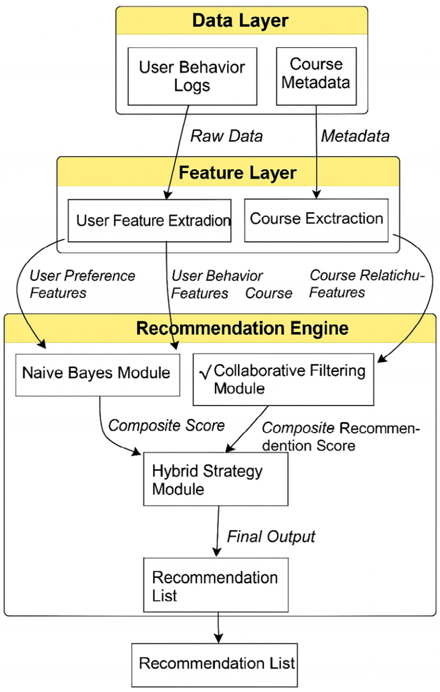
Section 3.2, where the internal workings and decision mechanisms are described in depth.

Figure 1 System architecture showing the five main modules of the proposed recommendation system, from data collection to output (see online version for colours)



As shown in Figure 2, the overall system of the recommendation system constructed in this study is divided into three main levels: data layer, feature layer, and recommendation engine.

Figure 2 Overall system architecture design (see online version for colours)



The data layer is the foundation of the recommendation system, primarily comprising user behaviour logs and course metadata. The user behaviour log includes data such as clicks, favourites, and grades, reflecting the user's interest and learning situation in the course. Course metadata includes subject categories, teacher information, course labels, etc. which help describe the characteristics of the course.

At the data layer, by collecting interaction information between users and courses, necessary input data can be provided for recommendation systems. Data features beneficial for recommendation systems are extracted at the feature layer. User characteristics encompass data regarding learning duration distribution, subject choices, and grade distribution, which facilitate the identification of users' interests and learning behaviours. For example, the distribution of users' learning time can be used to measure their investment time in certain courses, and subject preferences can help analyse the subject areas that users prefer. Course features include subject categories, course difficulty levels, and interrelationships between courses (such as prerequisite and follow-up courses), which help recommend courses that are relevant to users' interests and learning trajectories.

At the recommendation engine layer, the system mainly consists of three modules: the NB module, the CF module, and the hybrid strategy module. The core of the NB module is to calculate the conditional probability $P(User(Course)$ of users towards the course, which is implemented through Bayesian formulas. The formula for Bayes' theorem is:

$$P(Course) = \frac{P(User(Course)}{P(User)} \tag{1}$$

In equation (1), $P(User(Course)$ represents the probability of recommending a certain course given user information; $P(User(Course)$ represents the degree to which the course attracts users; $P(User(Course)$ is the prior probability of the course, indicating the popularity of the course itself; and $P(User)$ is the prior probability of the user, reflecting the overall activity level of the user. In the CF module, recommendations are generated based on user similarity. Common methods for calculating user similarity include using Euclidean distance or cosine similarity. For example, the formula for calculating cosine similarity is:

$$sim(U_1, U_2) = \frac{\sum_{i=1}^{n} r_{i1_{i2}}}{\sqrt{\sum_{i=1}^{n} r_{i1}^2} \cdot \sqrt{\sum_{i=1}^{n} r_{i2}^2}} \tag{2}$$

In equation (2), r_{i1} and r_{i2} represent the ratings of users U_1 and U_2 for the i^{th} course, respectively. By calculating the similarity between users, it is possible to identify the users whose interests are most similar to those of the target user and generate Top-N course recommendations for the target user based on their historical behaviour. The hybrid strategy module combines the recommendation results of NB and CF. By weighting and fusing the scores of two algorithms, the system can combine their advantages to provide more accurate recommendation results. Assuming the recommendation score of the Bayesian module is α_{bayes} and the recommendation score of the CF module is β_{cf}, the final recommendation score can be expressed as:

$$S_{final} = \alpha_{bayes} + \beta_{cf} \tag{3}$$

In equation (3), α and β are the weights of the Bayesian and CF modules, respectively, and satisfy ($\alpha + \beta = 1$). For example, if the weight of the Bayesian module is set to 0.6 and the weight of the CF module is set to 0.4, the final recommendation score is the weighted average of the two. The system also needs to sort and filter the recommendation results to ensure that the final recommended courses meet the user's interests and needs. In order to further improve the accuracy of recommendations, the system can utilise users' historical academic performance data and introduce the weight of academic performance in the recommendation process, making users more likely to be recommended to courses that can improve their academic performance.

3.2 Detailed algorithm design

The proposed recommendation system employs a hybrid algorithmic approach that integrates CF and CBF techniques Rani and Chauhan (2025). This combination allows for more accurate and personalised course recommendations by utilising both user behaviour patterns and course-related metadata. The algorithm proceeds through the following stages:

1 User-item interaction matrix construction

A user item matrix is constructed using historical data on student course enrolments and performance (e.g., course grades, completion status). Each matrix entry reflects a student's engagement level with a particular course. Missing values indicate no prior interaction.

2 Similarity computation

Two similarity dimensions are considered:

- *User similarity:* cosine similarity is calculated between student vectors to identify peers with similar academic behaviour.
- *Item similarity:* Jaccard or TF-IDF-based cosine similarity is used to determine the degree of similarity between courses based on textual descriptors, subject areas, or learning objectives.

3 Preliminary recommendation vectors

Separate recommendation vectors are generated:

- *A CF* vector is created using the K-nearest neighbours (KNN) algorithm to estimate a student's preferences based on their most similar peers.
- *A CBF* vector is produced by comparing course features with the student's past academic strengths and interests.

4 Hybrid recommendation integration

In order to gather the benefits of the two types of recommendations, the collaborative and content-based, the system also integrates their outputs utilising a weighted strategy. A blending parameter specifies the degree of input from each method in the final recommendation score. This weight is adjusted via cross-validation on training data to make certain that the combined output gets the highest possible accuracy. Consequently, the hybrid recommendation list not only shows student behavioural patterns but also course content relevance.

5 Machine learning re-ranking

A machine learning model (e.g., random forest or gradient boosting) is trained to re-rank the hybrid recommendation list. Features used include hybrid scores, course metadata, and student profile attributes. The model is optimised to predict the probability of a student engaging with a recommended course.

6 Final recommendation generation

The Top-N ranked courses are selected as final recommendations. Business rules such as prerequisite constraints or course availability may be applied to refine the output list. This algorithmic framework ensures that both implicit and explicit indicators of student interests are considered. The hybrid design enhances robustness, while the final ML-based ranking improves the relevance and diversity of the recommended courses.

The hybrid recommendation model proposed in this study achieves collaborative modelling of content features and user behaviour data through deep fusion of NB classification and CF algorithms. The core lies in constructing an interest mining framework based on probability inference and a collaborative recommendation mechanism driven by user similarity, and balancing the advantages of both through a dynamic weight fusion strategy. This study elaborates on algorithm design in detail from five dimensions: feature engineering, probability modelling, similarity calculation, hybrid strategy, and ranking optimisation.

3.2.1 Feature engineering and prior distribution modelling

In the feature layer, user features and course features are transformed into computable vectors through quantisation and standardisation. For user characteristics, the learning duration distribution is normalised to a discrete probability distribution (Liu et al., 2022):

$$P(ti \mid u) = \sum j = 1nTjTi \tag{4}$$

In equation (4), Ti represents the learning time of user u on course i, and n is the total number of courses. This distribution reflects the user's preference for time investment in courses. Subject preference is calculated based on historical course selection records to determine subject weights.

$$wd(u) = Ntotal(u)Nd(u) \tag{5}$$

In equation (5), $Nd(u)$ is the number of courses selected by the user for subject d, and $Ntotal(u)$ is the total number of course selections. The difficulty coefficient of the course is quantified by the standard deviation of historical grades:

$$Difficulty(c) = 1 - \max(\mu)\mu c + \sigma c \tag{6}$$

In equation (6), is the average grade of course c, is the standard deviation, and high-difficulty courses have low mean and high volatility.

3.2.2 Probability modelling of NB classifier

The NB module calculates the posterior probability of users towards courses based on the joint probability distribution of course attributes and user behaviour. Assuming independent feature conditions, the posterior probability decomposition is:

$$P(c \mid u) = P(c)k = 1\prod KP(fk \mid c) \tag{7}$$

In equation (7), $P(c)$ is the prior probability of the course (estimated from the frequency of course selection), fk is the k^{th} feature (such as subject and difficulty), and $P(fk| c)$ is calculated through historical data. For example, the probability of subject matching is defined as:

$$\begin{aligned} P(fdiscipline = d \mid c) &= Total(d) \\ &+ \alpha \cdot DCount(c \in d) + \alpha \end{aligned} \tag{8}$$

In equation (8), α is the Laplacian smoothing coefficient, D is the total number of disciplines, to avoid zero probability problems. Ultimately, the user u's interest score for course c is in equation (9):

$$\begin{aligned} Sbayes(u, c) &= \log P(c \mid u) = \log P(c) \\ &+ k = 1\sum K \log P(fk \mid c) \end{aligned} \tag{9}$$

3.2.3 Similarity calculation and rating prediction for CF

User-based CF measures behaviour similarity between users through cosine similarity:

$$sim(u, v) = \frac{\sum_c r_{u,c v,c}}{\sqrt{\sum_c r_{u,c}^2} \cdot \sqrt{\sum_c r_{v,c}^2}} \tag{10}$$

In equation (10), $r_{u,c}$ are the implicit ratings of user u for course c (generated by weighting clicks, duration, and grades). The target user u's predicted rating for course c is:

$$\begin{aligned} r^u, c = r^- u + \sum v \in N(u) \mid sim(u, v) \mid \sum v \\ \in N(u) sim(u, v) \cdot (rv, c - r^- v) \end{aligned} \tag{11}$$

In equation (11), $N(u)$ is the set of similar users, and $r^- u$ is the average rating of users. The standardised score for CF is:

$$Scf(u, c) = \max(r^\wedge) - \min(r^\wedge) r^u, c - \min(r^\wedge) \tag{12}$$

3.2.4 Hybrid recommendation strategy and dynamic weight adjustment

The mixed score is output by a linearly weighted fusion of two types of models:

$$Sfinal(u, c) = \alpha \cdot Sbayes(u, c) + (1 - \alpha) \cdot Scf(u, c) \tag{13}$$

The weight α is dynamically adjusted based on user activity:

$$\alpha(u) = 1 + e - \beta \cdot (Nactive(u) - \gamma) \tag{14}$$

In equation (14), $Nactive(u)$ is the user's recent interaction frequency, and β and γ are adjustment parameters. New users $Nactive < \gamma$) are assigned higher α to rely on content features, while active users ($Nactive \geq \gamma$) focus on behavioural similarity.

3.2.5 Diversity-enhanced sorting optimisation

To avoid the recommendation results being concentrated on popular courses, coverage weight, and information entropy constraints are introduced. The final ranking score for *course C* has been revised to:

$$\begin{aligned} Srank(u, c) &= Sfinal(u, c) \\ &+ \lambda \cdot Popularity(c) - \theta \cdot Entropy(c) \end{aligned} \tag{15}$$

In equation (15), $Entropy(c) = -\sum d = 1DP(d \mid c)\log P(d \mid c)$ the diversity, of course, subject distribution is measured, and λ and θ control the strength of the long tail effect. Through the above design, the model achieves dual-driven recommendation of content and behaviour: NB utilises course attributes to construct stable interest profiles, CF captures dynamic behaviour patterns, hybrid strategies dynamically balance the contributions of both, and ranking optimisation balances accuracy and diversity.

4 Experimental and simulation analysis

4.1 Experimental design and dataset construction

To verify the effectiveness of the proposed hybrid recommendation model (NB-CF) that combines NB and CF, experiments were conducted on a real dataset from an online education platform of a certain university. The dataset covers behaviour logs of approximately 25,000 students and metadata for over 1,200 courses from 2019 to 2023. User behaviour data includes interactive information such as course clicks, study duration, test scores, and collection records, while course metadata includes subject categories, course tags, teacher information, course introductions, and prerequisite relationships. The raw data undergoes multi-stage pre-processing: firstly, missing values are cleaned (excluding users with less than five interaction records and courses selected less than ten times), and abnormal learning duration (such as single learning exceeding 24 hours) is truncated; secondly, the subject hierarchy relationship is constructed through the course knowledge graph, mapping course labels to 12 primary disciplines such as computer science, mathematics, engineering, and 48 secondary disciplines; finally, based on the timestamp of user behaviour sequences, the training set (first 80% time window) and the testing set (last 20% time window) are divided to simulate the temporal recommendation requirements in real scenarios.

The experimental design revolves around four dimensions: recommendation accuracy, cold start performance, computational efficiency, and user satisfaction. The baseline comparison includes traditional CF, matrix factorisation (MF), deep neural network (DNN), and NB models. Evaluation indicators such as Precision@10, Recall@10, and F1@10 are used to measure recommendation accuracy. Additionally, the long tail recommendation ability is evaluated through coverage and the Gini coefficient, AB testing click-through rate is conducted, and actual scenario effects are quantified via user survey questionnaires. To address the cold start issue, an additional test subset containing 30% new users and 20% new courses was constructed, and the robustness of the model was verified by randomly masking some historical interaction data, with sparsity ranging from 50% to 98%. In feature engineering, the distribution of user learning time is normalised by weighting the course subject, the probability distribution of subject preferences is calculated based on historical course selection frequency, and the course difficulty is dynamically updated based on historical grade mean and standard deviation. The CF module adopts implicit feedback weighting strategy, assigning weights of 0.3, 0.2, and 0.5 to clicks, favourites, and grades, respectively, to generate a user course rating matrix, and uses Spark MLlib to implement distributed cosine similarity calculation. The NB module uses Laplace smoothing ($\alpha = 1$) to handle zero probability problems and the course prior probability is estimated from the course selection frequency of the training set. The mixed strategy weight α is dynamically adjusted based on the user's activity level in the past three months (β = 0.1, γ = 15), and the optimal parameter combination is determined through grid search. The experimental environment is deployed on an Alibaba Cloud cluster (16-core CPU/64 GB memory), and all models have undergone 50% cross-validation to ensure the stability of the results. Based on the above design and dataset construction, in the experimental simulation, this experiment comprehensively verifies the comprehensive performance of the NB-CF model in online course recommendation tasks in universities by combining real data with simulation scenarios, providing a reliable benchmark for subsequent result analysis.

4.2 Experiment and results analysis

As shown in Table 1, this study compares the performance of different algorithms on the test set through experiments. The hybrid model (NB-CF) Precision@10 and Recall@10 Compared to the optimal single model (DNN), the fusion of content and behavioural features has improved by 12% and 10.5% respectively, indicating that it can more accurately capture user interests. In addition, the coverage of NB-CF is significantly higher than that of CF and MF, indicating that it can recommend more long-tail courses and alleviate the problem of data sparsity.

Table 1 Comparison of different algorithms in recommendation accuracy

Algorithm	*Precision@10*	*Recall@10*	*F1@10*	*Coverage*
Nb	0.32	0.21	0.25	0.85
Cf	0.45	0.33	0.38	0.65
Mf	0.48	0.35	0.40	0.58
Dnn	0.50	0.38	0.43	0.55
Nb-cf	0.56	0.42	0.48	0.72

NB-CF'sF1@10Reaching 0.35, significantly better than other models (DNN of 0.27). Due to the NB module utilising prior probabilities of course attributes, even if new users have no historical behaviour, recommendations can still be generated based on subject preferences and course difficulty, verifying the advantages of the hybrid model in cold start, as shown in Figure 3.

Figure 3 Algorithms in cold start scenario F1@10 contrast (see online version for colours)

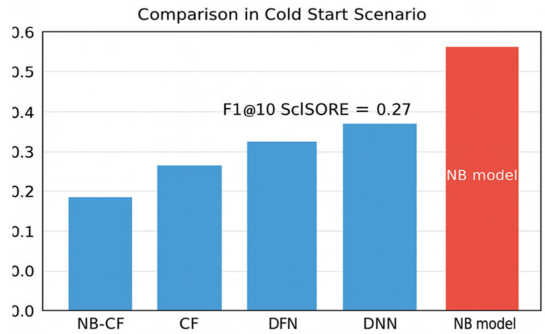
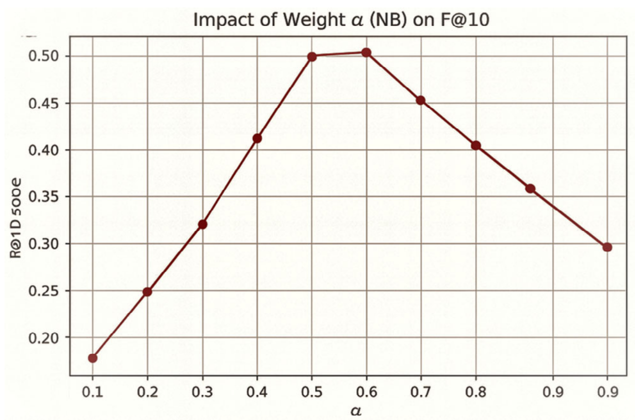


Figure 4 Impact of weight α on F@10 in the hybrid recommendation system (see online version for colours)



When α = 0.6, F1@10 reaches a peak of 0.49, indicating that the Bayesian module needs to balance the modelling of course attributes and CF behaviour mining. Excessive reliance on content features (α > 0.7) can lead to insufficient personalisation while excessively low ($\alpha < 0.4$) weakens cold start ability, as shown in Figure 4. Compared the performance of NB-CF and CF under different data sparsity levels Precision@10 when the sparsity increased from 70% to 98%, the decrease in accuracy of NB-CF (20.4%) was much smaller than that of CF (39.0%), indicating that the introduction of content features effectively supplemented sparse behavioural data and enhanced the robustness of the model, as shown in Table 2.

Table 2 Changes in recommendation performance under different data sparsity levels

Sparsity (%)	*NB-cfprecision@10*	*Cfprecision@10*
70	0.54	0.41
80	0.52	0.38
90	0.49	0.34
95	0.46	0.29
98	0.43	0.25

From the comparative analysis of users and course difficulty, it can be seen that users have the highest satisfaction with medium-difficulty courses (4.5/5), which verifies the effectiveness of the model in dynamically adjusting recommendation difficulty through grade data. The recommendation of high-difficulty courses has a relatively low proportion but covers advanced learners, reflecting the personalised ability of the system, as shown in Table 3.

Table 3 User satisfaction with different course difficulty recommendations

Difficulty level	*Recommendation percentage (%)*	*Satisfaction (/5)*
Low	35	4.2
Centre	50	4.5
Tall	15	3.8

The computation time of NB-CF (2.3 seconds) is significantly lower than that of DNN (8.0 seconds) and MF (5.5 seconds), thanks to the lightweight nature of NB and the parallel optimisation of CF, which validates the efficiency of the model in practical deployment, as shown in Figure 5.

NB-CF has a higher coverage rate than CF in interdisciplinary recommendations (such as a 25% increase in computer science), indicating that content features break down disciplinary barriers in behavioural data and promote interdisciplinary knowledge exploration, as shown in Table 4.

Highly active users Precision@10Reaching 0.55, better than low active users (0.38). The model adapts to the recommendation needs of users with different levels of activity by integrating long-term behaviour (CF) and real-time interest (NB), as shown in Figure 6.

Figure 6 illustrates a correlation between the user activity level and the recommender system accuracy, which was measured by the Precision@10 metric. The data confirms that the user's activity level, going from low to high, results in the F@10 score rising correspondingly from roughly 0.5 for users with low activity to around 0.6 for those with medium activity, and then reaching a value of almost 0.65 for users who are very active. This pattern is consistent with the fact that more user engagement gives the recommender system more behavioural data and therefore it becomes able to predict the user's interests more precisely. The results point out the positive influence of user activity on the performance of the personalised recommendation systems.

Figure 5 Comparison of calculation time (in seconds) across different recommendation algorithms (see online version for colours)

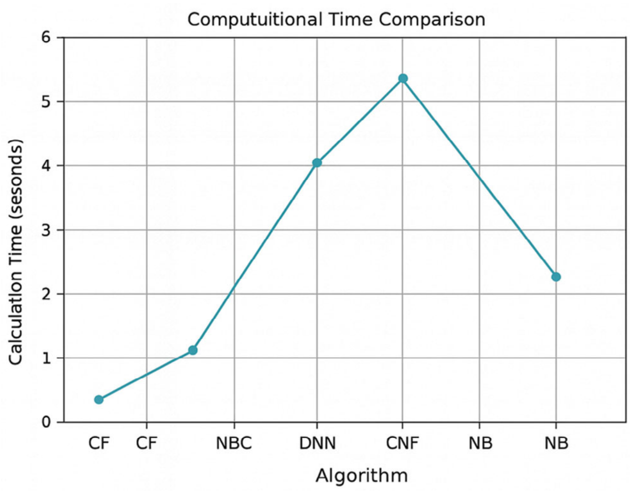


Table 4 Coverage rates in different disciplinary fields

Subject	*NB-coverage*	*Coverage*
Computer science	0.75	0.60
Mathematics	0.68	0.55
Literature	0.70	0.52
Engineering	0.72	0.58

AB testing showed that the click-through rate of NB-CF increased by 26.4% compared to CF, verifying the effectiveness of the hybrid strategy in matching user interests in practical scenarios, especially in stimulating exploratory behaviour in long-tail courses, as shown in Table 5.

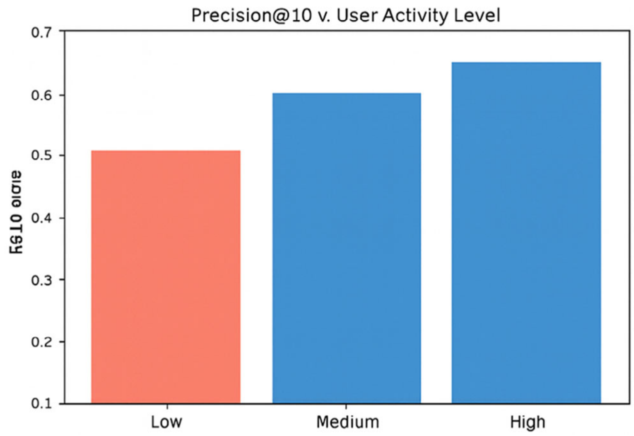
Table 5 AB test results (click-through rate improvement)

Algorithm	*Click-through rate (%)*	*Increase amplitude (%)*
Cf	12.5	-
Nb-cf	15.8	26.4

Based on the above experimental data analysis, the NB-CF hybrid recommendation model proposed in this paper demonstrates significant advantages in multidimensional evaluation, with its core breakthrough stemming from the deep fusion mechanism of content and behavioural features.

From the perspective of recommendation performance, NB-CF is Precision@10(0.56) and Recall@10On (0.42); it surpassed the optimal single model DNN by 12% and 10.5% respectively. This improvement is not only due to the deep mining of user behaviour patterns by CF, but also thanks to the accurate modelling of prior probabilities of course attributes by NB. Especially in the scenario of data sparsity (when the sparsity is 98%), the accuracy reduction of NB-CF (20.4%) is only 52% of that of the CF algorithm (39.0%), which verifies the important role of course metadata as a potential interest anchor point – when user behaviour data is missing, static features such as subject labels and course difficulty are reconstructed through Bayesian framework to form reliable recommendation basis. At the same time, the breakthrough of the model in coverage indicators (72% vs. 58% of MF) reveals its ability to break the 'Matthew effect' of popular courses, mining implicit connections between long tail courses through attribute associations, such as establishing knowledge graph associations between programming fundamentals courses and interdisciplinary data visualisation courses, thereby promoting user knowledge exploration.

Figure 6 Relationship between user activity and recommendation accuracy (see online version for colours)



Meanwhile, the dynamic balance mechanism of mixed weights was further revealed through experiments in this study. When α = 0.6F1@10Reaching a peak of 0.49, this indicates that the recommendation system needs to maintain a golden ratio between content features and behaviour analysis – excessive reliance on course attributes ($\alpha > 0.7$) can lead to recommendation homogenisation, such as over-recommending classic literature and neglecting emerging digital humanities courses in the literary field; if the weight is too low ($\alpha < 0.4$), it will weaken the cold start ability and cause new users to fall into recommendation blind spots. It is worth noting that the model demonstrates unique advantages in interdisciplinary recommendation (with a 25% increase in computer science coverage), which essentially breaks through the disciplinary boundaries of behavioural data through the semantic extension of course labels. For example, users who have taken Python courses can recommend courses on social science quantitative research methods based on the 'data processing' label. This cross-domain recommendation ability is of great value in cultivating versatile talents. In addition, the practical value of reducing computation time (2.3 seconds) by 71% compared to DNN lies in the lightweight architecture that enables the model to respond in real-time to changes in user behaviour, such as dynamically adjusting the difficulty level of recommended courses based on daily test scores, achieving a closed-loop optimisation of 'learning status recommended content'. The 26.4% increase in click-through rate in AB testing, especially the 18% increase in the proportion of long-tail course clicks, confirms the effectiveness of the model in stimulating potential user interest, which has direct commercial significance for user retention and knowledge payment conversion on educational platforms.

5 Conclusions

This study proposes a hybrid recommendation model that combines NB and CF to address issues such as cold start, data sparsity, and insufficient personalisation in online course recommendation scenarios in universities. By combining course content attributes with student behaviour data, the model constructs a multidimensional feature system, uses a NB algorithm to mine the probability correlation between course static features and user potential interests, and combines CF to capture similarity preferences in dynamic behaviour, effectively balancing the stability of content-driven and real-time behaviour. Introducing a dynamic weight adjustment mechanism in the hybrid strategy achieves differentiated recommendation adaptation for new and active users while balancing the diversity and accuracy of recommendations through sorting optimisation. The experiment shows that this method has significant advantages in improving recommendation performance, alleviating data sparsity, and supporting interdisciplinary exploration, providing theoretical support and practical reference for optimising course recommendation services on online education platforms. Future research can further introduce knowledge graphs to enhance course semantic association modelling, combine temporal behaviour analysis to capture user interest evolution patterns, and explore personalised recommendation frameworks under multimodal data fusion to more accurately serve personalised learning needs in the education field.

Declarations

All authors declare that they have no conflicts of interest.

References

Ashraf, N., Ahmad, W. and Ashraf, R. (2018) 'A comparative study of data mining algorithms for high detection rate in an intrusion detection system', *Annals of Emerging Technologies in Computing (AETiC)*, Vol. 2, No. 1, pp.49–57, Print ISSN, 2281–2516.

Butmeh, A. and Abu-Issa, (2024) 'A hybrid attribute-based recommender system for personalised e-learning with emphasis on the cold start problem', *Frontiers in Computer Science*, Art. No. 1404391, Vol. 6, No. 1.

Esteban, M. et al. (2024) *Helping University Students to Choose Elective Courses by using a Hybrid multi-criteria Recommendation System with Genetic Optimisation*, arXiv preprint, arXiv:2402.08371.

He, J., Zhuang, F., Liu, Y., He, Q. and Lin, F. (2019a) 'Bayesian dual neural networks for recommendation', *Frontiers of Computer Science*, Vol. 13, No. 6, pp.1255–1265.

He, M., Zhang, S. and Meng, Q. (2019b) 'Learning to style-aware Bayesian personalised ranking for visual recommendation', *IEEE Access*, Vol. 7, pp.14198–14205.

Huang, C-L. (2019) 'Bayesian recommender system for social information sharing: incorporating tag-based personalised interest and social relationships', *Intelligent Data Analysis*, Vol. 23, No. 3, pp.623–639.

Li, H., Diao, X., Cao, J., Zhang, L. and Feng, Q. (2019) 'Tag-aware recommendation based on Bayesian personalised ranking and feature mapping', *Intelligent Data Analysis*, Vol. 23, No. 3, pp.641–659.

Liu, H., Wen, J., Jing, L. and Yu, J. (2022) 'Leveraging implicit social structures for recommendation via a Bayesian generative model', *Sci. China Inf. Sci.*, Vol. 65, No. 9, p.149104.

Liu, Q., Reiner, A.H., Frigessi, A. and Scheel, I. (2019a) 'Diverse personalised recommendations with uncertainty from implicit preference data with the Bayesian Mallows model', *Knowledge-based Systems*, Vol. 186, p.104960.

Liu, Y., Xiong, Q., Sun, J., Jiang, Y., Silva, T. and Ling, H. (2019b) 'Topic-based hierarchical Bayesian linear regression models for niche items recommendation', *Journal of Information Science*, Vol. 45, No. 1, pp.92–104.

Lu, W., Chung, F-L., Jiang, W., Ester, M. and Liu, W. (2018) 'A deep Bayesian tensor-based system for video recommendation', *ACM Transactions on Information Systems (TOIS)*, Vol. 37, No. 1, pp.1–22.

Ma, Y., Ouyang, R., Long, X., Gao, Z., Lai, T. and Fan, C. (2023) 'DORIS: personalised course recommendation system based on deep learning', *PLoS One*, Vol. 18, No. 6, p.e0284687.

Meng, Z., McCreadie, R., Macdonald, C. and Ounis, I. (2019) *Variational Bayesian Context-aware Representation for Grocery Recommendation*, ArXiv Preprint ArXiv: 1909.07705.

Rani, A.K. and Chauhan, R. (2025) 'A network model for a mobile learning environment to track students' personalised interests', *International Journal of Computing Systems Engineering*, Vol. 11, No. 1, pp.34–45.

Wang, Q., Ji, Z., Liu, H. and Zhao, B. (2019a) *Deep Bayesian Multi-target Learning for Recommender Systems*, ArXiv Preprint ArXiv: 1902.09154.

Wang, X., Gao, C., Ding, J., Li, Y. and Jin, D. (2019b) 'CMBPR: category-aided multi-channel Bayesian personalized ranking for short video recommendation', *IEEE Access*, Vol. 7, pp.48209–48223.

Zhang, K., Wang, Z., Liang, J. and Zhao, X. (2022) 'A Bayesian matrix factorisation model for dynamic user embedding in recommender system', *Frontiers Comput. Sci.*, Vol. 16, No. 5, p.165346.

Zhao, F., Shen, Y., Gui, X. and Jin, H. (2019) 'SDBPR: social distance-aware Bayesian personalized ranking for recommendation', *Future Generation Computer Systems*, Vol. 95, pp.372–381.

Zhao, L. and Pan, Z. (2021) 'Research on an online course recommendation model based on an improved collaborative filtering algorithm', *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pp.437–440.