A Survey of Statistical Models and Algorithms in Personalized Recommender Systems: The Case of Skincare Recommendation

Hongtao Chen *

Guangzhou Foreign Language School, Guangzhou, China

* Corresponding Author Email: 18026215938@163.com

Abstract. In the era of information overload, personalized recommender systems are essential, particularly in critical domains such as skincare where incorrect suggestions have large physical risks. This presents a specialized problem whose demands for accuracy, explainability, and safety in the algorithms are unprecedented. This paper presents an extensive technical roadmap to the researcher, systematically examining the developmental history of recommender algorithms, covering the principal collaborative filtering and content-based filtering algorithms, the latent factor revolution with matrix factorization algorithms, and finishing with the newest deep learning and graph neural network-based algorithms for these systems. Some of this work covers the newest areas of research into how present systems can be trained to become "trustworthy decision partners", discussing causal inference for debiasing techniques, explainability techniques for building trust in users, and multi-dimensional, performance measures beyond accuracy. Using skin-care as a core subject, this paper examines how far the above paradigms are applicable to a very complex and critical real-world scenario. This will also provide an interesting view of future developments in hyper-personalized recommendation.

Keywords: Personalized Recommendation, Skincare Recommendation, Matrix Factorization, Deep Learning, Graph Neural Networks.

1. Introduction

1.1. Formal Definition and Background of the Recommendation Problem

The exponential developments of information technology have led society into an era of information overload wherein individual users find it difficult to effectively span an immense amount of data in an attempt to find the information they want. The personalized recommender systems have developed as an advanced information filtering technology. Their core objective is to predict a user's preference for items and proactively present information that may interest them, thereby enhancing user experience, increasing platform stickiness, and assisting in user decision-making [1].

Formally defined, let C be the set of all users and S be the set of all recommendable items. The core task of a recommender system is to learn a utility function $u: C \times S \to R$, where R is a totally ordered set (e.g., rating values or preference probabilities). This function u is used to estimate the degree of preference of a specific user $c \in C$ for a specific item $s \in S$. Therefore, the essence of recommendation is to estimate the utility value for unobserved pairs (c, s) in the user-item matrix and to recommend a subset of items $S' \in S$ to the user that maximizes the expected utility [1].

1.2. Skincare Recommendation: The Specificity of a High-Stakes Domain

Among many application areas, skincare e-commerce is a typical example of a "high-stakes" scenario. Unlike entertainment-related consumption such as books or movies, the decision consequences of skincare recommendations have direct physiological relevance. Incorrectly tailored recommendations can create skin irritations, allergies, or make existing skin problems worse [2]. These possible problems tremendously increase the requirements of accuracy, trustworthiness, and safety for the system.

This sophistication in the field comes because of the multidimensional and extremely specific nature of the data [3]. On the user side, a good recommendation must not depend too much on the shallow purchase history but must provide an in-depth, multi-dimensional understanding of the physiological characteristics of the user [4]. These characteristics consist, inter alia, of skin type (e.g., oily, dry, combination, sensitive), skin problems (e.g., acne, hyperpigmentation, wrinkles, etc.), allergy history, lifestyle, and even environment [5]. On the product side, the system must be able to parse the complex characteristics of the product. This includes understanding the long ingredient lists, determining active and potentially irritating ingredients, assessing the consequences of combining certain active ingredients, and identifying the nature of the products (e.g., cruelty-free etc.) [2].

Within this situation the aim of the recommender system is dual: it does not only want to optimize business measurements, such as CTR or conversions, but also to achieve the health of the user, both for medium and long terms, and to build trust with the users in the long term. This therefore strongly influences the design of the algorithms and the choices of algorithms in the recommender system, so that the technology is improved further in a more responsible and controllable way.

1.3. Core Challenges Driving Algorithmic Evolution: Sparsity and Cold Start

While recommender systems are common, the development of such systems has always faced two fundamental technical challenges: data sparsity and cold start. These two points are the main drivers of the continuous evolution of recommendation algorithms.

Data sparsity refers to the fact that in the vast majority of commercial applications, the user-item interaction matrix is enormously sparse. A user generally interacts with only a very small proportion (often less than 1%) of the total items. For instance, on a particular platform containing tens of thousands of products for skincare, a single user will have purchased an extremely limited variety. This extreme sparsity presents a serious challenge to traditional algorithms - which generally rely upon discovering overlapping interactions for either users or items (as in CF algorithms). When the amount of overlapping data becomes scant, therefore, the confidence in the similarity testing of such data sinks more and more to the point that it may be impossible to perform necessary calculations.

The cold-start problem, a particular manifestation of the sparsity dilemma [6], consists of three issues, to wit, the "new user cold start", where nothing is known of a newly registered user by the system and personalized recommendations cannot be made; the "new item cold start", where newly listed products lack records of interaction and the collaborative filtering models are unable to recommend them; and the "system cold start", where a world-wide shortage of data occurs at the inception of the system.

To surmount these difficulties, recommender systems have graduated from neighborhood methods to mixed systems capable of utilizing side information. Subsequently, they evolved into inductive learning models capable of automatically learning deep feature representations of data.

1.4. Paper Structure

This systematic review will trace the history of personalized recommendation algorithms, showing the gradual process by which the issues mentioned above were addressed. The next section will look at the traditional paradigms of collaborative filtering and content-based filtering. Section 3 will provide an in-depth analysis of latent factor models, represented by matrix factorization (MF). Sections 4 and 5 will explore the applications of deep learning models and graph neural networks (GNNs) in recommender systems, respectively. Section 6 will discuss cutting-edge research directions, including causal inference, explainability, novel evaluation systems, and future trends in specific domains. Section 7 will conclude the paper.

2. Fundamental Paradigms: Neighborhood Methods and Content Matching

In the early development of recommender systems, two classic paradigms were formed: collaborative filtering (CF) and content-based filtering (CBF). They established the theoretical and practical foundations for personalized recommendation from two distinct perspectives: "collective intelligence" and "individual preference" [1].

2.1. Collaborative Filtering: Preference Inference Based on the Crowd

Memory-based collaborative filtering is one of the earliest and most intuitive recommendation methods. Its core idea is to use the historical behavior data of a group to make predictions, based on the assumption that users with similar preferences in the past will likely maintain similar tastes in the future [5]. These methods compute directly on the user-item interaction matrix without training complex parametric models. This paradigm is mainly divided into user-based CF and item-based CF. The former predicts a target user's preference by finding a group of "neighbor" users with the most similar interests and aggregating their ratings for an item. The latter, in contrast, calculates the similarity between items, assuming that a user will tend to like items similar to those they have liked in the past. Whether user-based or item-based, the basic task is a matter of determining the distance, or more appropriately the similarity, between entities. This can be accomplished using common measures, such as Pearson correlation, or cosine similarity. The major advantages of CF are its easily grasped intuitive logic, and its ability to uncover latent interests across domains (serendipity). The disadvantages are just as glaring: it is strongly dependent on historical usage information and is weak or nonfunctional in cold-start and sparse-data situations [5]. In addition, as the user or item community increases in size, the computation and maintenance involved in the similarity matrices becomes tremendous, resulting in grave discontinuities in scalability.

2.2. Content-Based Filtering: Preference Matching Based on Attributes

CBF makes use of an entirely different rationale. In contrast to CF, no regard is given to the behavior of other users. Instead, the content attributes of the items themselves and the user's preference profile is where attention is placed. CBF aims to recommend items that are “similar on the content” to previously liked items [1]. The core of this method lies in feature engineering. First, a user profile must be constructed for each user, which may include demographic information, user-provided preferences, or interest tags extracted from their historical behavior. Second, a set of features describing the item's content (Item Profile) must be extracted, such as a skincare product's ingredients, efficacy, brand, etc. [3]. CBF can effectively solve the new item cold-start problem: as soon as content features are extracted from a new product, it can be immediately recommended to users whose profiles match it. Additionally, the recommendation logic is transparent and easily generates explainable reasons. However, its main drawback is over-specialization; by always recommending items similar to the user's known interests, it struggles to discover new points of interest for the user, limiting the novelty of the recommendations. At the same time, it relies heavily on domain knowledge for feature extraction and cannot solve the new user cold-start problem.

In a specialized domain like skincare, CBF plays a vital role. By structuring a user's skin type and needs [3] and matching them with attributes in a product knowledge base (containing ingredients, efficacy, etc.) [2], the system can provide safe, highly relevant recommendations. Yet, to create and maintain such an exact knowledge base entails a tremendous cost itself, which constitutes an industry barrier. The difference between CF and CBF exposes the fundamental trade-off in recommender system design between the issues of “exploration/discovery” and “relevance/safety,” which was the direct spur for the invention of hybrid recommender systems which combine the advantages of both.

3. The Latent Factor Revolution: Matrix Factorization Techniques

The era of MF techniques represented a paradigm shift in recommender systems following CF and CBF. This family of models became very popular for its impressively high performance in the famous Netflix Prize competition [7] and, because of its good prediction performance and high scalability, became the mainstream paradigm for recommender system research and applications for almost a decade.

3.1. The Mathematical Core of MF

The core idea of MF is that in complex scenarios like skincare recommendation, a user's preferences and a product's attributes are often determined by unobservable "latent factors". For example, a user might have different preferences along abstract dimensions like "pursuit of high efficacy" and "focus on natural/gentle"; correspondingly, a product might also have different attributes along these two dimensions. MF aims to automatically learn these latent factors from sparse rating data [7].

This method approximates the high-dimensional, sparse user-item rating matrix R (of dimension $m \times n$, where m is the number of users and n is the number of items) as the product of two low-rank, dense matrices: a user-latent factor matrix P (dimension $m \times f$) and an item-latent factor matrix Q (dimension $n \times f$), such that $R \approx PQ^T$. Here, f is the number of latent factors (e.g., $f = 50$), representing the number of "internal dimensions" we use to describe users and items.

In this model, each user u is represented by an f-dimensional vector $p_u \in \mathbb{R}^f$ (e.g., the value of this vector on the "high efficacy" dimension), and each item i is also represented by an f-dimensional vector $q_i \in \mathbb{R}^f$ (e.g., the product's score on the "high efficacy" dimension). These vectors co-exist in a shared latent factor space. The predicted rating $\hat{r}_{ui}$ of user u for item i is modeled as the dot product of the corresponding user and item vectors: $\hat{r}_{ui} = q_i^T p_u$. This intuitively measures the overall match between the user's preferences and the product's attributes across all f dimensions. The model's training objective is to find the optimal P and Q matrices that minimize the error between predicted and actual ratings. This is typically achieved by minimizing the regularized squared error over the set of known ratings κ [7]:

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \quad (1)$$

Here, λ is the regularization coefficient used to control model complexity and prevent overfitting. The two main algorithms for solving this optimization problem are Stochastic Gradient Descent (SGD) and Alternating Least Squares (ALS).

3.2. Enhanced Models: Biases, Temporal Dynamics, and Implicit Feedback

While the basic matrix factorization model is powerful, research by Koren et al. further demonstrated that its predictive power could be significantly enhanced by incorporating more real-world factors [7]. The first result is that of including bias terms. In real-world rating data there will be systematic differences that are unrelated to the intrinsic user-item interaction, which are due to certain users being "lenient" or "harsh", or certain items being "popular" or "niche". The model thus incorporates this bias by the addition of bias terms in the prediction formula: $\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$ (where μ is the global average rating, b_u is the user bias, b_i is the item bias). This separates the rating into global, individual and interaction effects, so that the latent factor model $q_i^T p_u$ can concentrate on modeling the pure personalized match [7].

The next aspect is the consideration of temporal dynamics. Users' interests and the popularity of items gradually change with time, e.g., the taste for seasonal skin-care items. In advanced models, time is a variable, so the bias terms or even the user latent factors are functions of time (e.g., $b_i(t)$, $p_u(t)$). This structure points to the dynamics of the evolution of user interests and the item life cycle.

Finally, implicit feedback is handled. In many real-world situations available may be far more implicit behavior representing approving like clicks, views, purchases than the explicit ratings. MF

can be adapted to this situation. For example, implicit interaction behavior may be treated as positive signals of preference and assigned a confidence level (many interactions means more confidence), items not interacted with would similarly be treated as negative signals with less confidence. The aim of the model shifts to fitting the confidence weighted preferences. [7, 8].

The advent of MF meant the paradigm shift from explicit calculation of similarity (as in neighborhood models) to implicit representation of similarity in the recommendation systems. In the latent factor space introduced the calculations of similarity were not explicit, but emergent from the model learning process. The ability to perform this abstraction meant that unknown correlations, which were not able to be extracted through traditional methods, could be captured which is a significant advance in the technology of recommendation systems.

4. The Deep Learning Era: Capturing Non-linearity and Sequential Dynamics

With the advances made by deep learning in applications such as computer vision and natural language processing, the models enabled by it, with their great representation learning and non-linear modeling capabilities, were also applied to recommender systems, thus launching a new era [9]. The deep learning models because of their flexibility break the linear assumptions made by standard statistical models such as MF. More complex and deeper interaction measures between users and items can therefore be possible and effectively model the dynamic evolution of user preferences.

4.1. Neural Collaborative Filtering: Beyond the Linear Dot Product

The core of MF is the dot product of user and item latent factor vectors, $\hat{r}_{ui} = q_i^T p_u$, which is inherently a linear interaction [7]. However, in the skincare domain, the relationship between user preferences and product attributes is often highly non-linear. For example, an "oily skin" user's preference for an "oil-control" product may be positively correlated, but a "dry skin" user's preference for the same product would be strongly negatively correlated. More complexly, a "combination skin" user's preference for a "gentle oil-control" product might not be expressible through a simple linear combination. The linear assumption of the dot product limits the model's ability to express such complex relationships.

Deep learning, particularly the multi-layer perceptron (MLP), provides a powerful tool for learning arbitrary non-linear functions, which can replace or enhance the dot product operation. Neural collaborative filtering (NCF) is a general framework [9] that considers MF as its special case [10]. In this framework, users and items are first converted into low-dimensional embedding vectors. Subsequently, instead of a simple dot product, these embedding vectors can be concatenated and fed into a MLP. Through multiple layers of non-linear activation functions (like ReLU), the MLP can learn an arbitrarily complex interaction function between the user and item embeddings to output the final prediction score. Google's Wide & Deep model is a landmark application of deep learning in recommendation [11]. It cleverly combines a simple linear model (the Wide part, for "memorizing" common feature combinations in historical data) with a deep neural network (the Deep part, for "generalizing" to unseen feature combinations) in a joint training framework. This architecture combines the interpretability and scalability of linear models with the strong generalization ability of deep models, and it has been widely adopted in the industry.

4.2. Modeling the User Journey: Sequential Recommendation Models

Users' interests are dynamic, and their behaviors often show sequentiality. In skincare, this sequentiality is particularly pronounced, and might be reflected in skincare steps (for example, a user and their purchasing flow are likely to be "cleanser -> toner -> serum -> moisturizer") or could alternatively signify a change in the user's needs (a user will buy "moisturising cream" in the winter, but switch to "light sunscreen) in summer). Both a user's short-term intent and their long-term interests jointly identify what action they will take next. Traditional static models (such as MF) learn a fixed preference vector for each user and thus do not readily incorporate this time dependency.

Sequential Recommendation models were developed precisely to solve this problem. Early research employed recurrent neural networks (RNNs) and their variants (like long short-term memory (LSTM) and gated recurrent unit (GRU)) [12]. They process a user's behavior sequence sequentially via a hidden state, which is treated as a summary of the user's dynamic interest. In recent years, the Transformer architecture, centered on the self-attention mechanism, has become dominant in sequential recommendation, with models like self-attentive sequential recommendation (SASRec) [13] and its successors (e.g., BERT4Rec [14]). Unlike RNNs, which process one item at a time, the self-attention mechanism can parallelly compute the mutual influences among all items in a user's history sequence and assign different attention weights to different historical items. This allows the model to flexibly capture both a user's long-term preferences (like loyalty to a specific brand) and short-term intent (like a recent focus on "whitening" efficacy), enabling more accurate real-time predictions.

4.3. Fusing Rich Content: The Representation Power of Deep Learning

A powerful aspect of deep learning is its capability of effectively perceiving the feature space of data automatically, from the unstructured input represented by raw data [9]. This allows recommender systems to break away from a CF system which takes input only in terms of user ids and item ids and takes into account more complete information given to models in the form of rich textual, visual, and multimodal information (e.g., visual bayesian personalized ranking (VBPR) [15]). In the skincare domain, natural language processing (NLP) methods (e.g., bidirectional encoder representations from transformers (BERT)) can be used in terms of analyzing a product description and user reviews to automatically derive semantic features such as “lightweight texture” or “friendly for oily, acne prone skin” and to even use ingredient systems to predict efficacy [2]. At the same time, computer vision (CV) techniques (e.g., Convolutional Neural Networks (CNNs)) can be used to analyze user-uploaded facial photographs, extracting information regarding skin type and automatically picking up specific issues such as “redness” or “dark spots” [2, 3]. Separately, the output of this structured diagnostic regime can also be incorporated directly into the recommendation engine to help define the various profiles for users.

The use of deep learning in recommender engines represents a fundamental shift from traditional feature engineering to end-to-end representation learning. Additionally, the advent of sequential models represents a transition in the understanding of user intent from static profiles to dynamic modeling therefore refocusing attention to “what the user is inclined to do now”.

5. Graph-Based Models: Recommending on Interconnected Networks

Recently, the framing of the recommendation problem within the framework of a graph and the utilization of GNNs for learning have emerged as some of the most advanced and active areas of research in recommender systems. The idea considers users and items as the nodes of a graph and their interactions to be edges. Powerful learning techniques for graph representation can then be utilized to exploit the rich, high-order structural information contained in the data.

5.1. From Bipartite Graphs to Heterogeneous Information Networks

User-product interaction information can be naturally modeled as a bipartite graph with user nodes on one side and product nodes on the other, edge connections between user and product nodes representing interactions. However, the information content available for modeling in a recommendation context extends far beyond pure user-product interaction. A more sophisticated modeling avenue is that of a heterogeneous information network (HIN). For example, in the context of skincare recommendation, an HIN can consist of multiple types of entities represented as nodes, such as user, product, ingredient, brand, efficacy type and skin problems. These nodes have connecting edges of types such as “product-contains-ingredient”, “product-comes-from-brand”, and “user-has-skin-problem”. This heterogeneous graph structure has the potential for richer expression

of the intricate relationships in the recommendation network, and enables the model to use multi-hop, cross-type links between entities.

5.2. Learning with Graph Neural Networks

A GNN is a form of deep learning model designed specifically for learning on graph input. The main idea is a recursive process of gathering information from the neighboring nodes of a particular node, termed "message passing" or "neighborhood aggregation," to learn a vector representation, or (embedding), of that node. This process is powerful for a heterogeneous graph (HIN) for skincare recommendation. For example, to learn the representation for "User A," the first layer (1-hop) of the GNN would aggregate information from "Product X" (which A purchased) and "Skin Problem Y" (which A has). In the second layer (2-hop), the model would further aggregate information from the neighbors of "Product X" (e.g., "Ingredient Z," "Brand W") and the neighbors of "Skin Problem Y" (e.g., "User B," who also has "Problem Y").

After multiple rounds of aggregation, the final representation of each node will encapsulate the structural information within its high-order neighborhood. Graph convolutional networks (GCNs) and their simplified variants (like LightGCN [16]) learn node representations by propagating and transforming neighbor features on the graph, focusing on capturing the collaborative signal (earlier GCN-based models like NGCF [17] were more complex). Graph attention networks (GATs), on the other hand, introduce an attention mechanism into the neighborhood aggregation process. This allows the model to learn a different attention weight for each neighbor (e.g., when aggregating information, the model might find "Ingredient Z" more important than "Brand W"), enabling it to selectively focus on more important neighbors and granting the model stronger expressive power.

5.3. Inductive Learning Capability of GNNs: Solving the New Item Problem

A key advantage of GNNs is their potential to solve the cold-start problem. Many early GNN models were transductive, meaning they could only generate embeddings for nodes seen during training. When new nodes (new users or items) were added to the graph, the model had to be retrained. However, architectures like GraphSAGE (Graph SAmple and aggreGatE) [18] were designed to be inductive. Instead of learning a fixed embedding for each node, it learns an aggregator function. This function defines how to dynamically generate a node's embedding by sampling and aggregating features from its neighbors. Therefore, in the skincare scenario, when a new product is listed, as long as its information (e.g., its ingredients, its brand) is added to the graph as new nodes and edges, the inductive GNN model can immediately apply the learned aggregator function to generate a meaningful embedding vector for it using information from its neighbors (i.e., ingredients and brand), all without retraining the entire model. This provides an effective and elegant solution to the new item cold-start problem of recommendation systems.

The introduction of GNNs represents a fundamental leap in the paradigm of recommender system modeling: from modeling isolated (user, item) pairs of users and items to the extensive modeling of an entire recommendation ecosystem. In particular, inductive GNNs represent a deep fusion of the two classic ideas of CBF and CF, as they utilize the collaborative structure to learn a function which can take content features to a representation which is collaborative-aware.

6. Emerging Frontiers: Towards More Robust and Trustworthy Recommendation

As the technology behind recommender systems continues to progress, research is beginning to shift its focus from almost exclusively developing metrics of predictive accuracy toward developing systems that are more fair, more explainable, and better able to meet the users' real needs. Causal inference, explainability, and evaluation systems that “go beyond accuracy” are three of the key areas of current research that lead in this transformation.

6.1. Causal Inference: The Leap from Correlation to Causation

Traditional recommender systems are essentially correlation learning models. They are good at learning correlations between variables, but they cannot distinguish correlation from causation. This causes the model to learn many "spurious correlations" in the recommendation context. A confounder that commonly occurs is popularity. An item can be recommended often because it is high quality or it has been promoted by many people, too. This can cause a feeding back loop of, "the more popular, the more recommended, the more recommended the more popular." This can give rise to popularity bias thus obfuscating the users true personalized preferences [19]. Causal inference is the procedure for estimating the true causal effect of an "influence" (e.g., an item recommended to the user) on an "outcome" (e.g., user satisfaction) while controlling for the effect of confounding variables. Using frameworks that comport with the potential outcome model or structural causal models (SCMs), researchers employ techniques such as backdoor adjustment and inverse propensity scoring (e.g., propensity score based debiasing [20]) to debias the training data. This helps the model learn the users pure preference for an item independent of the confounding variables such as popularity and exposure bias.

6.2. Explainable AI (XAI): Answering "Why" to Build User Trust

As models like deep learning and GNNs become widely used in recommender systems, these systems have increasingly become “black boxes,” with the decision-making logic behind them not readily understandable. Explainable recommendation attempts to provide the user with reasoning as to why recommendations are made to provide transparency of the process [21]. The application of explainability is especially necessary in high stakes areas such as skincare. Providing justification can greatly increase a system’s transparency, persuasiveness, effectiveness, trust and satisfaction to users [21]. A high quality of such an explanation might be: “This serum is recommended for you because it contains “hyaluronic acid” which is a substance that effectively addresses your concern of “dry skin,” further reviewed positively by customers characterized as “sensitive skin” as in your general description” (e.g., by reasoning about a knowledge graph [22]). This form of explanation encompassing product attributes, consumer needs and social feedback greatly enhances the user’s trust and acceptance of the recommendation.

6.3. Beyond-Accuracy Evaluation: Multi-Dimensional Metrics for Modern Recommender Systems

For a long time, the evaluation of recommender systems focused mainly on accuracy metrics, such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) [23]. However, both research and practice have shown that high accuracy does not always translate to high user satisfaction. A system that recommends only very popular bestsellers may have very high accuracy but cannot add personalization or do anything to help users discover new items. Therefore, for the application of producing recommendation lists (Top-N Recommendation) the ranking of items is of immense importance. A number of ranking aware metrics have been widely used, namely, Precision@K, Recall@K, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) [23]. Additionally, it is crucial that a well-thought out system for evaluation considers different dimensions to accuracy [24], such as: coverage, the percentage of the total items in the catalogue that can be recommended by the system; diversity, the divergence between the items in a single recommendation list; and novelty & serendipity, the ability of the system to help users broaden their experience and interests. This group of metrics reflects the long-term value of the system exploited and the overall good health of the ecosystem of the platform.

6.4. Specific Challenges and Future Trends in Skincare Recommendation

Nevertheless, there are a significant number of unresolved problems in the specific area of skincare advice which provide direction for future research. The first, and perhaps the most important, is dynamic user profiling. A user's skin condition is seldom static, but is influenced by age, seasonal

influences, hormonal influences, etc. [4]. How to develop a dynamic user model that is capable of describing and adapting to these long-term and short-term physiological changes is clearly a very great problem. The second problem is that of the quantification of the interactions between the ingredients. The activity of skincare products is largely dependent upon the synergistic or antagonistic actions of several ingredients when used in the same formulation. Future research must build models of ingredients from a chemical and bioinformatics basis to quantify the complex relationships that exist between the ingredients [2]. Third, the integration of unstructured expert knowledge is relevant. There exists a vast body of skincare knowledge in the advice of dermatologists, professional aestheticians and unstructured text. It is essential to extract, authenticate, and integrate this expert knowledge in recommendation models to enhance the professionalism of the recommendations. Finally, there is privacy and protection of sensitive data. The skin condition of the user, the history of allergies and other important facts should be regarded as highly sensitive personal health information. Thus, the design of privacy preserving recommendation systems (i.e., federated learning, differential privacy) is of high importance.

In future skincare recommendation, because of powerful algorithms, the field will likely see the advent of a huge new level of “hyper-personalization”. This will include the integration of real-time data from wearables or sensors to dynamically alter skincare advice, optimally fitted for current environmental circumstances (UV index, air humidity) [4]; with the falling cost of genetic testing, integrating a user's genetic data into their profile to achieve gene-based customized recommendations; and with the rise of large language models (LLMs), which provide a new paradigm for recommender systems [25], the interaction method will shift from one-way "push" to two-way "dialogue," where users can ask questions in natural language and receive trustworthy explanations, just as they would from a human expert.

7. Conclusion

This paper has reviewed systematically the evolutionary history of statistical models and algorithms in personalized recommender systems, showing a clear technology track from heuristic approaches based on the directly observable data (CF, content of products), to statistical models that mine latent structures in the data (MF), finally to representation learning models that can automatically learn complex non-linear relationships and high-order structural information (Deep learning, GNNs). On this basis, the paper further explored emerging frontier directions, including causal inference, explainability, and multi-dimensional evaluation systems. By using skincare, a high-stakes domain, as an example, this paper has analyzed the applicability and limitations of different paradigms in solving specific challenges such as data sparsity, cold start, and the integration of expert knowledge. It aims to provide a comprehensive and insightful technical roadmap for researchers and practitioners in related fields.

References

[1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 2005, 17 (6): 734-749.

[2] Lee J, Yoon H, Kim S, et al. Deep learning-based skin care product recommendation: A focus on cosmetic ingredient analysis and facial skin conditions. Journal of Cosmetic Dermatology, 2024, 23 (6): 2066-2077.

[3] Ramrakhiani N, Kalbande D. A comprehensive review of AI-powered skincare product recommendation systems: From data collection to user experience. E-Learning and Digital Media, 2024, 20427530241304073.

[4] Grech V S, Kefala V, Rallis E. Cosmetology in the era of artificial intelligence. Cosmetics, 2024, 11 (4): 135.

[5] Chen R, Hua Q, Chang Y S, et al. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. IEEE access, 2018, 6: 64301-64320.

[6] Panda D K, Ray S. Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. Journal of Intelligent Information Systems, 2022, 59 (2): 341-366.

[7] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009, 42 (8): 30-37.

[8] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE international conference on data mining, 2008: 263-272.

[9] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives. ACM computing surveys (CSUR), 2019, 52 (1): 1-38.

[10] He X, Liao L, Zhang H, et al. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web, 2017: 173-182.

[11] Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems. In Proceedings of the 1st workshop on deep learning for recommender systems, 2016: 7-10.

[12] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939, 2015.

[13] Kang W C, McAuley J. Self-attentive sequential recommendation. In 2018 IEEE international conference on data mining (ICDM), 2018: 197-206.

[14] Sun F, Liu J, Wu J, et al. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM international conference on information and knowledge management, 2019: 1441-1450.

[15] He R, McAuley J. VBPR: visual bayesian personalized ranking from implicit feedback. In Proceedings of the AAAI conference on artificial intelligence, 2016.

[16] He X, Deng K, Wang X, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020: 639-648.

[17] Wang X, He X, Wang M, et al. Neural graph collaborative filtering. In Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, 2019: 165-174.

[18] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. Advances in neural information processing systems, 2017, 30.

[19] Luo H, Zhuang F, Xie R, et al. A survey on causal inference for recommendation. The Innovation, 2024, 5 (2).

[20] Schnabel T, Swaminathan A, Singh A, et al. Recommendations as treatments: Debiasing learning and evaluation. In international conference on machine learning, 2016: 1670-1679.

[21] Zhang Y, Chen X. Explainable recommendation: A survey and new perspectives. Foundations and Trends in Information Retrieval, 2020, 14 (1): 1-101.

[22] Wang X, He X, Cao Y, et al. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019: 950-958.

[23] Parra D, Sahebi S. Recommender systems: Sources of knowledge and evaluation metrics. In: Advanced Techniques in Web Intelligence-2: Web User Browsing Behaviour and Preference Analysis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 149-175.

[24] Shani G, Gunawardana A. Evaluating recommendation systems. In: Recommender systems handbook. Boston, MA: Springer US, 2010: 257-297.

[25] Wu L, Zheng Z, Qiu Z, et al. A survey on large language models for recommendation. World Wide Web, 2024, 27 (5): 60.